Pentest Web

Clase 2: Reconocimiento

Clase 2: Reconocimiento

N.	Clase	M1	M2	M3	M4
1	**Introducción**	Contexto	Ciberseguridad	HTTP	Hacktitud
2	**Reconocimiento**	Subfinder	Nmap	FFuF	BurpSuite
3	**Acceso**	Fundamentos	Criptografía	Tecnología	IDOR
4	**Incursión**	CVSS	Divulgación	Listas	Otros
5	**Lógica**	Negocio	Flujo	Aritmética	Diseño
6	**Inyección**	SQL	OS	Código	Parámetros
7	**informe**	Equipos	Objetivo	Metodología	Reporte
8	**Conclusión**	Resumen	Reflexiones	CVE	Futuro

Clase 2: Reconocimiento

N.	Clase	M1	M2	M3	M4
1	**Introducción**	Contexto	Ciberseguridad	HTTP	Hacktitud
2	**Reconocimiento**	Subfinder	Nmap	FFuF	BurpSuite
3	**Acceso**	Fundamentos	Criptografía	Tecnología	IDOR
4	**Incursión**	CVSS	Divulgación	Listas	Otros
5	**Lógica**	Negocio	Flujo	Aritmética	Diseño
6	**Inyección**	SQL	OS	Código	Parámetros
7	**informe**	Equipos	Objetivo	Metodología	Reporte
8	**Conclusión**	Resumen	Reflexiones	CVE	Futuro

Clase 2: Reconocimiento

M	Nombre	Descripción
1	Subfinder	Enumeración de subdominios
2	Nmap	Escaneo de puertos
3	FFuF	Fuzzing de parámetros
4	BurpSuite	Herramienta estrella del pentest web

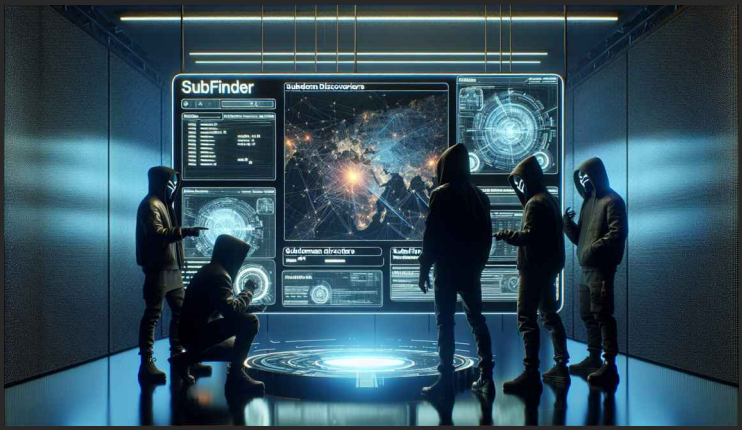
Módulo 1: Subfinder

Módulo 1: Subfinder

S	Nombre	Descripción
1	Descripción	Funcionalidad básica de Subfinder
2	Fuentes	Orígenes de la información sobre subdominios
3	Avanzado	Aplicaciones avanzadas de Subfinder
4	Integración	Conexión de Subfinder con otras herramientas

Sesión 1: Descripción

(Módulo 1: Subfinder)



¿Qué es Subfinder?

Subfinder es una herramienta de código abierto implementada en Go y diseñada para la recolección **pasiva** de subdominios.

Permite a los pentesters descubrir **subdominios** de un dominio objetivo.

Esto facilita el **reconocimiento** y la **enumeración** de activos en una evaluación de seguridad.

Uso de Subfinder

```
# Install
go install -v
 ↪  github.com/projectdiscovery/subfinder/v2/cmd/subfinder@latest

# Help always helps
subfinder -h

# Execute
subfinder -d uc.cl
```

Ejemplo de Subfinder con uc.cl

```
~/Software/Latex/ClassPentestWeb (main) [0]                    [1118/1149]
$ subfinder -d uc.cl

                __     _____           __
   _____  __/ /_  / __(_)___  ____/ /__  _____
  / ___/ / / / __ \/ /_/ / __ \/ __  / _ \/ ___/
 (__  ) /_/ / /_/ / __/ / / / / /_/ /  __/ /
/____/\__,_/_.___/_/ /_/_/ /_/\__,_/\___/_/

                projectdiscovery.io

[INF] Current subfinder version v2.6.8 (outdated)
[INF] Loading provider config from /home/mtourneboeuf/.config/subfinder/provider-config.yaml
[INF] Enumerating subdomains for uc.cl
chovellen.uc.cl
rrhh.adc.dci.uc.cl
politicaspublicas.uc.cl
roble.uc.cl
s10.es.soc.uc.cl
pandora2.ad.uc.cl
biosaga-pregrado.bio.uc.cl
smtp-auth.adc.dci.uc.cl
fapchile.uc.cl
[0] 1:vim- 2:bash* 3:SGPT  4:Dalle                    mtourneboeuf@martint
```

Ejemplo de Subfinder con uc.cl

```
www.centromanuellarrain.uc.cl                                    [0/1149]
biomedicina.sitios.ing.uc.cl
api.mat.uc.cl
bdrt-hist.uc.cl
duraspace.hh.dca.uc.cl
workflow.ec.adc.dci.uc.cl
www.padrejuc3ext.uc.cl
mail.congresosocial.uc.cl
docupersonal-des.uc.cl
mail.pastoral.uc.cl
alumni.mat.uc.cl
mat1100.mat.uc.cl
thebridge.ing.uc.cl
becas125.uc.cl
fcov.dca.uc.cl
cpanel.online.agronomia.uc.cl
semillero.uc.cl
plataforma.adultomayor.uc.cl
cvirtual2.uc.cl
aceptaprod01.dci.uc.cl
[INF] Found 1124 subdomains for uc.cl in 30 seconds 4 milliseconds
~/Software/Latex/ClassPentestWeb (main) [0]
$
[0] 1:vim- 2:bash* 3:SGPT  4:Dalle                    mtourneboeuf@martint
```

Sesión 2: Fuentes

(Módulo 1: Subfinder)

Fuentes de datos

Subfinder utiliza múltiples fuentes de datos para la recolección de subdominios.

- APIs de servicios de terceros (como VirusTotal, SecurityTrails, etc.)
- Bases de datos públicas
- Herramientas de escaneo de DNS

Fuentes abiertas

```
subfinder -ls 2> /dev/null  | grep -v '\*$'
```

alienvault	anubis	commoncrawl	crtsh	digitorus
hackertarget	rapiddns	sitedossier	waybackarchive	hudsonrock

Fuentes con token

```
subfinder -ls 2> /dev/null  |  sed -n -e 's/\*$//p' | sed -e
 ↪  'N;N;N;s/\n/| /g'
```

bevigil	binaryedge	bufferover	c99	**censys**
certspotter	chaos	chinaz	dnsdb	**dnsdumpster**
dnsrepo	fofa	fullhunt	**github**	hunter
intelx	netlas	leakix	passivetotal	quake
redhuntlabs	robtex	securitytrails	**shodan**	threatbook
virustotal	**whoisxmlapi**	zoomeyeapi	**facebook**	builtwith

Sesión 3: Avanzado

(Módulo 1: Subfinder)

Opciones avanzadas

Subfinder ofrece varias **opciones avanzadas** que permiten personalizar la búsqueda, tales como:

- `-d`: Especificar el dominio de entrada.
- `-all`: Seleccionar todas las fuentes (lento).
- `-proxy`: Utilizar un proxy web.
- `-silent`: No mostrar la salida en la consola.
- `-dL`: Especificar un archivo de entrada.
- `-o`: Especificar un archivo de salida.

Sesión 4: Integración

(Módulo 1: Subfinder)

Integración con otras herramientas

Subfinder se puede integrar fácilmente con otras herramientas de reconocimiento y escaneo, como Nmap y BurpSuite, para proporcionar un enfoque más completo en la evaluación de seguridad.

Ejemplo de flujo de trabajo

1. **Recolección de Subdominios**: Utilizar **Subfinder** para obtener una lista de **subdominios** de un dominio objetivo.
2. **Verificación de Subdominios**: Usar herramientas como **Nmap** para escanear los subdominios descubiertos y **verificar** los servicios en ejecución.
3. **Análisis de Resultados**: Analizar los subdominios y servicios encontrados con herramienta como **BurpSuite** para **identificar** posibles vectores de ataque.

Subfinder: conclusión

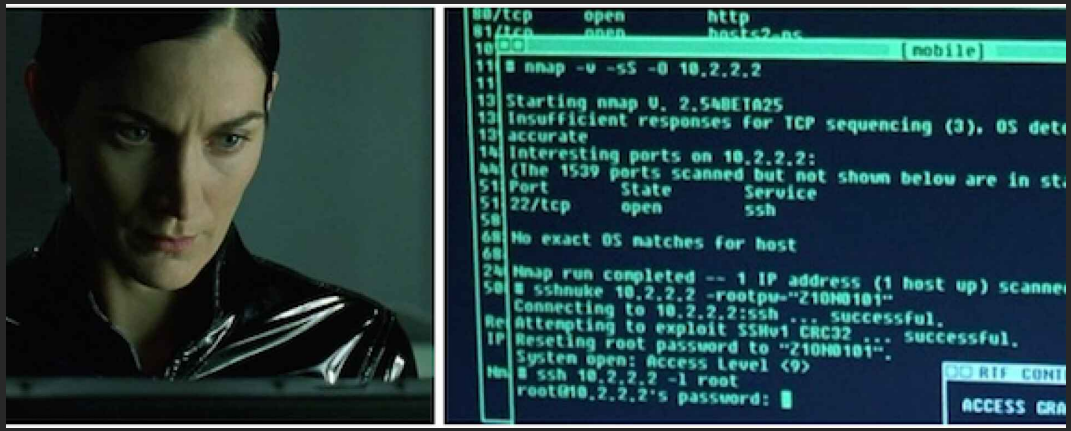
Subfinder es una herramienta esencial en el arsenal de un pentester.

Facilita la recolección de información crítica sobre **subdominios** que suelen pertenecer a la superficie de exposición del blanco y pueden ser explotados durante una evaluación de seguridad.

Su facilidad de uso y capacidad de integración la convierten en una opción popular para la fase de reconocimiento en pruebas de penetración web.

Es el primer paso!

Módulo 2: Nmap



Módulo 2: Nmap

S	Nombre	Descripción
1	Equipos	Identificación de dispositivos activos en una red
2	Puertos	Escaneo de puertos abiertos en los dispositivos
3	Configuración	Opciones avanzadas de configuración
4	Servicios	Determinación de las versiones de las aplicaciones

Sesión 1: Equipos

(Módulo 2: Nmap)

Nmap base

```
sudo nmap www.uc.cl
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2025-04-06 12:35 -04
Nmap scan report for www.uc.cl (34.206.41.117)
Host is up (0.13s latency).
Other addresses for www.uc.cl (not scanned): 52.201.177.18
rDNS record for 34.206.41.117:
 ↪  ec2-34-206-41-117.compute-1.amazonaws.com
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 10.42 seconds
```

Escanear equipos

```
sudo nmap -sn -oA results 10.11.12.0/24
```

Opción de escaneo	Descripción
10.11.12.0/24	Rango de red objetivo
-sn	Desactiva el escaneo de puertos
-oA result	Almacena los resultados en result.*

Rangos de red

Rango	Prefijo	Cantidad
10.11.12.0/24	10.11.12.*	256
10.11.12.0/16	10.11.*.*	65.536
10.11.12.0/8	10.*.*.*	16.777.216
10.11.12.0/0	*.*.*.*	4.294.967.296

Escanear lista de equipos

```
# Archivo de lista
sudo nmap -sn -oA results -iL hosts.txt

# Lista en stdin
sudo nmap -sn -oA results 10.11.12.4 10.11.12.53 10.11.12.88
sudo nmap -sn -oA results 10.11.12.{4,53,88}

# Rango tipo de Perl
sudo nmap -sn -oA results 10.11.12.4-58

# Rango de red
sudo nmap -sn -oA results 10.11.12.0/24
```

Sesión 2: Puertos

(Módulo 2: Nmap)

Escanear puertos

```
# Puertos principales
sudo nmap www.uc.cl --top-ports=10

# Puertos específicos
sudo nmap www.uc.cl -p 80,443,800-1000

# Todo los puertos, como -p 0-65536
sudo nmap www.uc.cl -p-

# Ejemplo real
sudo nmap -vvv --reason ctf.tinmarino.com -T5 -p9000-10000
```

Estado de puertos

Estado	Descripción
Open	Conexión establecida; puede ser TCP, UDP o SCTP
Closed	El puerto está cerrado; se recibió un paquete RST
Filtered	No se puede determinar el estado; sin respuesta o error
Unfiltered	Accesible, pero no se puede determinar el estado
Open/Filtered	Sin respuesta; puede estar protegido por un firewall
Closed/Filtered	Estado indeterminado mediante escaneo IP ID idle

Ejemplo real

```
sudo nmap ctf.tinmarino.com -vvv --script vuln -sV --reason
 ↪  --packet-trace -T5 -oA /tmp/ctf -p9000-10000
```

```
Nmap scan report for ctf.tinmarino.com (54.226.26.40)
Host is up, received reset ttl 54 (0.13s latency).
rDNS record for 54.226.26.40:
 ↪  ec2-54-226-26-40.compute-1.amazonaws.com
Scanned at 2025-04-06 15:11:13 -04 for 60s
Not shown: 998 closed ports
Reason: 998 resets
```

Ejemplo real

```
PORT     STATE SERVICE    REASON         VERSION
9101/tcp open  jetdirect? syn-ack ttl 53
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
9103/tcp open  jetdirect? syn-ack ttl 53
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
```

Ejemplo real

```
PORT     STATE SERVICE    REASON         VERSION
9141/tcp open  http       syn-ack ttl 53 Apache httpd 2.4.62
 ↪  ((Debian))
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20;
 ↪  withinhost=ctf.tinmarino.com
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://ctf.tinmarino.com:9141/challenge.php
|     Form id: form-upload
|     Form action: /uploadify/uploadify.php
```

Ejemplo real

```
PORT     STATE SERVICE    REASON         VERSION
9141/tcp open  http       syn-ack ttl 53 Apache httpd 2.4.62
 ↪  ((Debian))
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   /test.html: Test page
|_  /info.php: Possible information file
| http-fileupload-exploiter:
|
|     Failed to upload and execute a payload.
```

Ejemplo real

```
PORT     STATE SERVICE    REASON         VERSION
9141/tcp open  http       syn-ack ttl 53 Apache httpd 2.4.62
 ↪  ((Debian))
|_http-jsonp-detection: Couldn't find any JSONP endpoints.
|_http-litespeed-sourcecode-download: Request with null byte did
 ↪  not work. This web server might not be vulnerable
|_http-server-header: Apache/2.4.62 (Debian)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2017-1001000: ERROR: Script execution failed (use
 ↪  -d to debug)
|_http-wordpress-users: [Error] Wordpress installation was not
 ↪  found. We couldn't find wp-login.php
```

Sesión 3: Configuración

Módulo 2: Nmap)

Tipos de escaneos

Para ayuda: `nmap` o `man nmap`.

También probar la autocompletación de Bash mediante <tab>.

```
# The fastest
sudo nmap -sS -T5 -n -Pn ctf.tinmarino.com -p 9141

# All
sudo nmap -A -T5 ctf.tinmarino.com -p 9141
```

Importancia del escaneo de puertos

El escaneo de puertos es una técnica esencial en la ciberseguridad que permite a los profesionales identificar servicios activos y vulnerabilidades en un sistema.

Al igual que un mecánico utiliza herramientas específicas para diagnosticar problemas, los expertos en seguridad emplean diversas técnicas de escaneo para evaluar la seguridad de una red.

Importancia del escaneo de puertos

Con el conocimiento adecuado, los usuarios pueden seleccionar el método de escaneo más efectivo para cada situación, mejorando así la eficacia de sus pruebas de penetración.

Además, el escaneo de puertos ayuda a cartografiar la superficie de ataque de un sistema, permitiendo a los administradores de red fortalecer sus defensas.

Técnicas de escaneo de puertos

Opción	Nombre	Descripción
-sS	SYN (TCP)	Escaneo por defecto, rápido y sigiloso. No completa las conexiones TCP.
-sT	TCP Connect	Escaneo por defacto cuando no hay privilegios. Establece conexiones TCP completas.
-sU	UDP Scan	Escaneo de puertos por UDP. Más lento y complicado que TCP.

Técnicas de escaneo de puertos

Opción	Nombre	Descripción
-sA	ACK (TCP)	Escaneo de mensajes con el flag TCP ACK. Cartografía reglas de firewall. No determina puertos abiertos.
-sN	NULL (TCP)	Envía paquetes sin flags Útil para evadir detección.

Opciones de depuración

Opción	Descripción
`-v3`	Verbose
`-d3`	Verbose extremo (Debug)
`--packet-trace`	Muestra las transacciones de red
`--reason`	Explica la razón de las decisiones automáticas
`--min-rate 10000`	Envía paquetes rápidamente
`--stats-every=5s`	Muestra el progreso del escaneo cada 5 segundos

Opciones útiles

Opción	Descripción
`-oA nmap`	Escribe `nmap.gnmap`, `nmap.nmap` y `nmap.xml`
`--T5`	Timing agresivo (5 es el más alto)
`--min-rate 10000`	Envía paquetes rápidamente
`-n`	Sin búsqueda de DNS inversa
`-Pn`	Sin escaneo de ping
`-A`	All: OS, versiones, scripts y traceroute
`-p-`	Escanea todos los puertos
`--script=default,vulners`	Usa el script de vulnerabilidades

Sesión 4: Servicios

Módulo 2: Nmap)

Escaneo de servicios

```
sudo nmap -sV ctf.tinmarino.com -p 9141
```

```
Nmap scan report for ctf.tinmarino.com (54.226.26.40)
Host is up (0.14s latency).
rDNS record for 54.226.26.40:
 ↪  ec2-54-226-26-40.compute-1.amazonaws.com

PORT     STATE SERVICE VERSION
9141/tcp open  http    Apache httpd 2.4.62 ((Debian))
```

Scripts NSE

Nmap Scripting Engine (NSE) permite crear y utilizar scripts en Lua para interactuar con servicios.

```
nmap --script-help=\*
```

Categorías de scripts NSE

Categoría	Descripción
auth	Credenciales de autenticación
broadcast	Descubrimiento de hosts
brute	Fuerza bruta para iniciar sesión
default	Scripts predeterminados
discovery	Evaluación de servicios accesibles

Categorías de scripts NSE

Categoría	Descripción
dos	Verificación de DoS
exploit	Explotación de vulnerabilidades
external	Uso de servicios externos
fuzzer	Identificación de vulnerabilidades
intrusive	Scripts que afectan el sistema

Categorías de scripts NSE

Categoría	Descripción
malware	Detección de malware
safe	Scripts no intrusivos
version	Detección de versiones de servicios
vuln	Identificación de vulnerabilidades

Nmap recordar

- Ejecutar como root (`sudo nmap`, ya que Nmap es inteligente y puede enviar paquetes en bruto como root.
- Realiza un escaneo rápido y uno lento para comenzar a trabajar temprano.
- Cartografíar huéspedes, puertos y firewalls desde su lado (evitar escanear todos los puertos en todos los huéspedes con `-Pn`).
- Leer la documentación (`man nmap`) ya que la herramienta es potente y la documentación muy instructiva.

Nuclei el hermano menor de Nmap

Nuclei es una herramienta de escaneo de vulnerabilidades.

Utiliza plantillas para detectar problemas de seguridad en aplicaciones web y servicios.

Uso básico de Nuclei

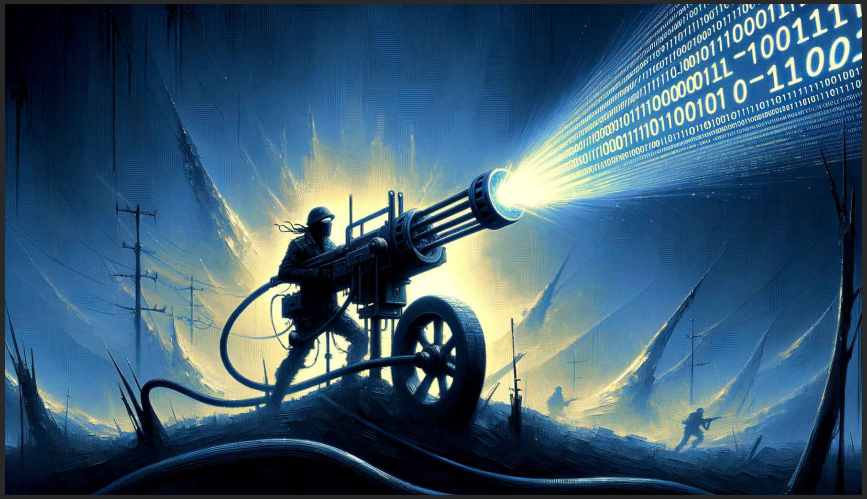
```
nuclei -u https://www.ejemplo.com
nuclei -l lista_de_urls.txt -o resultados.txt
```

```
Iniciando Nuclei 2.0 (https://nuclei.projectdiscovery.io) a las
 ↪  2025-04-06 12:35 -04
Informe de escaneo para www.ejemplo.com
Vulnerabilidades encontradas:
- XSS en /pagina
- Inyección SQL en /api
Nuclei finalizado: 2 vulnerabilidades encontradas en 5.42
 ↪  segundos
```

Nuclei Vs Nmap

Característica	Nuclei	Nmap
Propósito	Escaneo de vulnerabilidades	Escaneo de puertos y servicios
Tipo de Escaneo	Basado en plantillas	Independiente
Salida	Detalles de vulnerabilidades	Detalles de puertos y servicios
Uso	Evaluación de seguridad	Cartografía de red

Módulo 3: FFuF



Módulo 3: FFuF

S	Nombre	Descripción
1	Rutas	Exploración de caminos de URL
2	Dominios	Ataques de fuerza bruta para subdominios
3	Parámetros	Intento y error de variables y valores
4	Técnicas	Métodos para generar diccionarios

FFuF

FFuF es una herramienta de Fuzzing[1] HTTP para:

- Directorios
- Archivos y extensiones
- VHosts
- Nombres de parámetros
- Valores de parámetros

[1] El Fuzzing es una técnica que envía datos aleatorios a aplicaciones para detectar vulnerabilidades y errores. (Viene del ingles «fuzzy» que significa «turbio»)

Sesión 1: Rutas

(Módulo 3: FFuF)

Directorios, archivos, extensiones

```
ffuf -w list.txt -u http://SERVER_IP:PORT/FUZZ

ffuf -w list.txt -u http://SERVER_IP:PORT/FUZZ \
  -recursion \
  -e .php
```

Sesión 2: Dominios

(Módulo 3: FFuF)

Subdominios

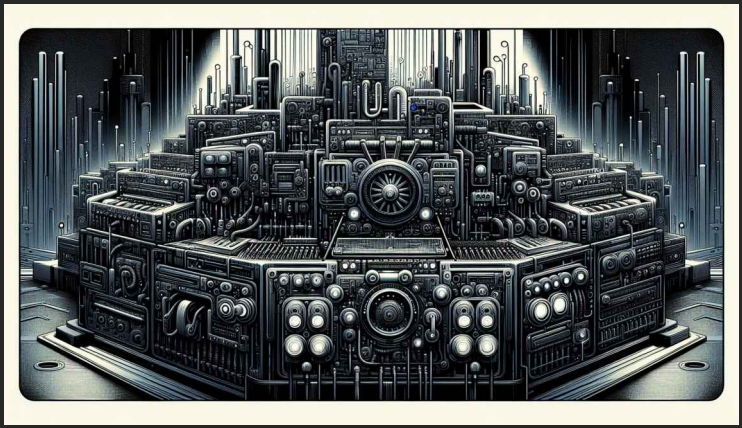
```
ffuf -w ~/SecList/DNS/subdomains:FUZZ -u
 ↪  https://FUZZ.tinmarino.com/
```

Equipos virtuales y cabeceras

```
# VHost
ffuf -w list.txt -u http://ctf.tinmarino.com/ \
  -H 'HOST: FUZZ.tinmarino.com'

# X-Forwarded-For
ffuf -w cabecera.txt -u http://ctf.tinmarino.com/ \
  -H 'FUZZ: c2c.tinmarino.com'
```

Sesión 3: Parámetros

(Módulo 3: FFuF)



Matcher

```
-mc                 Match HTTP status codes, or "all" for
 ↪  everything. (default: 200-299,301,302,307,401,403,405,500)
-ml                 Match amount of lines in response
-mmode              Matcher set operator. Either of: and, or
 ↪  (default: or)
-mr                 Match regexp
-ms                 Match HTTP response size
-mt                 Match how many milliseconds to the first
 ↪  response byte, either greater or less than. EG: >100 or <100
-mw                 Match amount of words in response
```

Filter

```
-fc                 Filter HTTP status codes from response.
 ↪  Comma separated list of codes and ranges
-fl                 Filter by amount of lines in response. Comma
 ↪  separated list of line counts and ranges
-fmode              Filter set operator. Either of: and, or
 ↪  (default: or)
-fr                 Filter regexp
-fs                 Filter HTTP response size. Comma separated
 ↪  list of sizes and ranges
-ft                 Filter by number of milliseconds to the
 ↪  first response byte, either greater or less than. EG: >100
 ↪  or <100
-fw                 Filter by amount of words in response. Comma
 ↪  separated list of word counts and ranges
```

Combinaciones de listas

```
ffuf \
  -w ~/seclists/../common.txt:FUZZ_1 \
  -w ~/seclists/../web-extensions.txt:FUZZ_2 \
  -t 200 -rate 10000 \
  -u http://ctf.tinmarino.com:9141/FUZZ_1.FUZZ_2
```

Solicitud en archivo

```
ffuf -w list.txt -request req1.txt
```

Cuidado con el **Content-Length**

POST Content-Type

Tipo de Contenido	Descripción
text/plain	Texto plano
application/json	JSON
application/xml	XML
application/x-www-form-urlencoded	Formularios
multipart/form-data	Archivos
application/octet-stream	Binarios
image/jpeg	Imagen JPEG

Sesión 4: Técnicas

(Módulo 3: FFuF)

Generar diccionario localmente

```
printf "%d\n" {1..1000}  # Number
printf '%s\n' {A..Z}  # Letter upper case
printf '%b' $(printf '\\x%x\\x0A' {32..126})  # Character
printf "%%%02X\n" {0..255}  # URL encoding

# Copiar rápido
alias x='xsel --input --clipboard'

git clone --depth=10 https://github.com/danielmiessler/SecLists
```

Generar diccionario de la aplicación

Primero descargar todo el sitio web.

```
wget -k -p -r -c https://www.tinmarino.com
```

Generar diccionario de la aplicación

Después utilizar expresiones regulares para obtener rutas en *strings*.

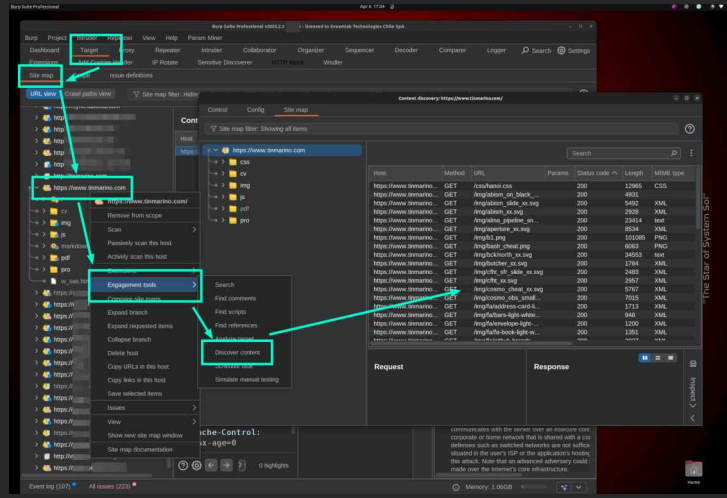
```
rg '"([.?/a-z0-9_-]*)"' -r '$1' \
  -NoI \
  -g '*.{js,html}' \
  index.html js/pro.js \
  | sed 's|/|\r|g' \
  | sort -u \
  > /tmp/tin.dic
```

Generar diccionario de la aplicación

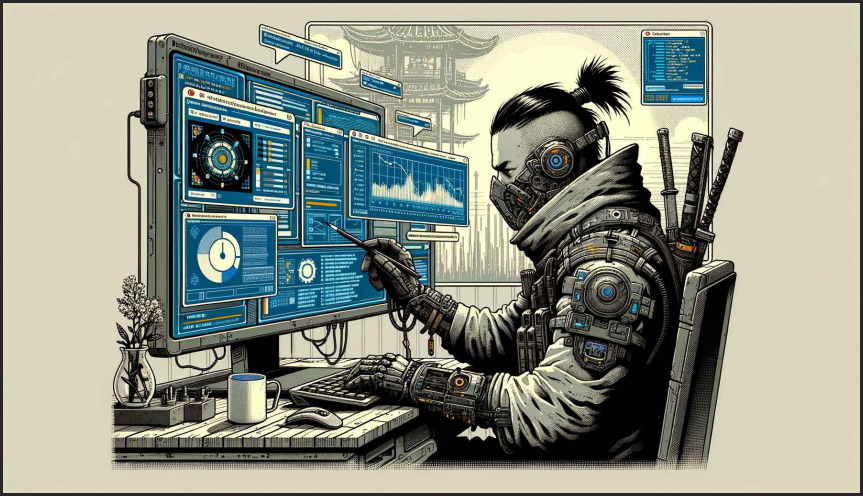
Finalmente buscar más páginas escondidas.

```
ffuf -w /tmp/tin.dic \
  -t 200 -rate 10000 \
  -recursion \
  -e md,html,pdf \
  -u https://www.tinmarino.com/FUZZ
```

Descubrir contenido con BurpSuite



Módulo 4 BurpSuite



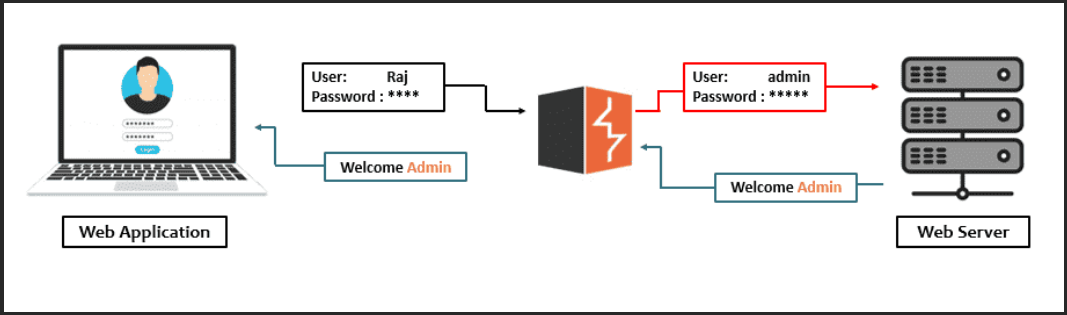
Módulo 4 BurpSuite

S	Nombre	Descripción
1	Proxy	Interceptor de tráfico HTTP
2	Escáneres	Herramientas automáticas para analizar sitios o URL
3	Herramientas	Utilidades de la interfaz gráfica
4	Extensiones	Complementos útiles

Sesión 1: Proxy

(Módulo 4: BurpSuite)

Que es un proxy web?



(Imagen de Raj Chandel)

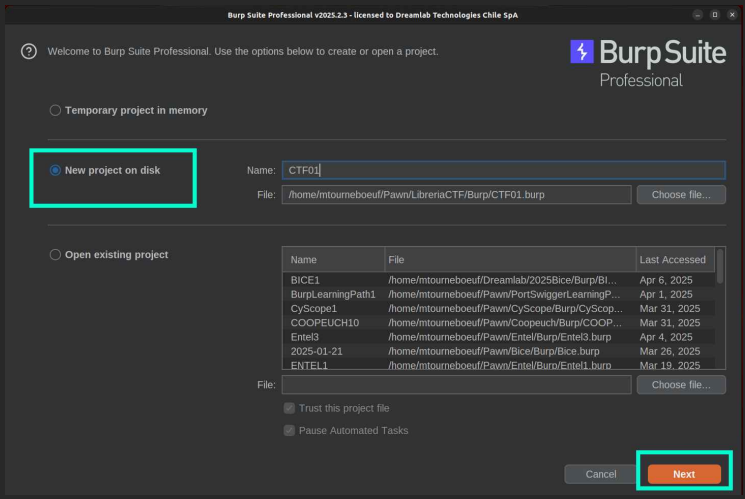
Que es un proxy web?

Los proxies web actúan como intermediarios entre el navegador y el servidor, capturando solicitudes.

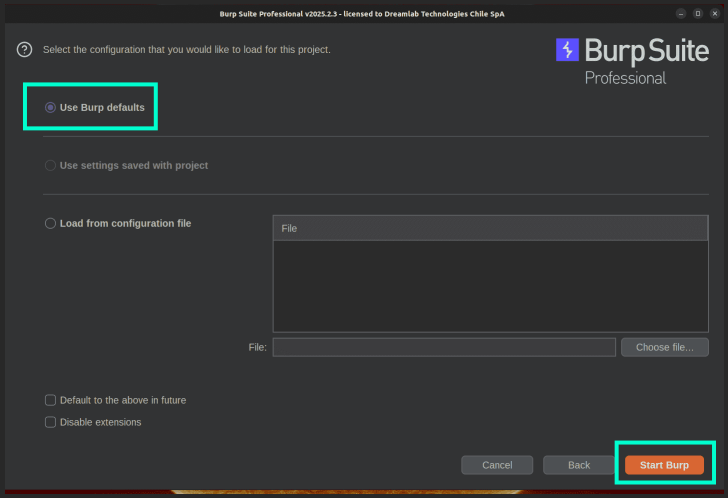
De que sirve un proxy web?

- **Capturar** solicitudes HTTP.
- **Modificar** las solicitudes capturadas.
- **Enviar** solicitudes arbitrarias (fuzz).
- **Análisar** el tráfico web.
- **Escanear** vulnerabilidades en aplicaciones web.
- **Cartogafiar** la arquitectura de aplicaciones web.

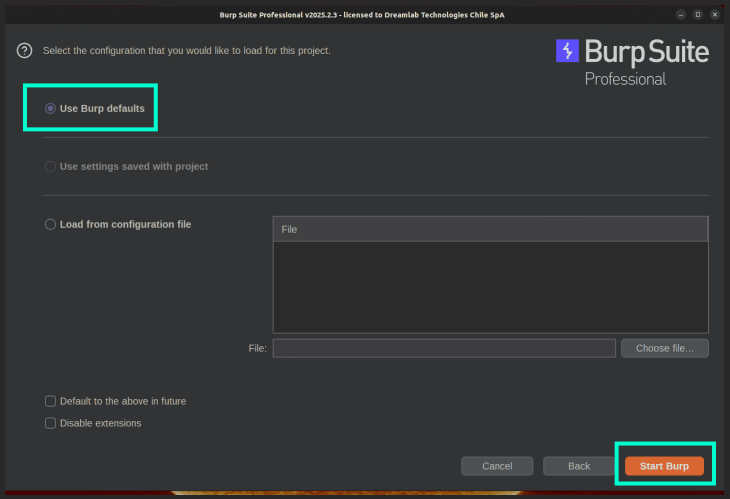
Abrir BurpSuite



Abrir BurpSuite



Abrir el navegador de BurpSuite



Crear un servicio HTTP local

```php
<?php
header('Content-Type: application/json');
echo json_encode([
    'method' => $_SERVER['REQUEST_METHOD'],
    'headers' => getallheaders(),
    'body' => file_get_contents('php://input')
]);
?>
```

```
php -S localhost:8000
```

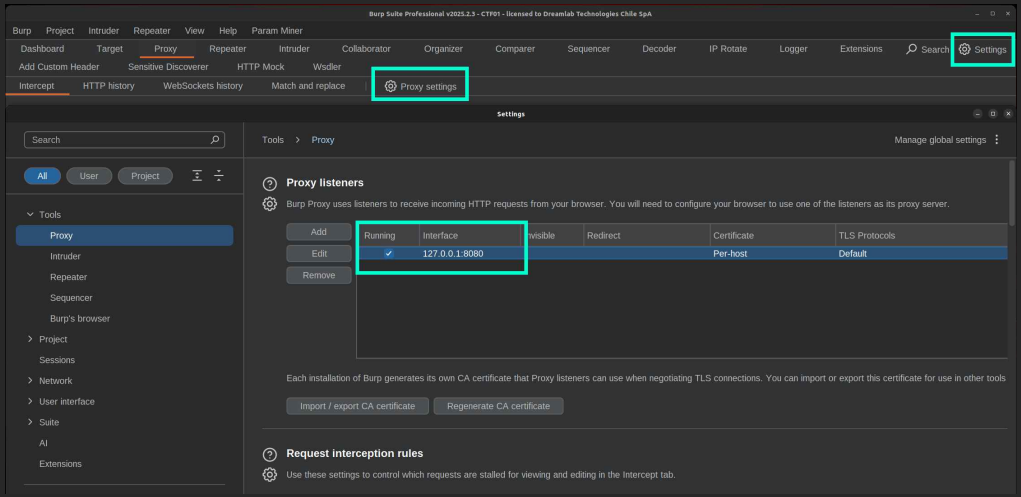
Visitar el sitio

- http://localhost:8000
- http://127.0.0.1:8000
- http://[::1]:8000

Usos de BurpSuite

1. Interceptar solicitudes (Proxy)
2. Interceptar respuestas (Proxy)
3. Remplazar texto automáticamente (Proxy / Match and Replace)
4. Repetir solicitudes (Repeater)
5. Infiltrarse (Intruder)

Configuración del proxy



Sesión 2: Scanner

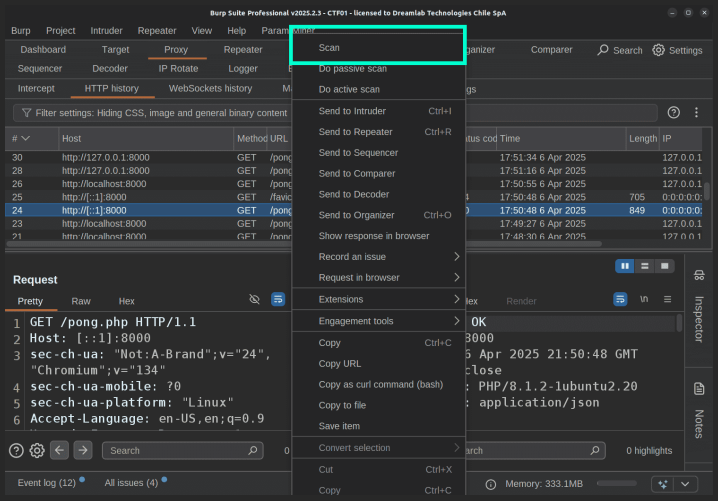
(Módulo 4: BurpSuite)

Scanners

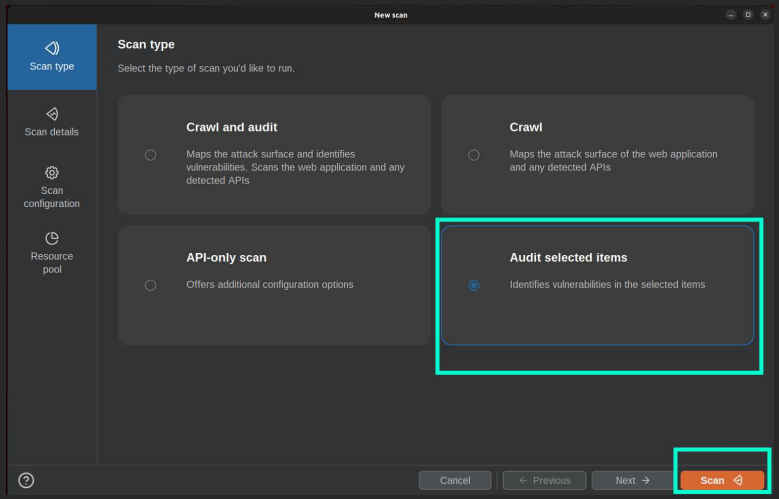
Para iniciar un escaneo en BurpSuite, tenemos las siguientes opciones:

1. Iniciar un escaneo en una solicitud específica del Historial del Proxy.
2. Iniciar un nuevo escaneo en un conjunto de objetivos.
3. Iniciar un escaneo en elementos dentro del alcance.

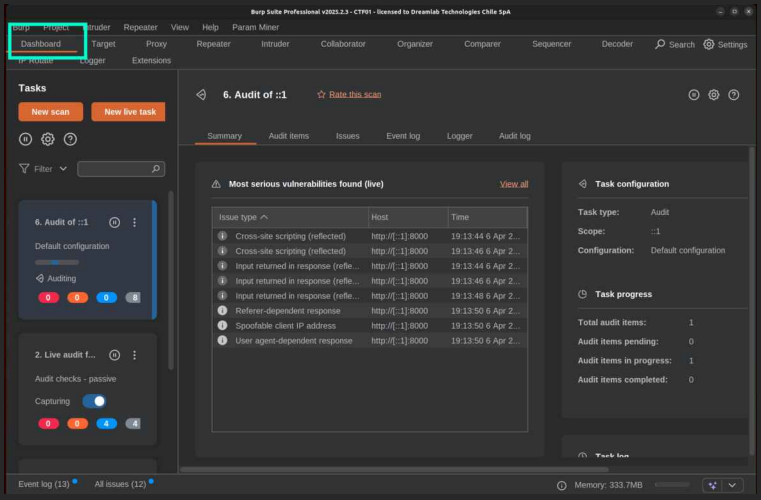
Escaneo de *endpoint*



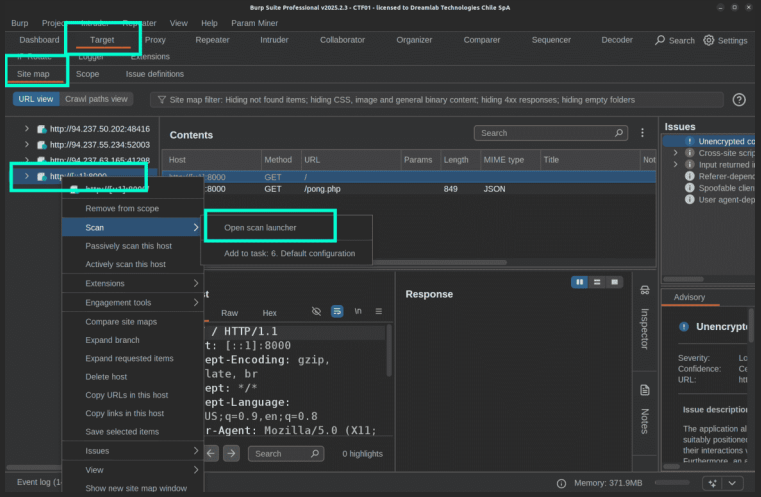
Escaneo de *endpoint*



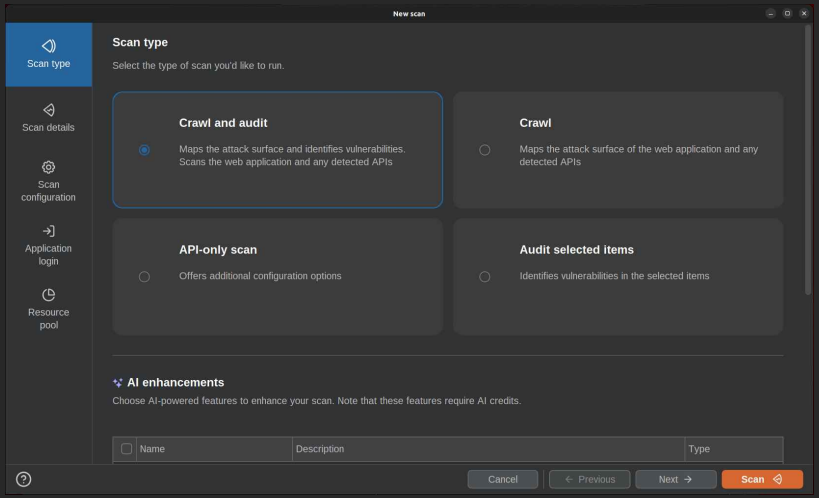
Escaneo de *endpoint*



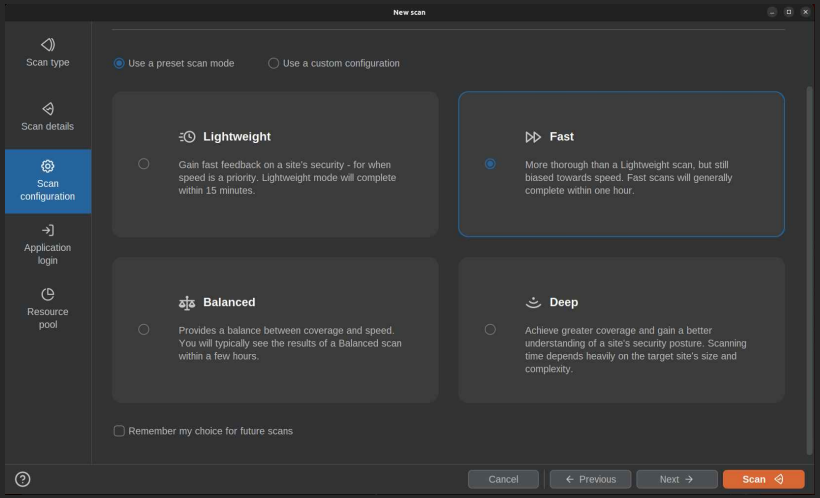
Escaneo de sitios



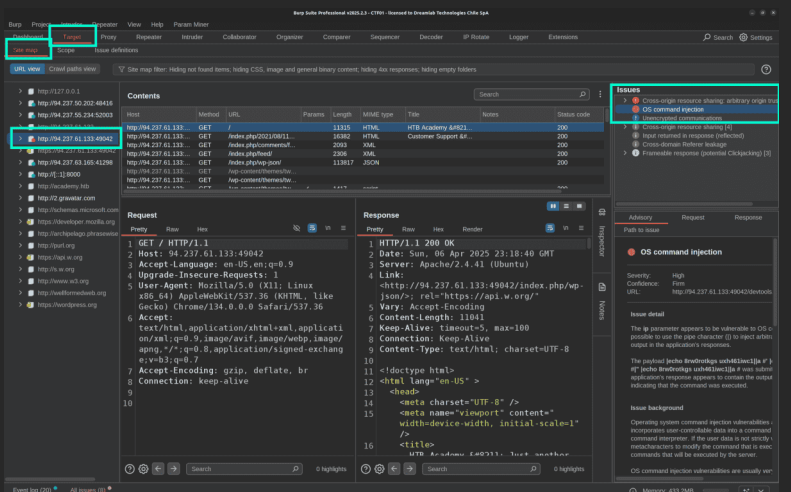
Escaneo de sitios



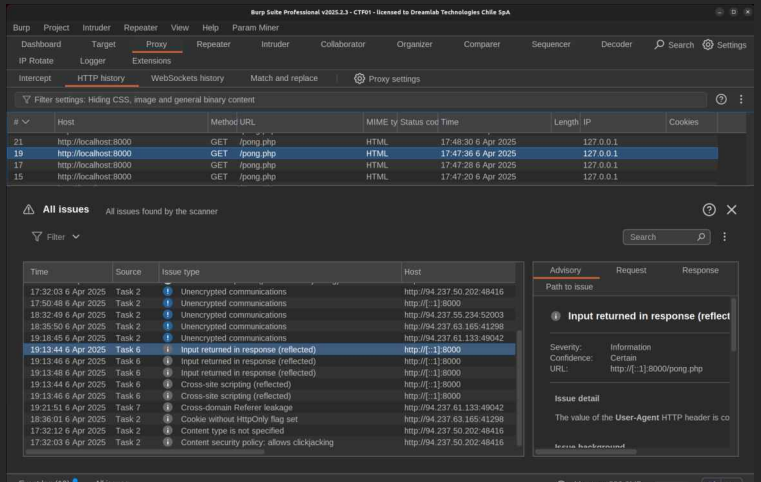
Escaneo de sitios



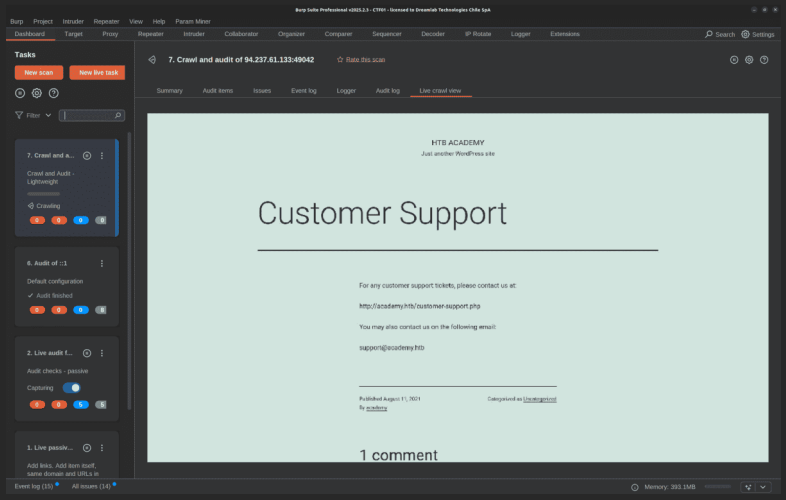
Vulnerabilidad encontrada



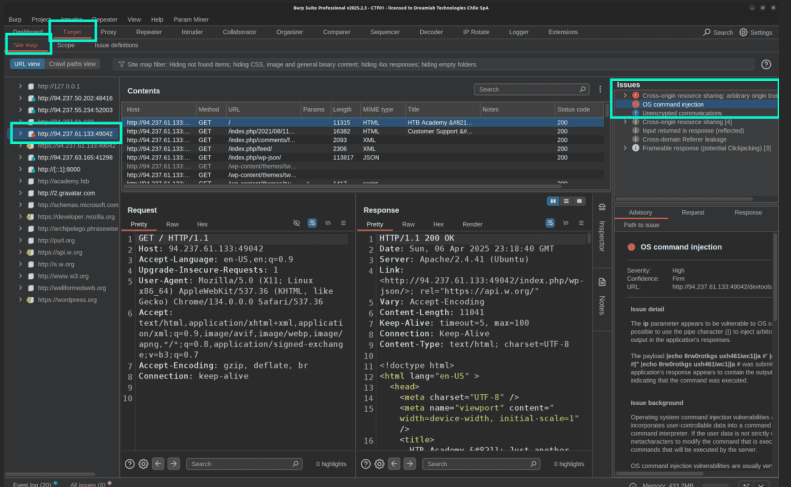
Escaneo pasivo



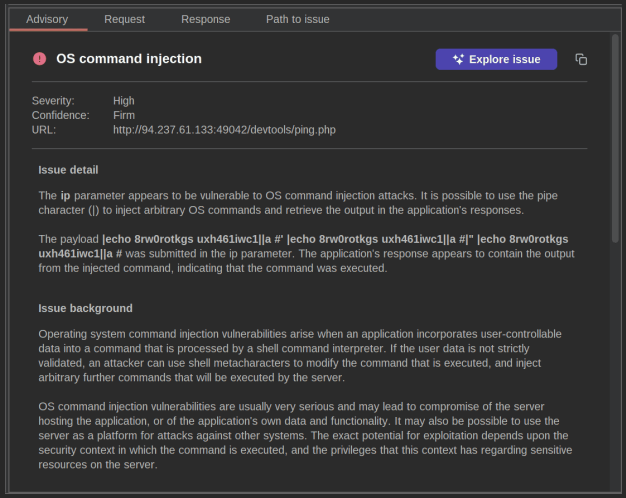
BurpSuite Crawler



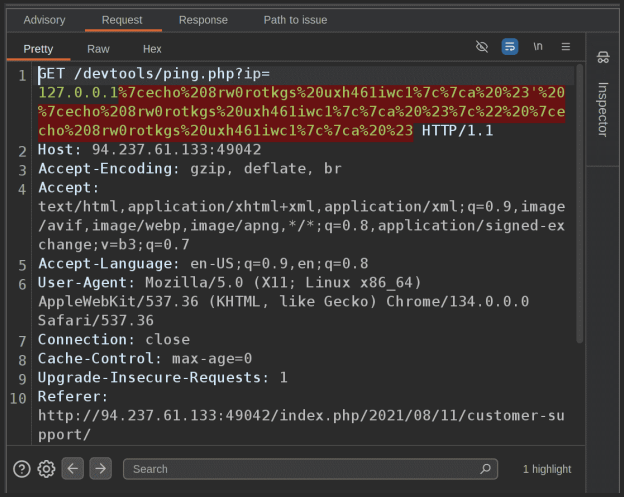
Vulnerabilidad encontrada



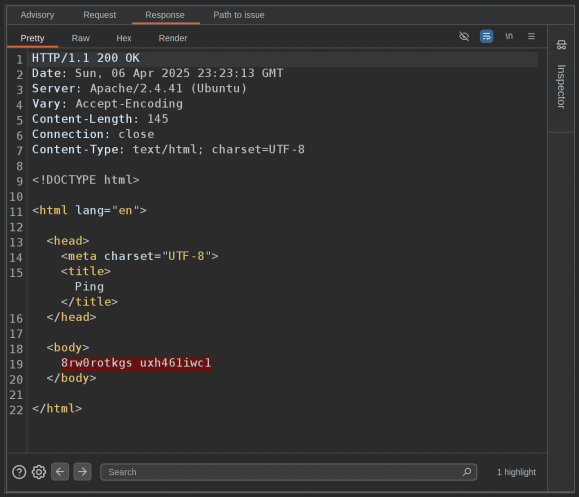
Reproducir vulnerabilidad



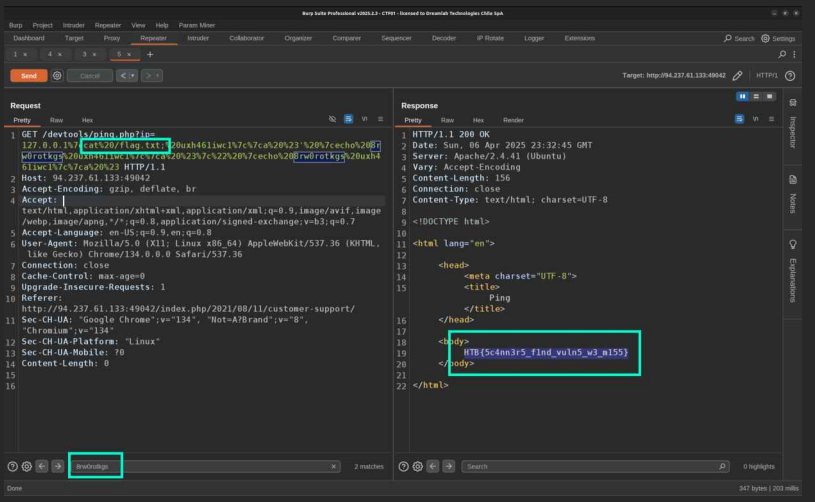
Reproducir vulnerabilidad



Reproducir vulnerabilidad



Reproducir vulnerabilidad



Reproducir vulnerabilidad

Advisory Request Response Path to issue

Path to location of Request

Step	Action	Destination URL
1	Requested http://94.237.61.133:49042/	http://94.237.61.133:49042/
2	Clicked "Customer Support"	http://94.237.61.133:49042/index.php/2021/08/11/c
3	Clicked "/ping.php?ip=127.0.0.1"	http://94.237.61.133:49042/devtools/ping.php?ip=1

Reflexión final sobre escáneres

No olvidar detener los escaneos automaticos!

Sesión 3: Herramientas

(Módulo 4: BurpSuite)

Usos avanzados de BurpSuite

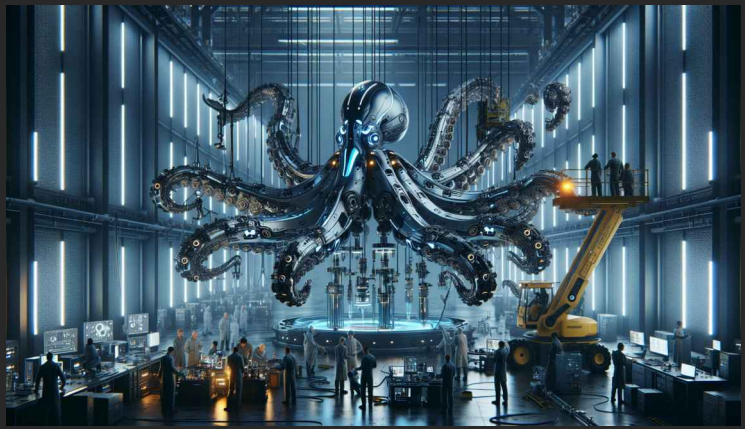
1. Collaborator
2. Organizer
3. Comprarer
4. Search
5. Crawler
6. Scanner

Herramientas que permiten proxy

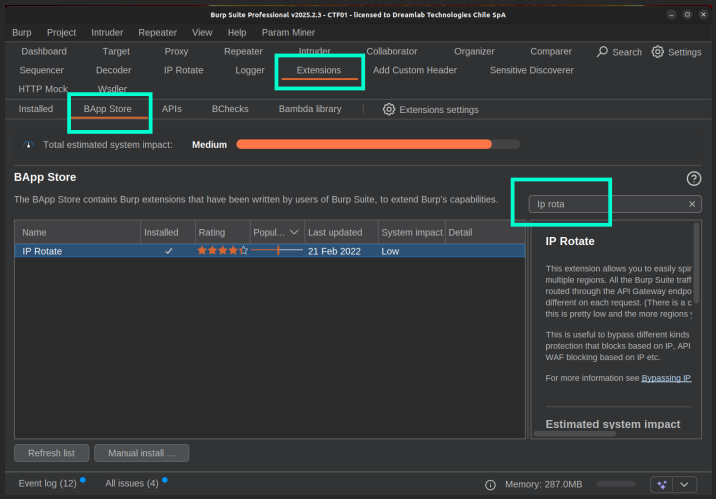
```
proxychains curl http://www.tinmarino.com:9001  # sudo vim
 ↪  /etc/proxychains.conf

nmap --proxies http://127.0.0.1:8080 www.tinmarino.com -pPORT
 ↪  -Pn -sC

msfconsole  # set PROXIES HTTP:127.0.0.1:8080
```

Sesión 4: Extensiones

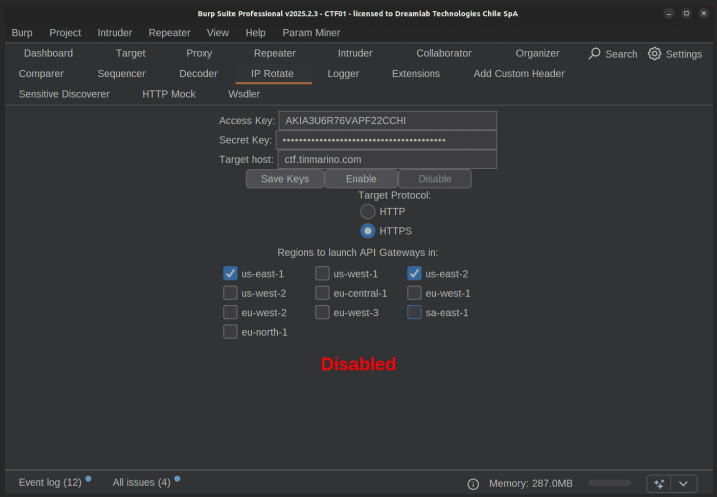
(Módulo 4: BurpSuite)



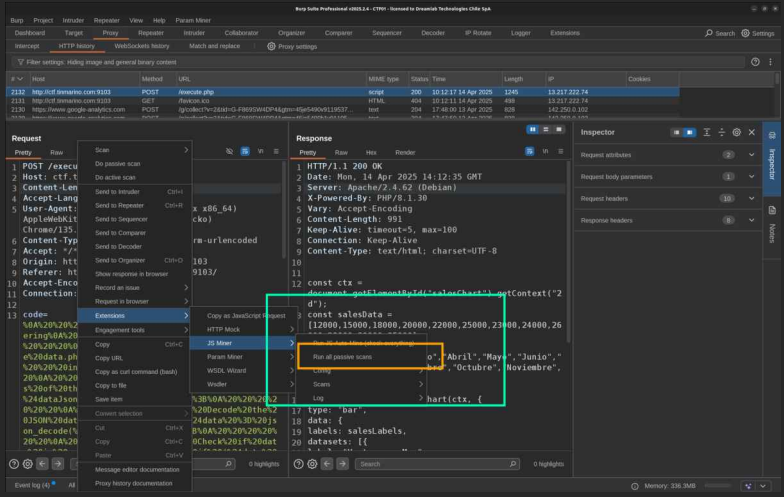
Bapp Store: Instalar extensiones



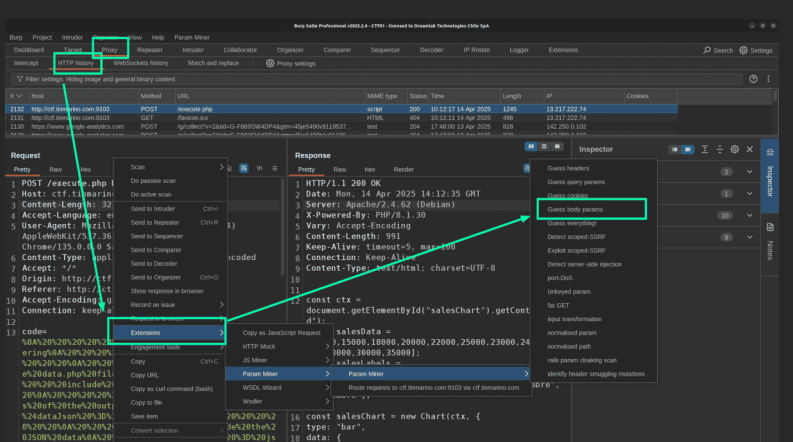
Ip Rotate


Burp Suite Professional v2025.2.3 - CTF01 - licensed to Dreamlab Technologies Chile SpA
Burp Project Intruder Repeater View Help Param Miner
Dashboard Target Proxy Repeater Intruder Collaborator Organizer Search Settings
Comparer Sequencer Decoder IP Rotate Logger Extensions Add Custom Header
Sensitive Discoverer HTTP Mock Wsdler
Access Key: AKIA3U6R76VAPF22CCHI
Secret Key: ••
Target host: ctf.tinmarino.com
Save Keys Enable Disable
Target Protocol:
HTTP
HTTPS
Regions to launch API Gateways in:
us-east-1 us-west-1 us-east-2
us-west-2 eu-central-1 eu-west-1
eu-west-2 eu-west-3 sa-east-1
eu-north-1
Disabled
Event log (12) All issues (4) Memory: 287.0MB

JS Miner


Burp Suite Professional v2025.2.4 - CTF01 - licensed to Dreamlab Technologies Chile SpA
Request
Response
Inspector
Scan
Do passive scan
Do active scan
Send to Intruder
Send to Repeater
Send to Sequencer
Send to Comparer
Send to Decoder
Send to Organizer
Show response in browser
Request an issue
Request in browser
Extensions
Engagement tools
Copy
Copy URL
Copy as curl command (bash)
Copy to file
Save item
Convert selection
Cut
Copy
Paste
Message editor documentation
Proxy history documentation
Copy as JavaScript request
HTTP Mock
JS Miner
Param Miner
WSDL Wizard
Wsdler
Run all passive scans
Config
Scans
Log
HTTP/1.1 200 OK
Date: Mon, 14 Apr 2025 14:12:35 GMT
Server: Apache/2.4.62 (Debian)
X-Powered-By: PHP/8.1.30
Vary: Accept-Encoding
Content-Length: 991
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
const ctx =
const salesData =
type: 'bar',
data: {
labels: salesLabels,
datasets: [{
Memory: 336.3MB

Param Miner


Param Miner
Request
Response
Inspector
Guess body params
Extensions
Param Miner