Secure Operating Systems – Final Project Report

Submitted to Professor Ido Dubrawsky
ENPM695-Secure Operating Systems, Spring 2016
By: Apoorvasaraswati Naik (UID:115034327)
Radhika Pai (UID: 114924131)

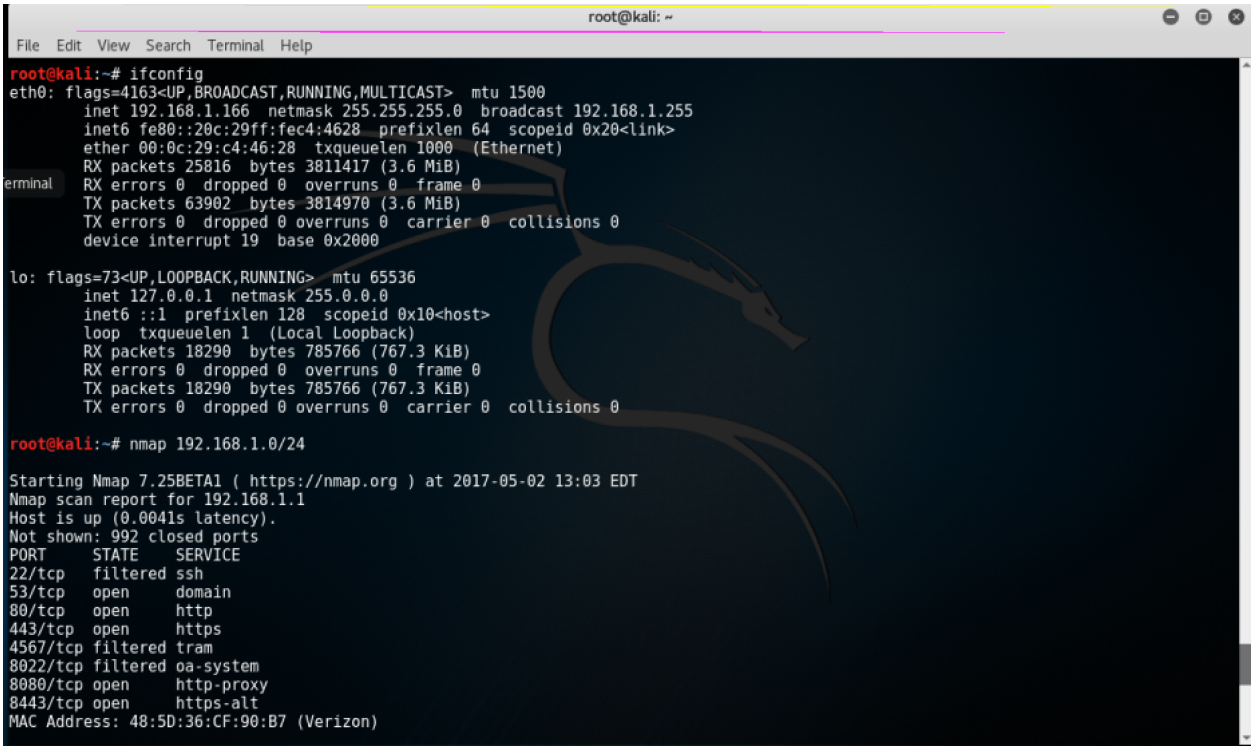
Task Group 1: Evaluate the security of the system

This task group includes evaluating the security of the system from both the outside as well as the inside. To accomplish this goal, the student must be able to determine what services are running on the system, what are their vulnerabilities, and how resilient the system is to intrusion. The following tasks are meant to help the student accomplish the goals of this task group:

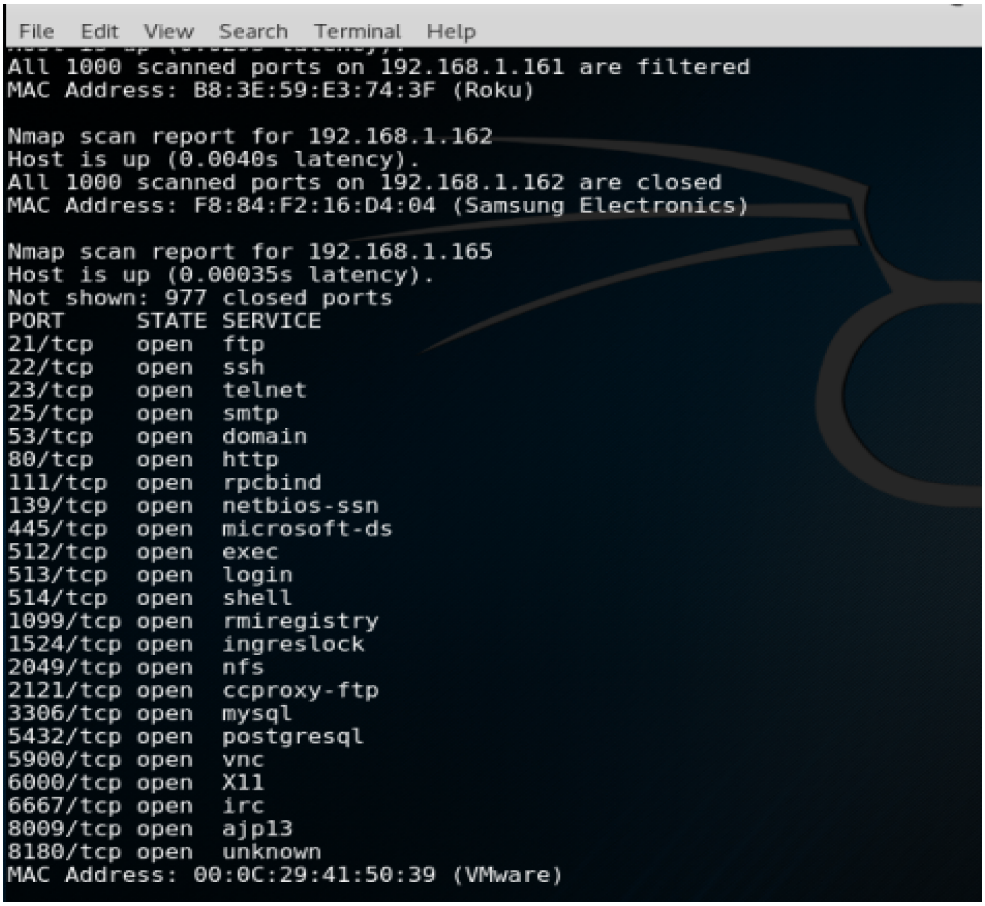
*In this project, we have exploited the ENPM695 VM in two ways using **metasploit.** One way is through exploiting **vsftpd service running on port 21.** Another exploit uses **telnet using 1524 port.***
*With the first exploit, we were able to achieve **Group 1 Tasks**. With second exploit, we completed **Group2 Tasks.***

Task 1: Determine the running and open services on the system (10 points) – points are awarded for identifying all of the open and running services as well as explaining how the information was gathered.

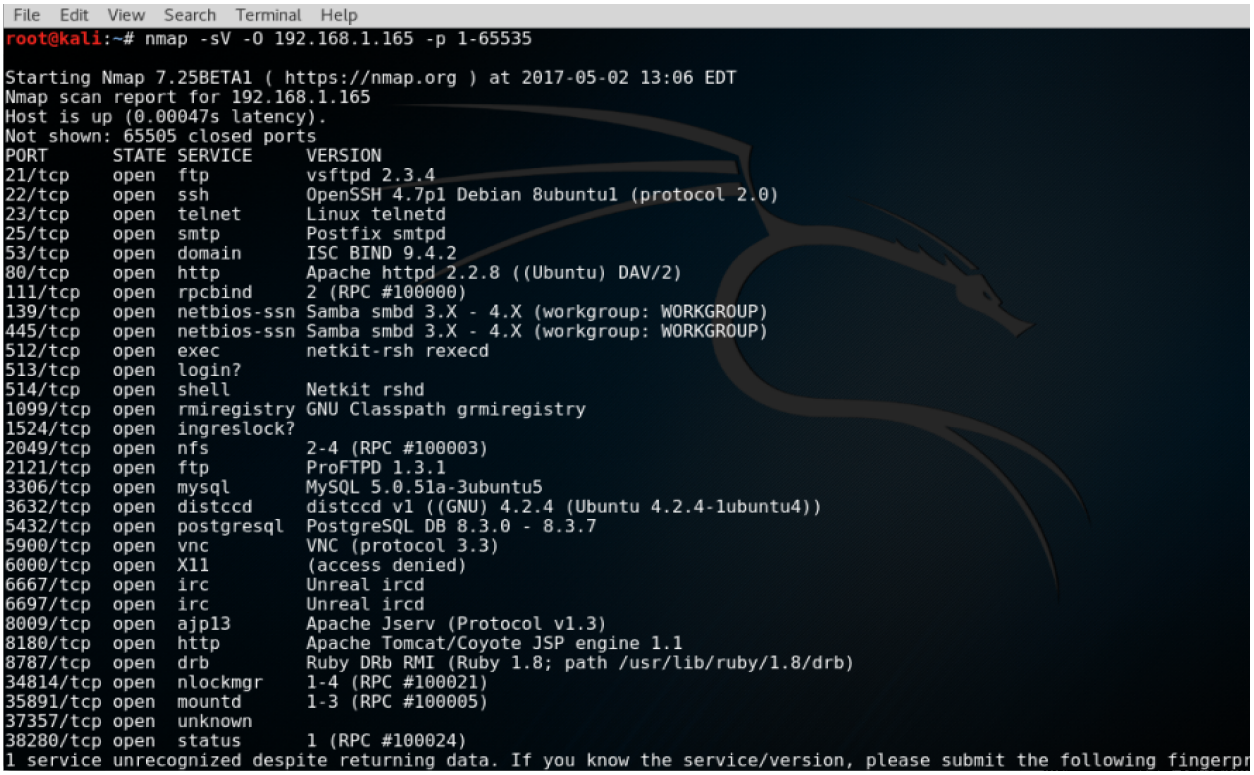
Ifconfig to figure out what is the ip of Kali and then nmap to figure out the ip of linux vm. Command used to gather information is "*ifconfig*" to get ip of Kali and *"nmap 192.168.1.0/24"*



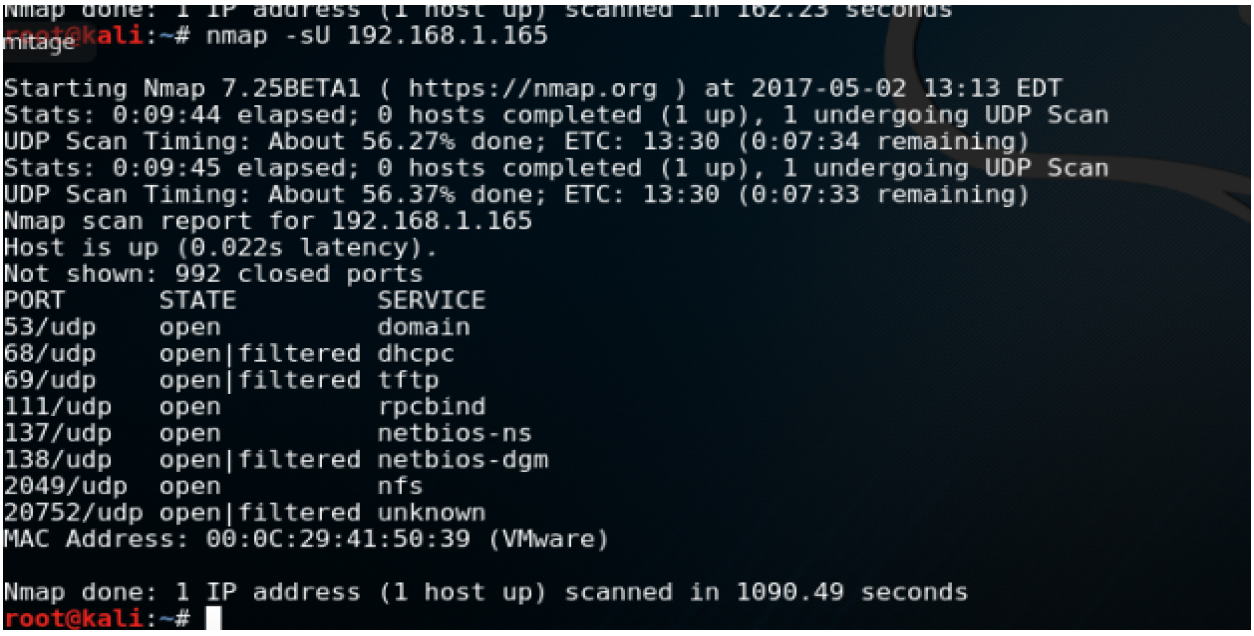
nmap scan report showing ip of linux vm i.e 192.168.1.165



nmap to find TCP open ports. Command used to gather information is *"nmap –sV –O 192.168.1.165 –p 1-65535".*
The screenshot shows the open port and the services running



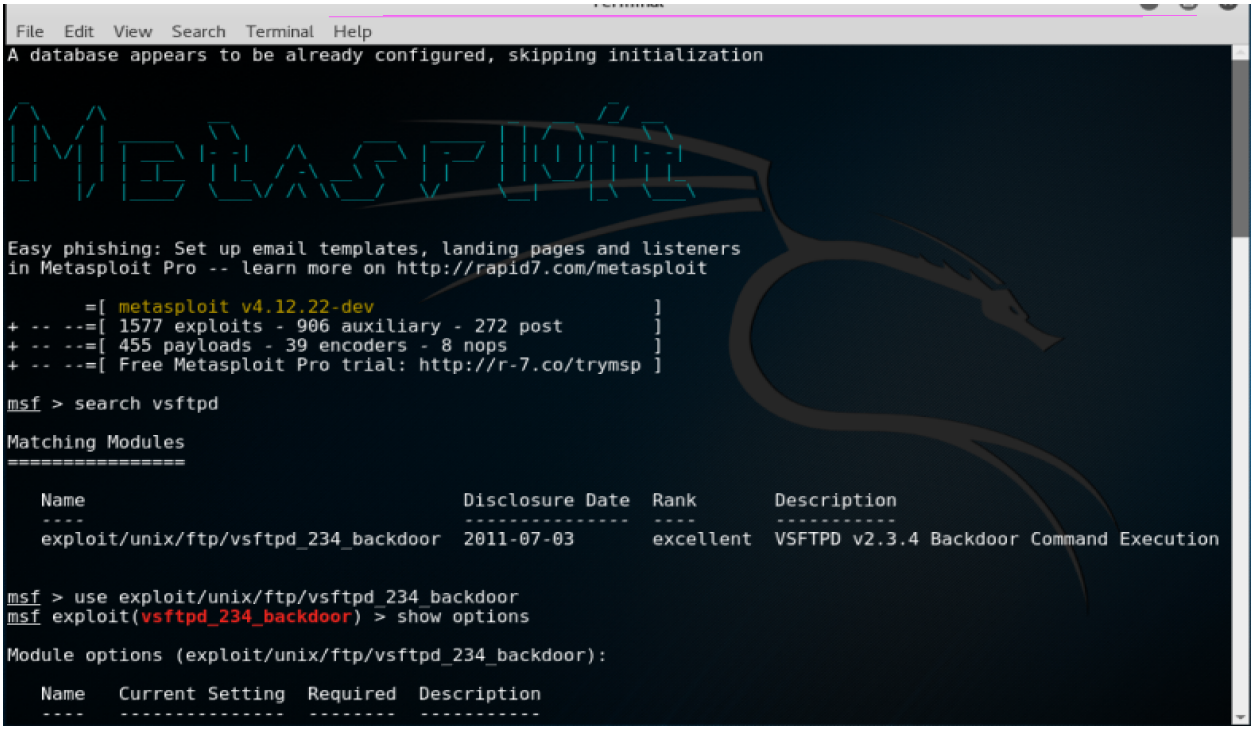
nmap to find UDP open ports, the command used is *"nmap –sV –O 192.168.1.165 –p 1-65535".*
The open port and running services are captured in the screenshot.



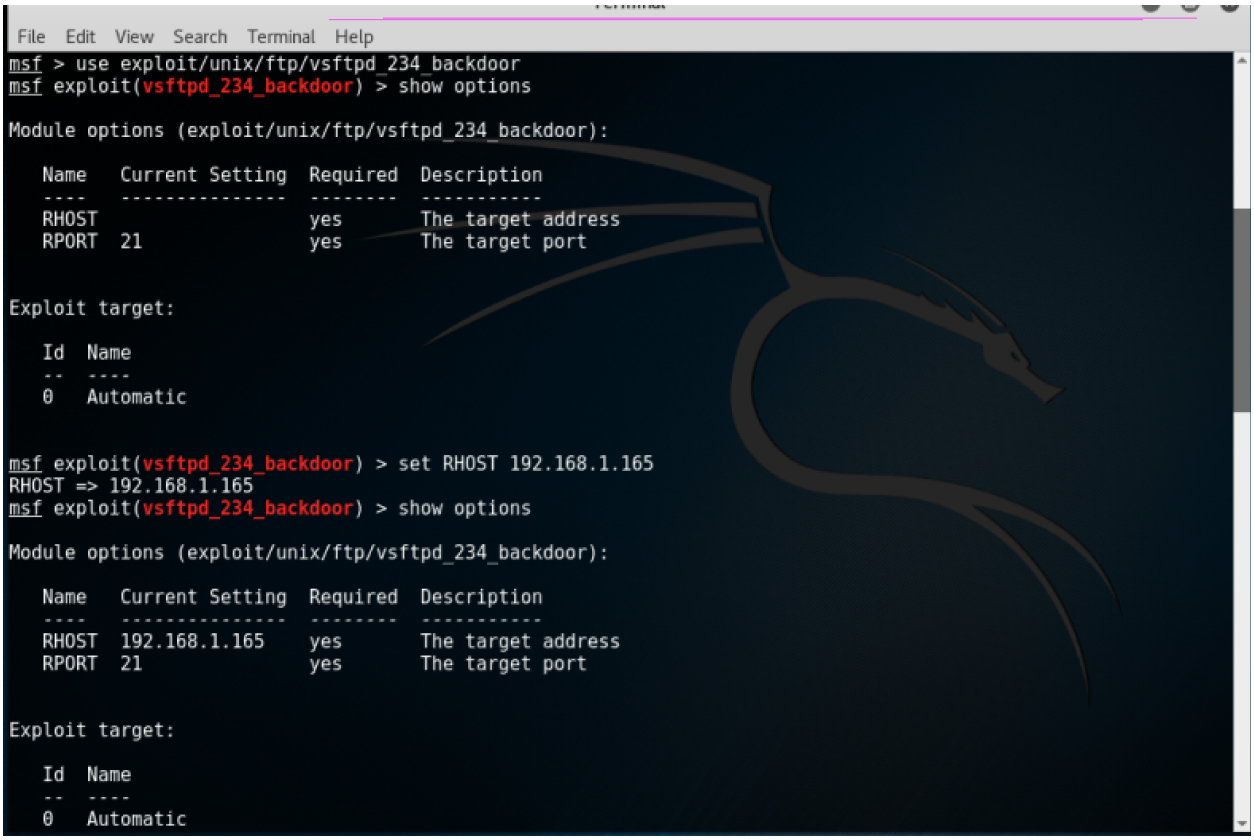
Task 2: Access the system by exploiting a vulnerable running or open service (10 points) – points are awarded based on a description of how the service was exploited as well as the tools used.

We can exploit the vulnerability by using **metasploit** from kali.

From the screenshots, it is evident that port 21/tcp was open with service vsftpd running on it. The below screenshots show that how this open service is exploited using metasploit. Open metasploit and type ***search vsftpd*** which shows that there is a backdoor and can exploited.
Next command is ***use exploit/unix/ftp/vsftpd_234_backdoor*** and followed by ***show options***

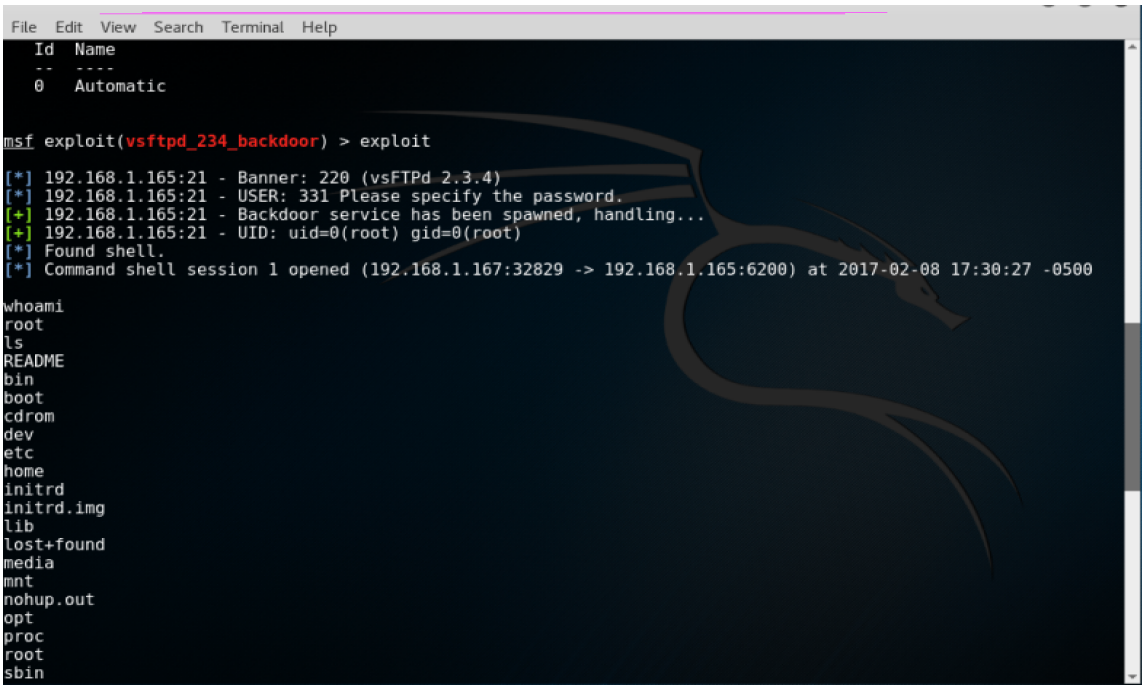


Since there is no RHOST set, set RHOST as the linux vm to exploit it by command ***set RHOST 192.168.1.165*** followed by show options to see whether RHOST is set.

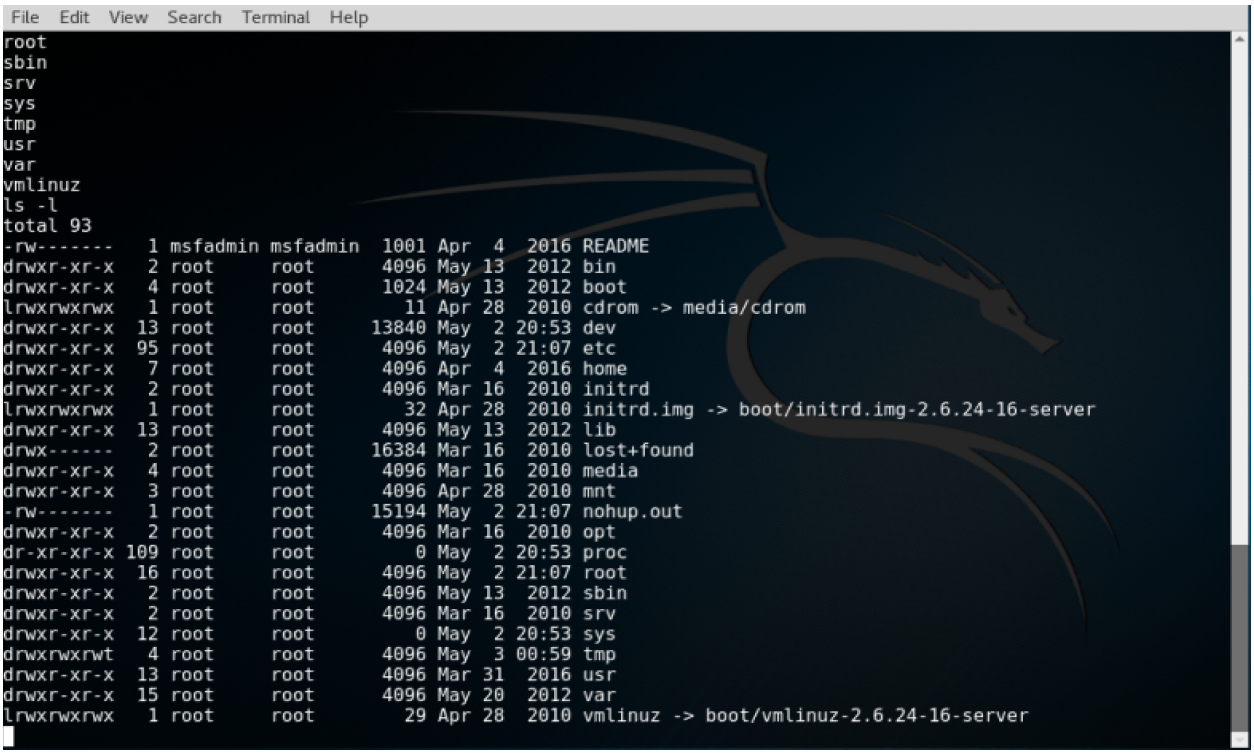


After setting RHOST as linux vm ip, type command ***exploit*** to exploit the vm.

Confirmed the exploit using commands such as whoami, ls and ls-l.



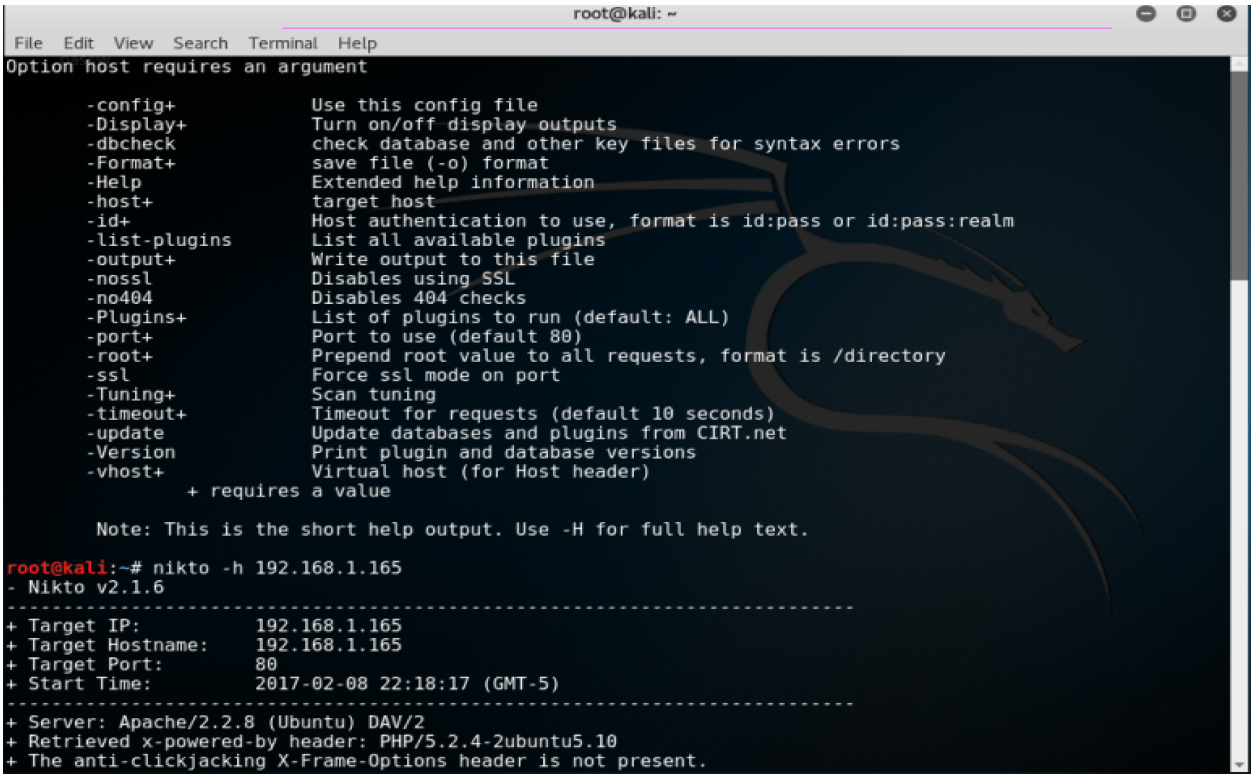
Continued exploit...



Task 3: Detail the flaws in the web server running on the system (10 points) – points are awarded based on detailing five flaws in the web server .

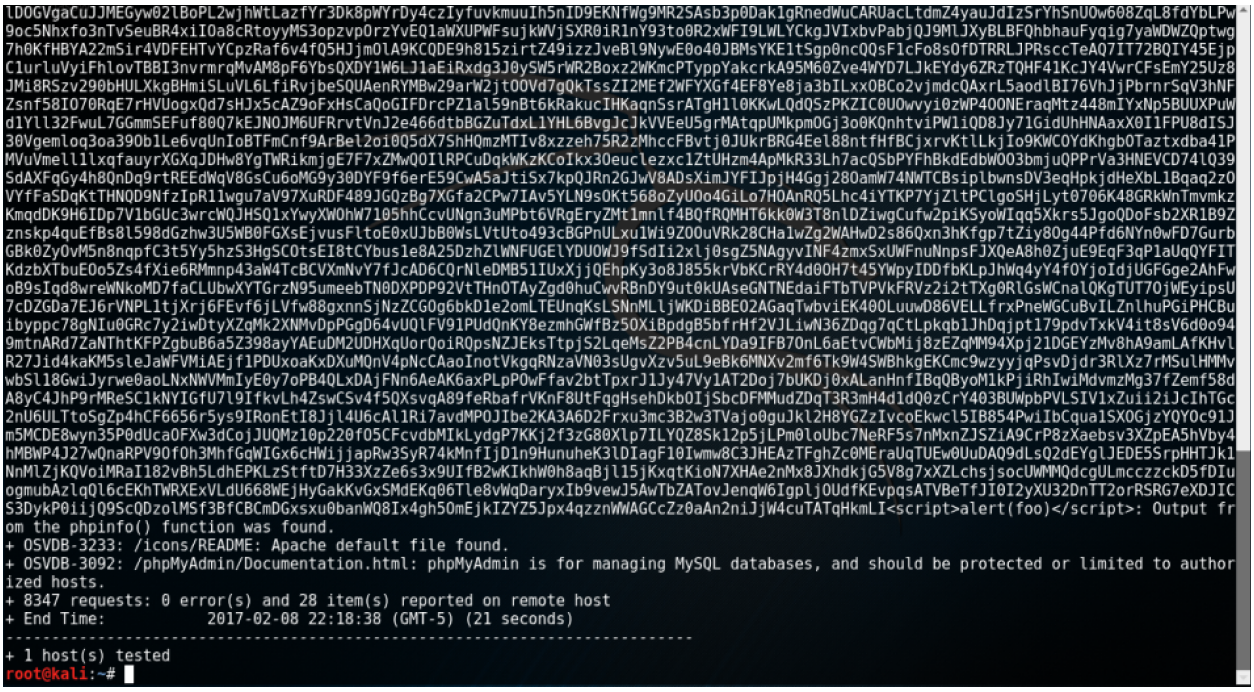
1. The Apache web server is running on port 80 which allows web traffic to flow unencrypted and attackers can easily intercept the information on the network via man-in-middle-attacks.
2. It may also invite number of attackers specially the ones trying to brute force user credentials by probing IPs on port 22 which is the default listening port for SSH. If port 22 would respond a brute force attack may appear. Despite its wide acceptance, there are still threats and occasionally software vulnerabilities associated with using SSH. For example, common libraries used by many implementations of SSH – like OpenSSL – may be reported. The mere fact that an SSH server is running and accessible from the Internet will invite attacks.
3. Open port 21 – FTP is vulnerable and allows remote attackers to cause a denial of service. It is also prone to attacks by Trojan horses/backdoors like Blade runner, Doly Trojan, Invisible FTP to name a few.
4. Open port 23 runs Telnet which used for remote maintenance of many networking communications devices including routers and switches, and often provides access to remote system with admin privileges. Given access to a server, or a network router of a corporate network or ISP, an attacker can perform a great deal of mischief. The level of access provided by telnet makes it a valuable commodity for individuals attempting to gain unauthorized access to systems or networks. This makes port 23 a very common target of attackers during network scans and reconnaissance attempts. Trojans that use port 23 include ADM worm, Fire Hacker, Telnet Pro to name a few.
5. The web server has web pages that provide access to directories without user authentication. If there is any sensitive information stored in the directories it can be misused by the attacker.
6. Attacker can create a backdoor and get access to the system, thus it is vulnerable to backdoor attacks.
7. A webserver page provides information regarding the name/version of the webserver and the port on which it is running. This information can be used by an attacker to find the loopholes and hack the system based on the vulnerabilities the server/version and the open port has.
8. Apache server in general is a large surface for attack via the vulnerabilities it has like click-jacking attacks, CSRF attacks, XSS attacks, stack based overflow attacks, DOS attacks, input validation attacks, SQL injection attacks, impersonation attacks, URL interpretation attacks to name a few.

9. A hacker can easily gain access to the system using easy passwords like ‘passw0rd’ within few attempts.
10. From the below scan using Nikto, we can see that the web server is vulnerable to click-jacking attacks due to absence of X-frame options header.
11. It is also vulnerable to XSS attacks due to absence of XSS protection header.
12. Moreover, Apache/2.2.8 being used appears to be outdated as the current versions in use are atleast Apache/2.4.12 which is comparatively less vulnerable to attacks.
13. The server leaks inodes via Etags with header found with file /phpMyAdmin/ChangeLog.
14. Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.



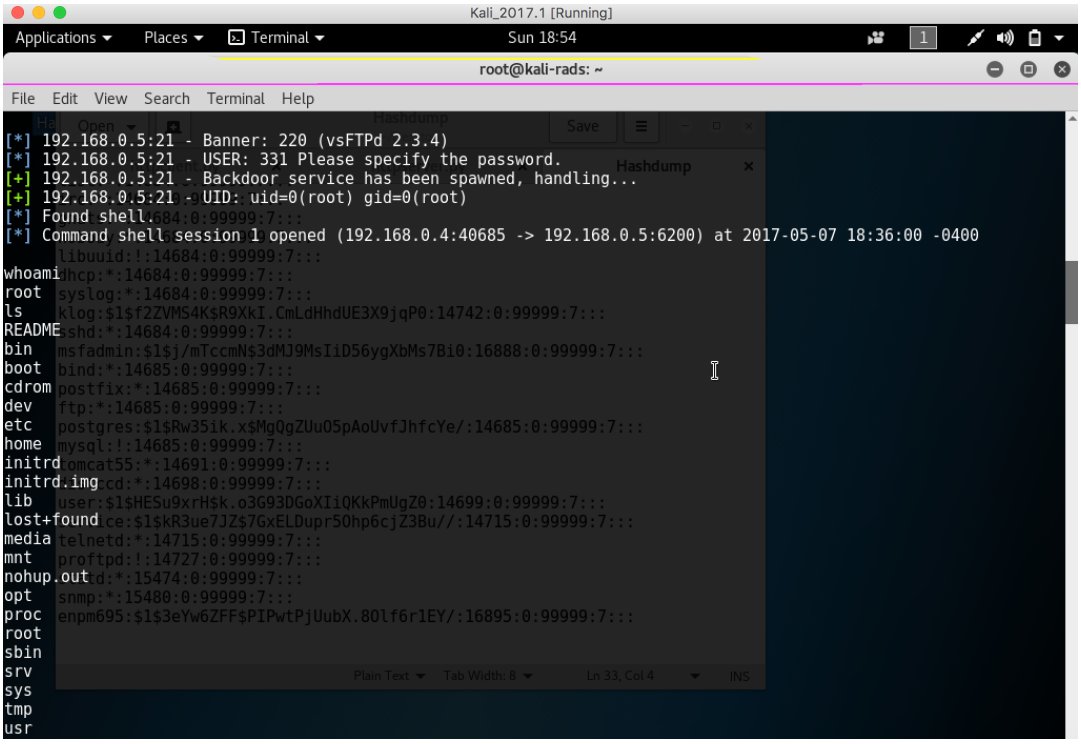
```
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some
 forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the
 site in a different fashion to the MIME type
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names
. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: in
dex.php
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and
 2.2.29 are also current.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /phpinfo.php?VARIABLE=<script>alert('Vulnerable')</script>: Output from the phpinfo() function was found.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via
 certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via
 certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via
 certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via
 certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected
 or limited to authorized hosts.
+ Server leaks inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mt
ime: Tue Dec  9 12:24:00 2008
+ OSVDB-3092: /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or
```

```
root@kali: ~
File  Edit  View  Search  Terminal  Help
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file n
ames. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were f
ound: index.php
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release)
 and 2.2.29 are also current.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /phpinfo.php?VARIABLE=<script>alert('Vulnerable')</script>: Output from the phpinfo() function was fou
nd.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information
 via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information
 via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information
 via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information
 via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be prote
cted or limited to authorized hosts.
+ Server leaks inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540
, mtime: Tue Dec  9 12:24:00 2008
+ OSVDB-3092: /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected
 or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This giv
es a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ /phpinfo.php?GLOBALS[test]=<script>alert(document.cookie);</script>: Output from the phpinfo() functio
n was found.
+ /phpinfo.php?cx[]=JwQtLoNMvOiNh0zXcJ6jtwHKxcdx54is9IH9kB8wJAM0tlfnxK6fSXtaDVGMC61CMPI9yOEJXt0dQUfbBxED
Te6ijxfBMrTFdQwdkod13b11pJcVE3hps4mdkngMRlzCBwHDDcXjH4UX61Gb5Jjuw8MOOjO6e0BD4HLqKJWbf4dsxNTco2VyTqTcmxL2
K4UDX2bNnZzuIikmniEG0ANvayvfB6O5FV4fPLbC7cr4vJPgKqx0mwAQt7ONGLZo3sWzkMt7W6dveoQq5TCy1DCRcYA6bHh5SE25YPFq
TcNktGqG6n6xRwELpFZQ4tzLlydyyXRwTzx1s8KI9rIQeCSwvIMk4VHNZ9cZG2L1KHOzIpDqLwWuwhLFmZUnVfWMaFlDOGVgaCuJJMEG
```

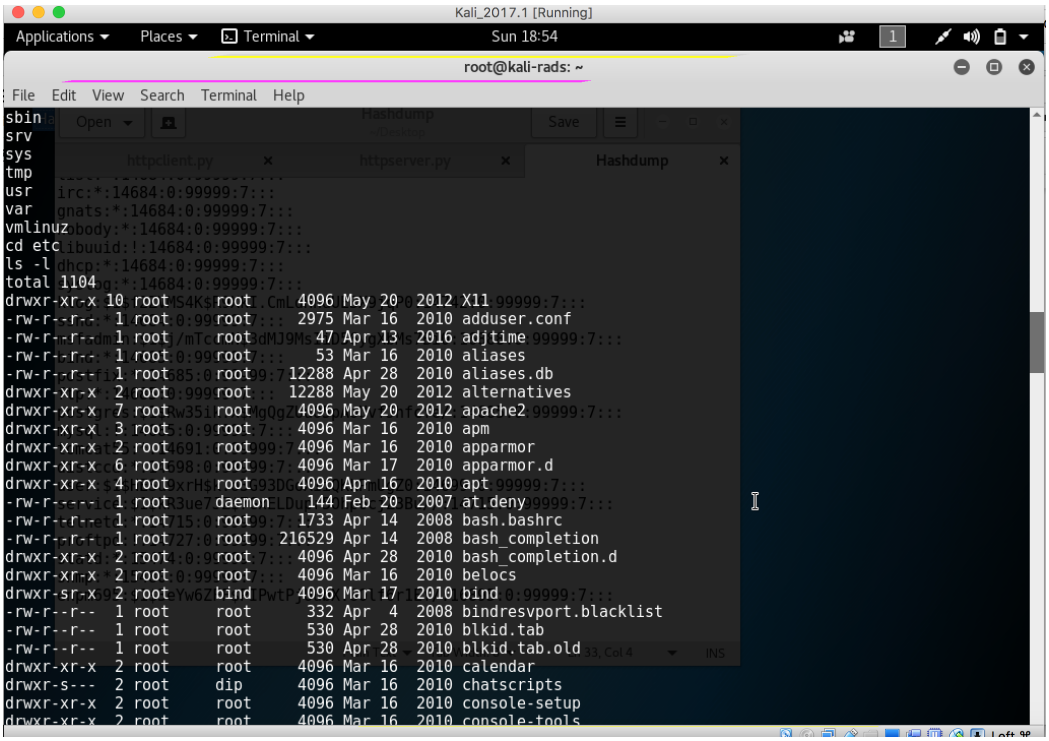

lDOGVgaCuJJMEGyw02lBoPL2wjhWtLazfYr3Dk8pWYrDy4czIyfuvkmuuIh5nID9EKNfWg9MR2SAsb3p0Dak1gRnedWuCARUacLtdmZ4yauJdIzSrYhSnUOw608ZqL8fdYbLPw
9oc5Nhxfo3nTvSeuBR4xiIOa8cRtoyyMS3opzvpOrzYvEQ1aWXUPWFsujkWVjSXR0iRlnY93to0R2xWFI9LWLYCkgJVIxbvPabjQJ9MlJXyBLBFQhbhauFyqig7yaWDWZQptwg
7h0KfHBYA22mSir4VDFEHTvYCpzRaf6v4fQ5HJjmOlA9KCQDE9h815zirtZ49izzJveBl9NywE0o40JBMsYKE1tSgp0ncQQsF1cFo8sOfDTRRLJPRsccTeAQ7IT72BQIY45Ejp
C1urluVyiFhlovTBBI3nvrmrqMvAM8pF6YbsQXDY1W6LJ1aEiRxdg3J0ySW5rWR2Boxz2WKmcPTyppYakcrkA95M60Zve4WYD7LJkEYdy6ZRzTQHF41KcJY4VwrCFsEmY25Uz8
JMi8RSzv290bHULXkgBHmiSLuVL6LfiRvjbeSQUAenRYMBw29arW2jtOOVd7gQkTssZI2MEf2WFYXGf4EF8Ye8ja3bILxxOBCo2vjmdcQAxrL5aodlBI76VhJjPbrnrSqV3hNF
Zsnf58IO70RqE7rHVUogxQd7sHJx5cAZ9oFxHsCaQoGIFDrcPZ1al59nBt6kRakucIHKaqnSsrATgH1l0KKwLQdQSzPKZIC0UOwvyi0zWP4OONEraqMtz448mIYxNp5BUUXPuW
d1Yll32FwuL7GGmmSEFuf80Q7kEJNOJM6UFRrvtVnJ2e466dtbBGZuTdxL1YHL6BvgJcJkVVEeU5grMAtqpUMkpmOGj3o0KQnhtviPW1iQD8Jy71GidUhHNAaxX0I1FPU8dISJ
30Vgemloq3oa39Ob1Le6vqUnIoBTFmCnf9ArBel2oi0Q5dX7ShHQmzMTIv8xzzeh75R2zMhccFBvtj0JUkrBRG4Eel88ntfHfBCjxrvKtlLkjIo9KWCOYdKhgbOTaztxdba41P
MVuVmell1lxqfauyrXGXqJDHw8YgTWRikmjgE7F7xZMwQOIlRPCuDqkWKzKCoIkx3Oeuclezxc1ZtUHzm4ApMkR33Lh7acQSbPYFhBkdEdbWOO3bmjuQPPrVa3HNEVCD74lQ39
SdAXFqGy4h8QnDq9rtREEdWqV8GsCu6oMG9y30DYF9f6erE59CwA5aJtiSx7kpQJRn2GJwV8ADsXimJYFIJpjH4Ggj28OamW74NWTCBsiplbwnsDV3eqHpkjdHeXbL1Bqaq2zO
VYfFaSDqKtTHNQD9NfzIpRl1wgu7aV97XuRDF489JGQzBg7XGfa2CPw7IAv5YLN9sOKt568oZyUOo4GiLo7HOAnRQ5Lhc4iYTKP7YjZltPClgoSHjLyt0706K48GRkWnTmvmkz
KmqdDK9H6IDp7V1bGUc3wrcWQJHSQ1xYwyXWOhW7105hhCcvUNgn3uMPbt6VRgEryZMt1mnlf4BQfRQMHT6kk0W3T8nlDZiwgCufw2piKSyoWIqq5Xkrs5JgoQDoFsb2XR1B9Z
znskp4quEfBs8l598dGzhw3U5WB0FGXsEjvusFltoE0xUJbB0WsLVtUto493cBGPnULxu1Wi9ZOOuVRk28CHa1wZg2WAHwD2s86Qxn3hKfgp7tZiy8Qg44Pfd6NYn0wFD7Gurb
GBk0ZyOvM5n8nqpfC3t5Yy5hzS3HgSCOtsEI8tCYbus1e8A25DzhZlWNFUGElYDUOWJ9fSdIi2xlj0sgZ5NAgyvINF4zmxSxUWFnuNnpsFJXQeA8h0ZjuE9EqF3qP1aUqQYFIT
KdzbXTbuEOo5Zs4fXie6RMmnp43aW4TcBCVXmNvY7fJcAD6CQrNleDMB51IUxXjjQEhpKy3o8J855krVbKCrRY4d0OH7t45YWpyIDDfbKLpJhWq4yY4fOYjoIdjUGFGge2AhFw
oB9sIqd8wreWNkoMD7faCLUbwXYTGrzN95umeebTN0DXPDP92VtTHnOTAyZgd0huCwvRBnDY9ut0kUAseGNTNEdaiFTbTVPVkFRVz2i2tTXg0RlGsWCnalQKgTUT7OjWEyipsU
7cDZGDa7EJ6rVNPL1tjXrj6FEvf6jLVfw88gxnnSjNzZCGOg6bkD1e2omLTEUnqKsLSNnMLljWKDiBBEO2AGaqTwbviEK40OLuuwD86VELLfrxPneWGCuBvILZnlhuPGiPHCBu
ibyppc78gNIu0GRc7y2iwDtyXZqMk2XNMvDpPGgD64vUQlFV91PUdQnKY8ezmhGWfBz5OXiBpdgB5bfrHf2VJLiwN36ZDqg7qCtLpkqblJhDqjpt179pdvTxkV4it8sV6d0o94
9mtnARd7ZaNThtKFPZgbuB6a5Z398ayYAEuDM2UDHXqUorQoiRQpsNZJEksTtpjS2LqeMsZ2PB4cnLYDa9IFB7OnL6aEtvCWbMij8zEZqMM94Xpj21DGEYzMv8hA9amLAfKHvl
R27Jid4kaKM5sleJaWFVMiAEjf1PDUxoaKxDXuMQnV4pNcCAaoInotVkgqRNzaVN03sUgvXzv5uL9eBk6MNXv2mf6Tk9W4SWBhkgEKCmc9wzyyjqPsvDjdr3RlXz7rMSulHMMv
wbSl18GwiJyrwe0aoLNxNWVMmIyE0y7oPB4QLxDAjFNn6AeAK6axPLpPOwFfav2btTpxrJ1Jy47Vy1AT2Doj7bUKDj0xALanHnfIBqQByoM1kPjiRhIwiMdvmzMg37fZemf58d
A8yC4JhP9rMReSC1kNYIGfU7l9IfkvLh4ZswCSv4f5QXsvqA89feRbafrVKnF8UtFqgHsehDkbOIjSbcDFMMudZDqT3R3mH4d1dQ0zCrY403BUWpbPVLSIV1xZuii2iJcIhTGc
2nU6ULTtoSgZp4hCF6656r5ys9IRonEtI8Jjl4U6cAl1Ri7avdMPOJIbe2KA3A6D2Frxu3mc3B2w3TVajo0guJkl2H8YGZzIvcoEkwcl5IB854PwiIbCqua1SXOGjzYQYOc91J
m5MCDE8wyn35P0dUcaOFXw3dCojJUQMz10p220fO5CFcvdbMIkLydgP7KKj2f3zG80Xlp7ILYQZ8Sk12p5jLPm0loUbc7NeRF5s7nMxnZJSZiA9CrP8zXaebsv3XZpEA5hVby4
hMBWP4J27wQnaRPV9Of0h3MhfGqWIGx6cHWijjapRw3SyR74kMnfIjD1n9HunuheK3lDIagF10Iwmw8C3JHEAzTFghZc0MEraUqTUEw0UuDAQ9dLsQ2dEYglJEDE5SrpHHTJk1
NnMlZjKQVoiMRaI182vBh5LdhEPKLzStftD7H33XzZe6s3x9UIfB2wKIkhW0h8aqBjl15jKxqtKioN7XHAe2nMx8JXhdkjG5V8g7xXZLchsjsocUWMMQdcgULmcczzckD5fDIu
ogmubAzlqQl6cEKhTWRXExVLdU668WEjHyGakKvGxSMdEKq06Tle8vWqDaryxIb9vewJ5AwTbZATovJenqW6IgpljOUdfKEvpqsATVBeTfJI0I2yXU32DnTT2orRSRG7eXDJIC
S3DykP0iijQ9ScQDzolMSf3BfCBCmDGxsxu0banWQ8Ix4gh5OmEjkIZYZ5Jpx4qzznWWAGCcZz0aAn2niJjW4cuTATqHkmLI<script>alert(foo)</script>: Output fr
om the phpinfo() function was found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-3092: /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to author
ized hosts.
+ 8347 requests: 0 error(s) and 28 item(s) reported on remote host
+ End Time: 2017-02-08 22:18:38 (GMT-5) (21 seconds)

+ 1 host(s) tested
root@kali:~#

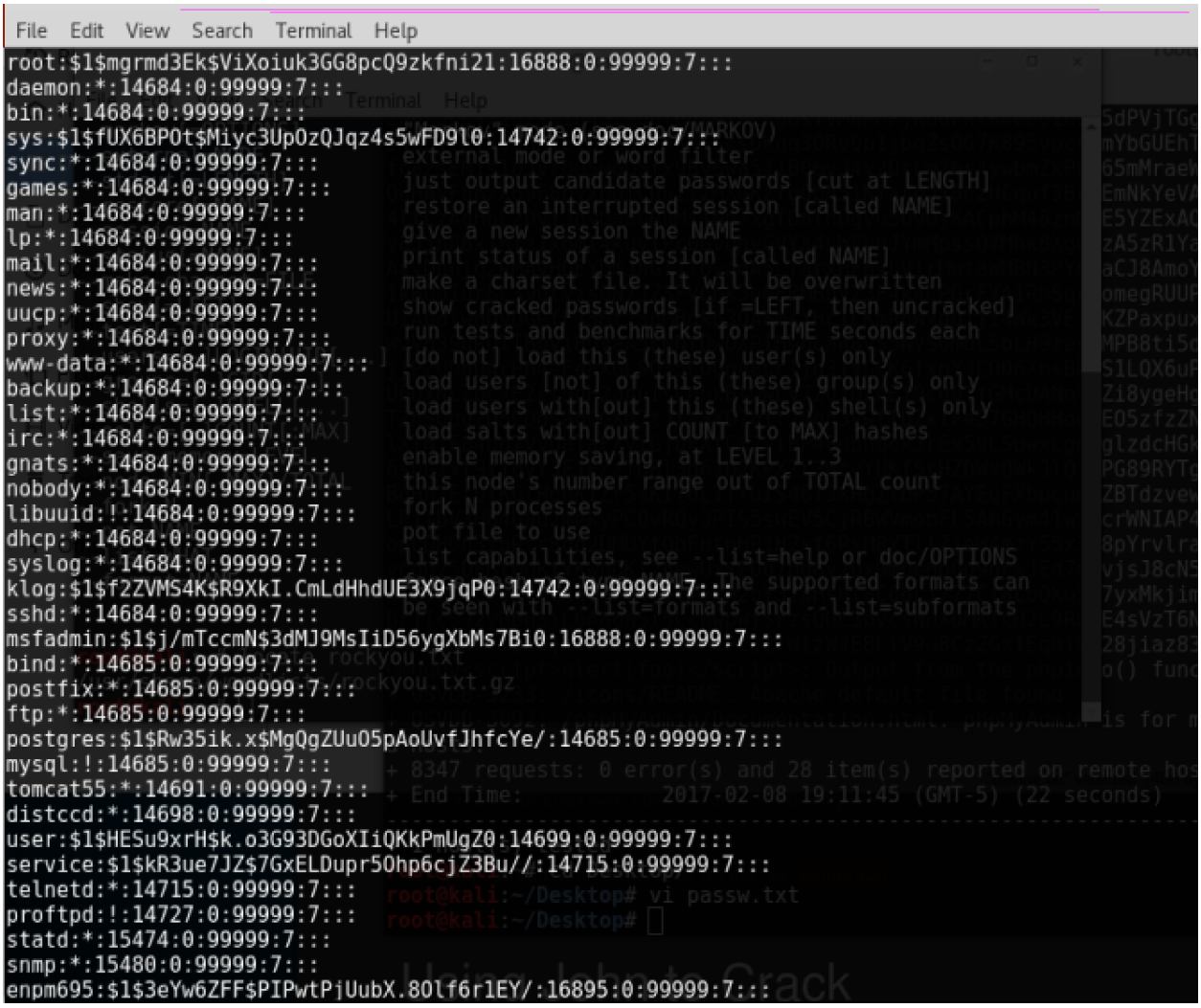
Task 4: Crack passwords (10 points) – crack 3 user account passwords will award students the full 10 points. Crack the root password will award an additional 10 points
In the target system, find the list of files using **ls**



Navigate to directory **etc** and check for **shadow** file to view password hashes.

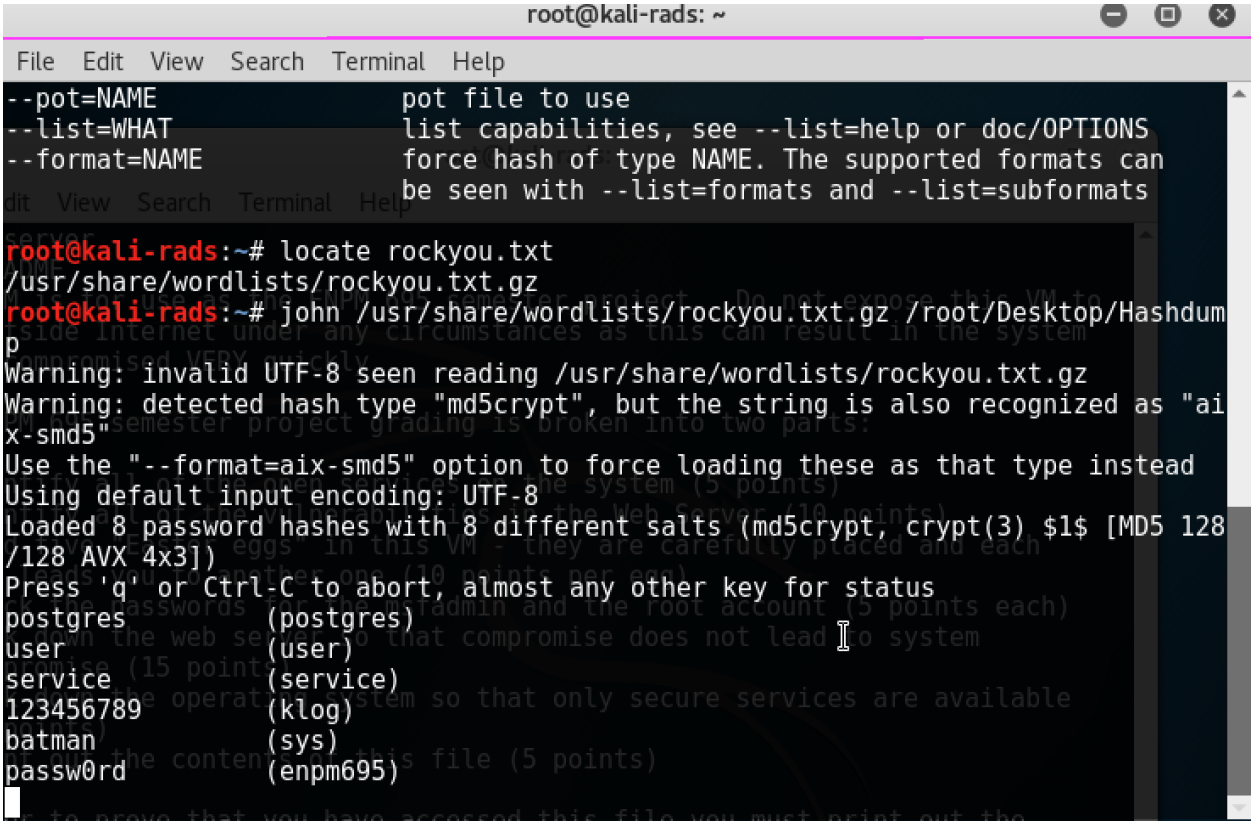


cat shadow to display its contents.



Save the contents in a file.

Goto Kali Application, and use John the Ripper to decrypt the file contents.

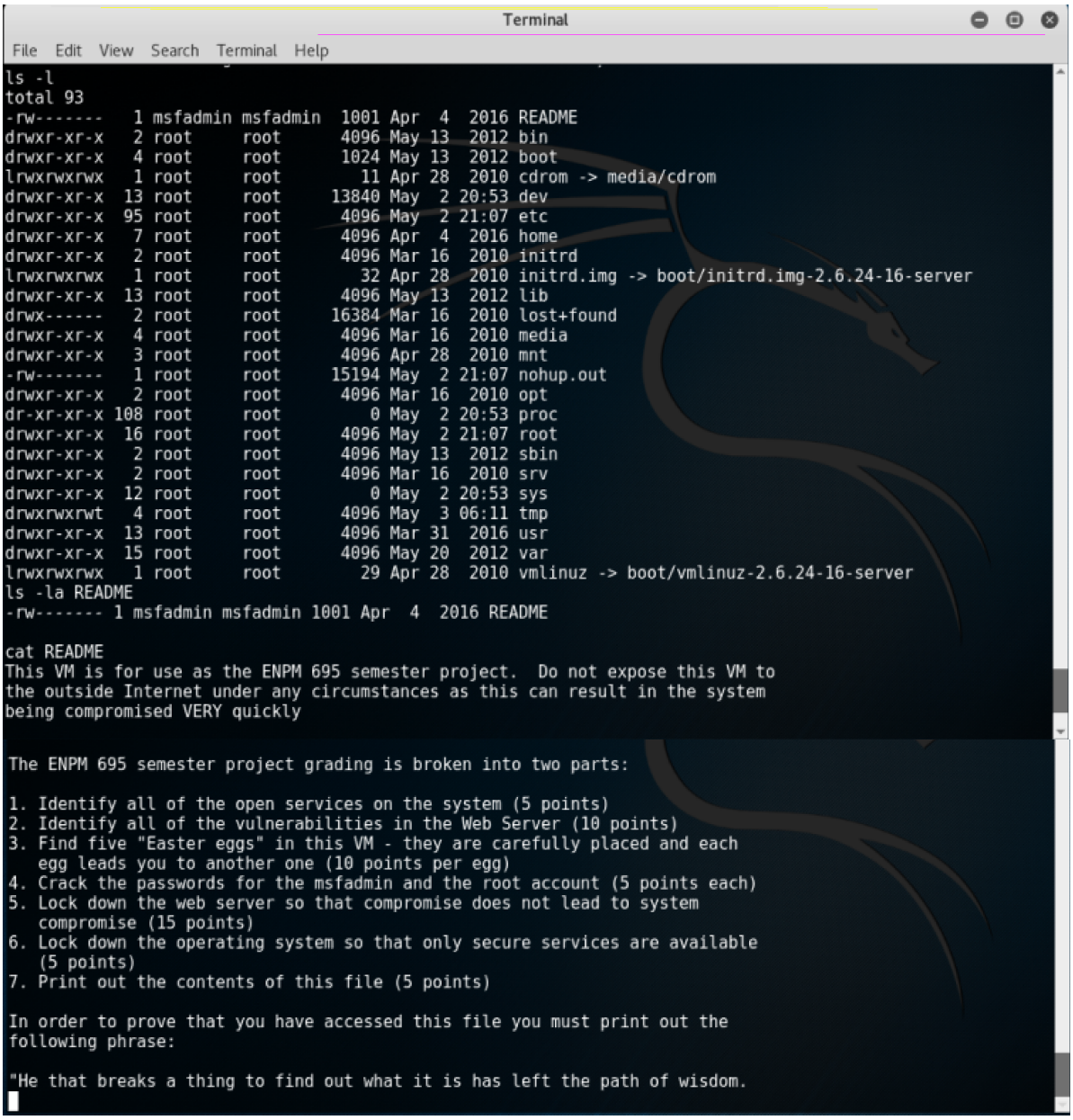


Task 5: Find out the information stored in a specific file (/README) within the system (10 points) – accessing the file and providing its contents (5 points). Identifying and explaining the protection around the file (5 points)

Within the target system, navigate to the README file as shown below.

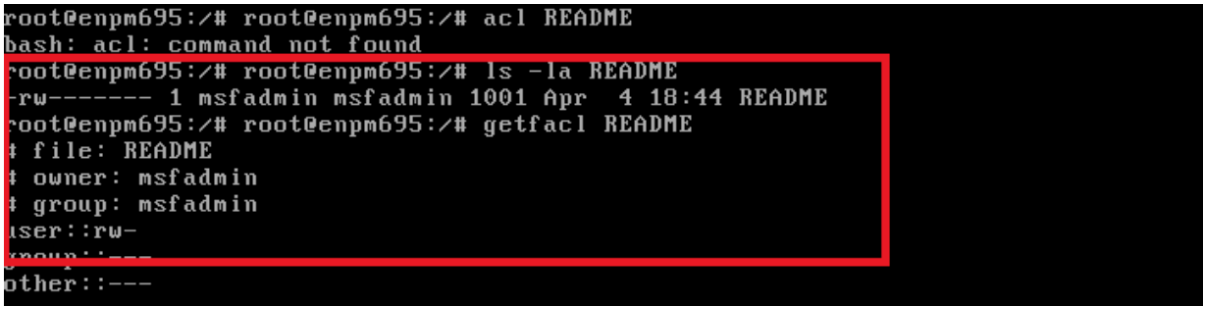
cd ../..
ls -l

cat README to display its contents.



The protection of the README file are as follows:

Only the owner of the file i.e msfadmin can read and write to the file.



Task 6: Define the Attack Surface of the system and web server (5 points each)

The System has following attack surfaces which an attacker can use to gain entry.

1. The Apache web server is running on port 80 which allows web traffic to flow unencrypted and attackers can easily intercept the information on the network via man-in-middle-attacks.
2. It may also invite number of attackers specially the ones trying to brute force user credentials by probing IPs on port 22 which is the default listening port for SSH. If port 22 would respond a brute force attack may appear. Despite its wide acceptance, there are still threats and occasionally software vulnerabilities associated with using SSH. For example, common libraries used by many implementations of SSH – like OpenSSL – may be reported. The mere fact that an SSH server is running and accessible from the Internet will invite attacks.
3. Open port 21 – FTP is vulnerable and allows remote attackers to cause a denial of service. It is also prone to attacks by Trojan horses/backdoors like Blade runner, Doly Trojan, Invisible FTP to name a few.
4. Open port 23 runs Telnet which used for remote maintenance of many networking communications devices including routers and switches, and often provides access to remote system with admin privileges. Given access to a server, or a network router of a corporate network or ISP, an attacker can perform a great deal of mischief. The level of access provided by telnet makes it a valuable commodity for individuals attempting to gain unauthorized access to systems or networks. This makes port 23 a very common target of attackers during network scans and reconnaissance attempts. Trojans that use port 23 include ADM worm, Fire Hacker, Telnet Pro to name a few.
5. A hacker can easily gain access to the system using easy passwords like ‘passw0rd’ within

few attempts.

6. Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.

The Web server has following attack surfaces which an attacker can use to gain entry.

1. The web server has web pages that provide access to directories without user authentication. If there is any sensitive information stored in the directories it can be misused by the attacker to gain access to the system.
2. Attacker can create a backdoor and get access to the system, thus it is vulnerable to backdoor attacks.
3. A webserver page provides information regarding the name/version of the webserver and the port on which it is running. This information can be used by an attacker to find the loopholes and hack the system based on the vulnerabilities the server/version and the open port has.
4. Apache server in general is a large surface for attack via the vulnerabilities it has like click-jacking attacks, CSRF attacks, XSS attacks, stack based overflow attacks, DOS attacks, input validation attacks, SQL injection attacks, impersonation attacks, URL interpretation attacks to name a few.
5. The web server is vulnerable to click-jacking attacks due to absence of X-frame options header and this can be used by an attacker to trick the user and gain access.
6. It is also vulnerable to XSS attacks due to absence of XSS protection header, thus providing another entry point to the system and allowing attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
7. The server leaks inodes via Etags with header found with file /phpMyAdmin/ChangeLog.
8. Also, an attacker can access default accounts in applications on the server (example: DVWA), unused pages, unpatched flaws, unprotected files and directories, etc. to gain unauthorized access to or knowledge of the system.

Task Group 2: Improve the security of the system

The second task group is to focus the student on how to improve the security of the system. This includes reducing the attack surface by removing services and locking down applications running on the system. The following tasks must be completed to get full credit for this portion of the project:

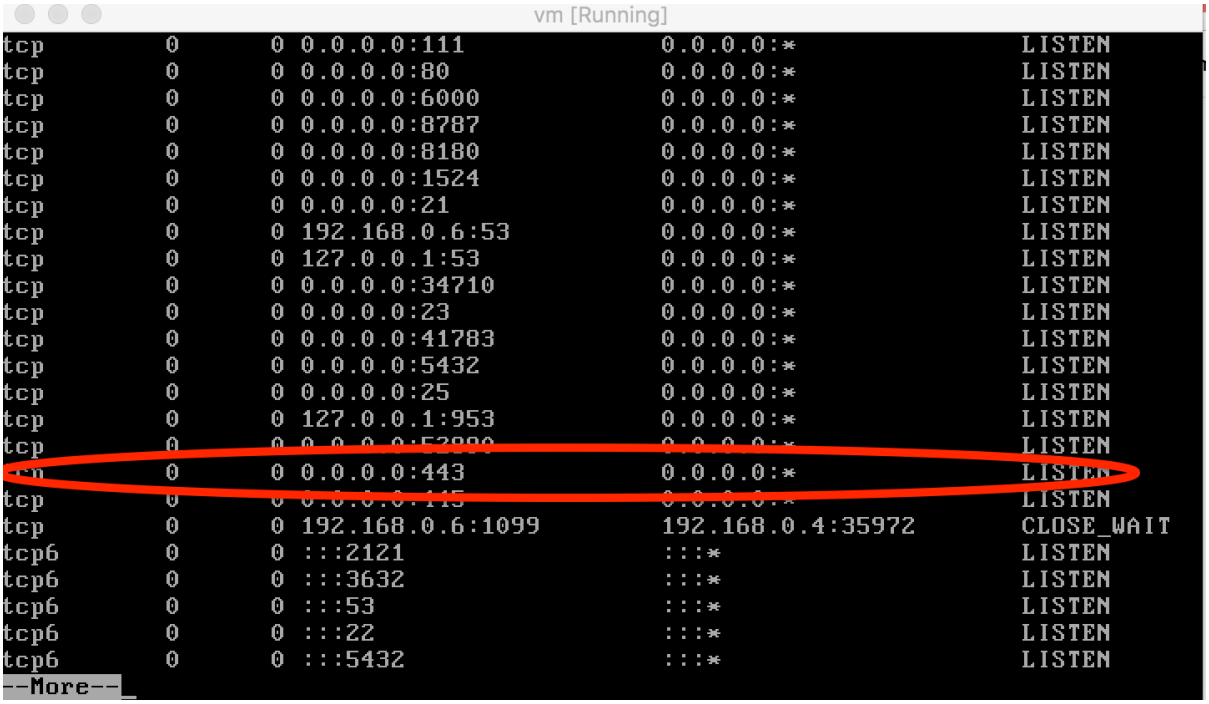
Task 1: Lock down the web server (20 points) – point breakdown is as follows:

a. Setting up HTTPS (10 points)

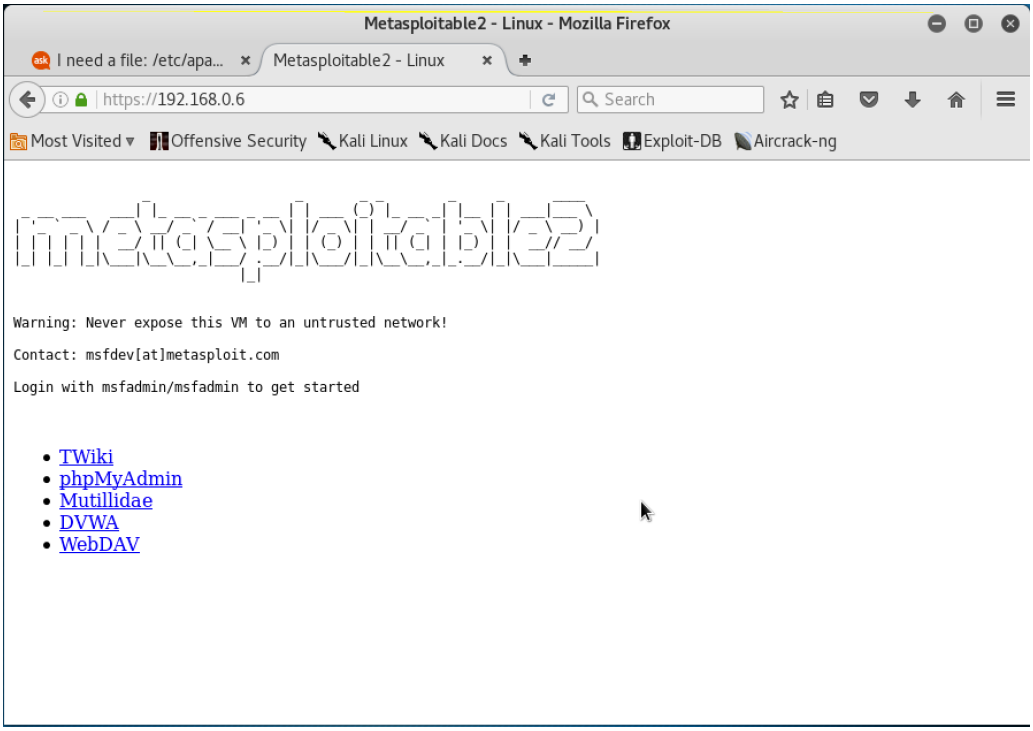
Steps used to set up https on the linux server are as shown in https://samsclass.info/120/proj/p17-https.html

(not able to add additional screenshots as the target system displays only one page)

The below screenshot shows 443 is running.

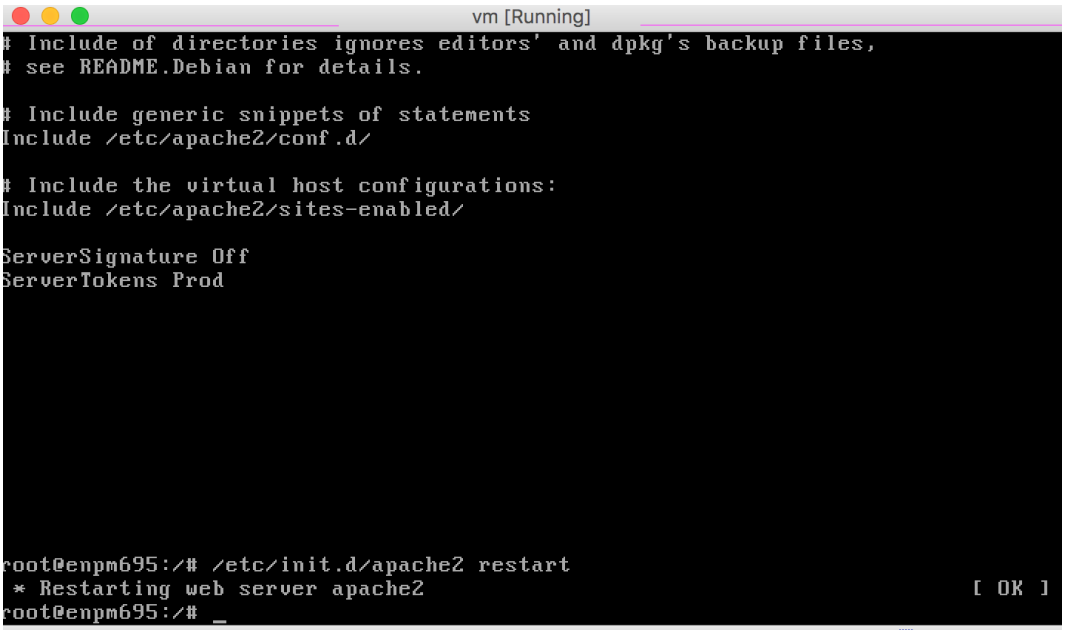


On restarting the apache server, we can see that the https has been configured on the server as shown below.

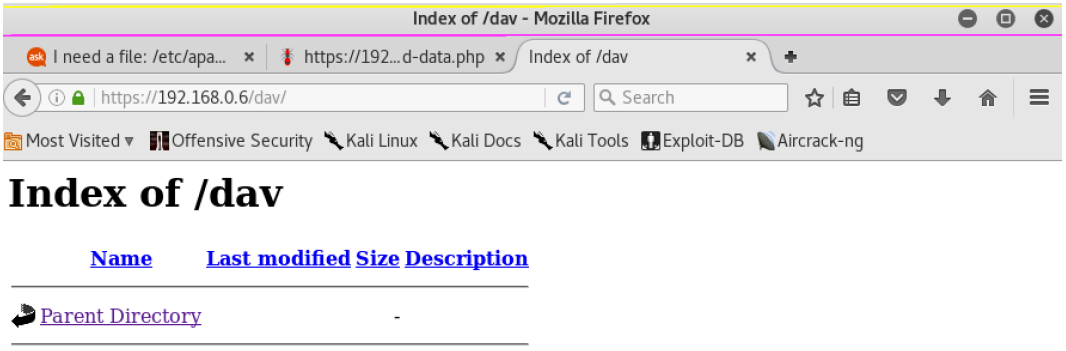


b. Eliminating identifying information that the web server gives out (5 points)

The /dav page on the webserver displays information like server name/version/OS details which can be used by an attacker to exploit the vulnerabilities based on these details.
To eliminate this information,we need to navigate to /etc/httpd/conf/httpd.conf and add below contents to end of the file as show in the screenshot.
ServerSignature Off
ServerTokens Prod



The site now no longer displays the identifying information.



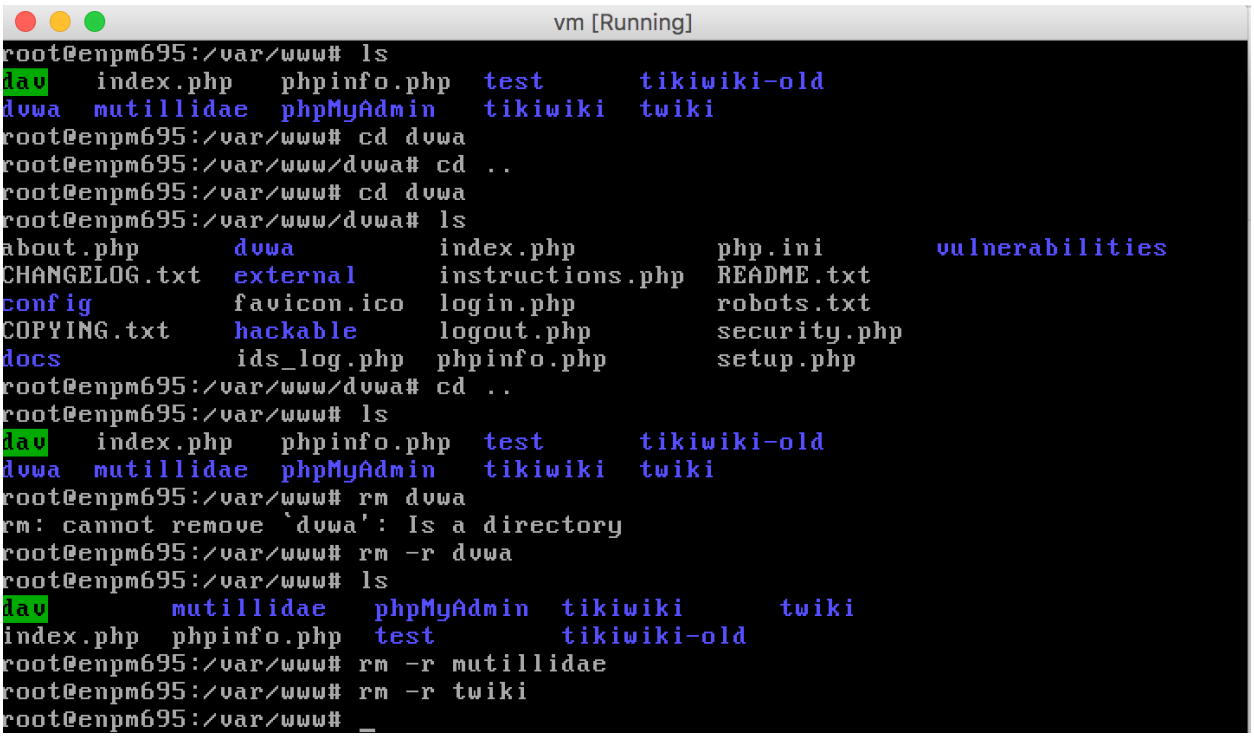
Reference: http://ask.xmodulo.com/turn-off-server-signature-apache-web-server.html

c. Eliminate vulnerable applications (5 points)

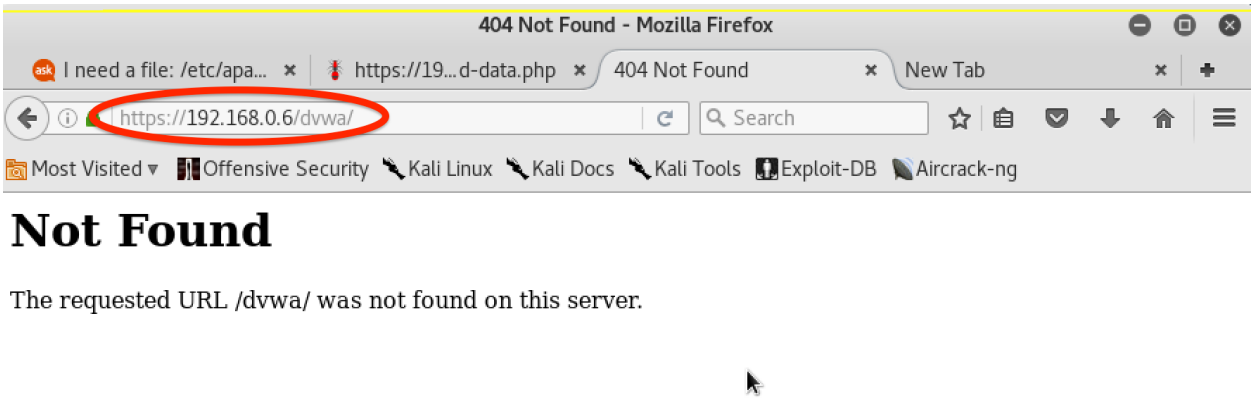
The webserver has below vulnerable applications hosted on it.

1. Twiki
2. DVWA
3. Mutillidae

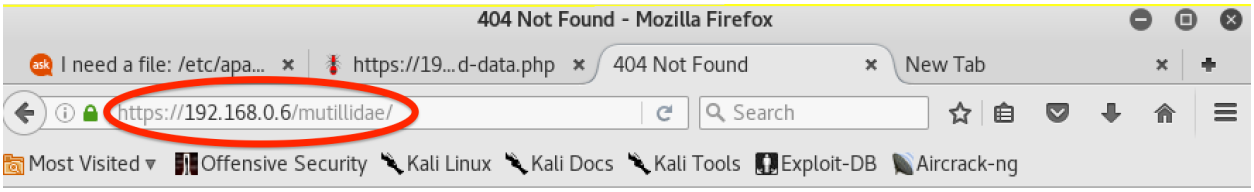
These applications have been removed using command **rm –r directoryname**



The application DVWA is no longer available.



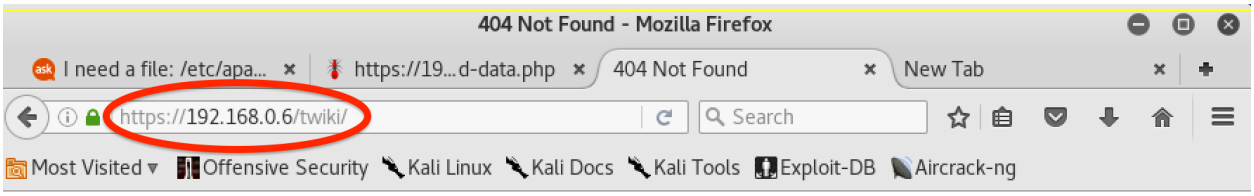
The application Mutillidae is no longer available.



Not Found

The requested URL /mutillidae/ was not found on this server.

The application Twiki is no longer available.



Not Found

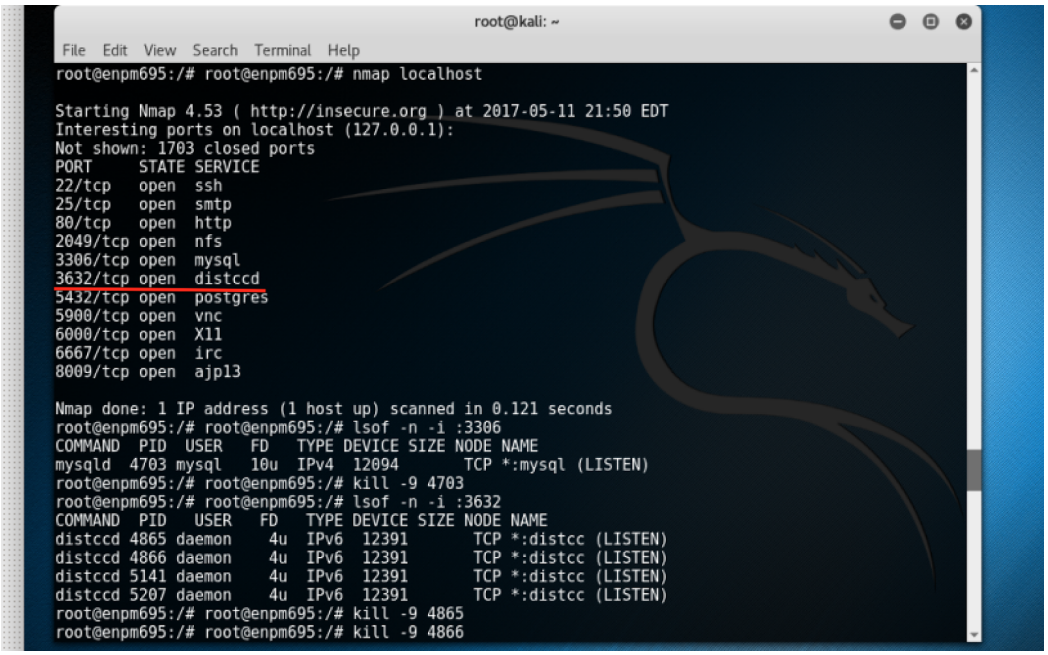
The requested URL /twiki/ was not found on this server.

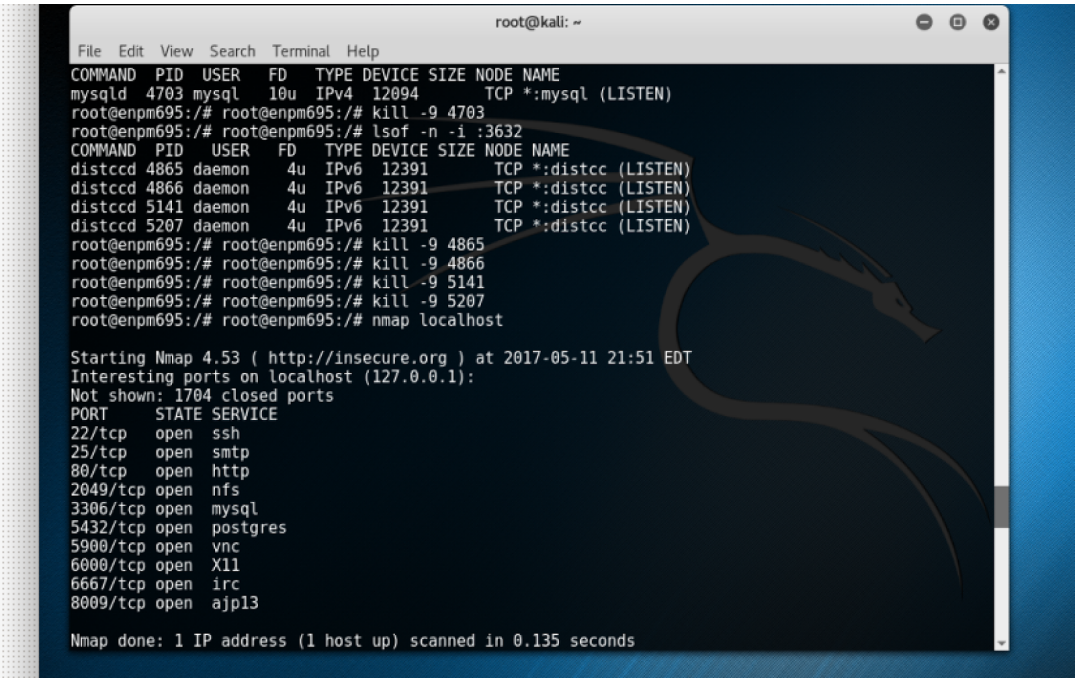
Task 2: Lock down the operating system so that only the following services are readily available:
Secure Shell
HTTP/HTTPS
Mail
Note: You may not use the iptables firewall to do this – you must do this by stopping services from running (5 points each - 15 points) and only running services that are secure

This task can be achieved by killing the services which are running on the open ports.

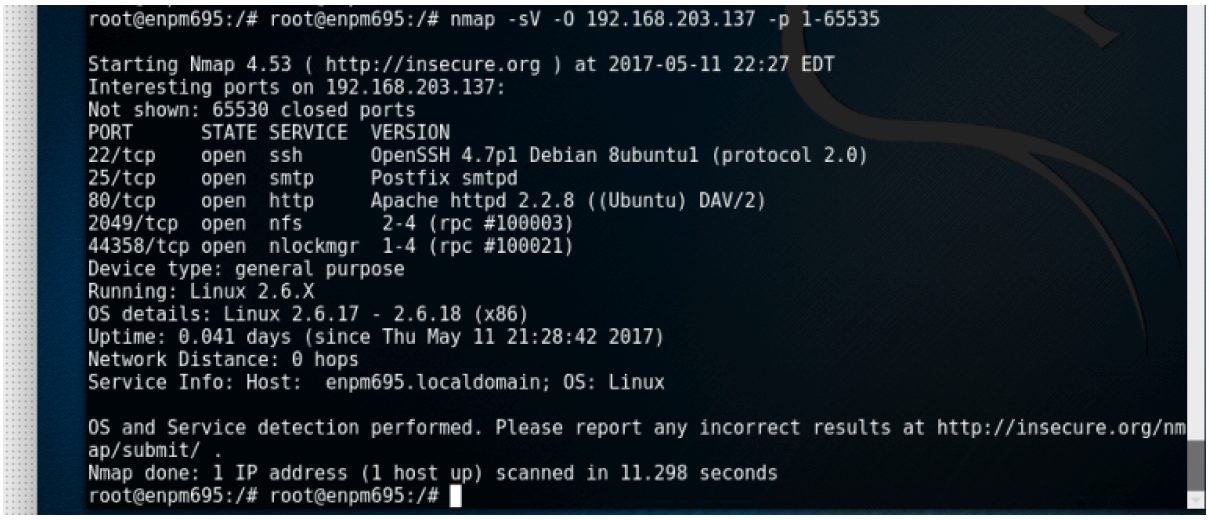
The commands used lsof –n –I : <port number> , gives PID of the service running on the specific port mentioned.

Kill -9 <PID> this command is used to kill the PID.

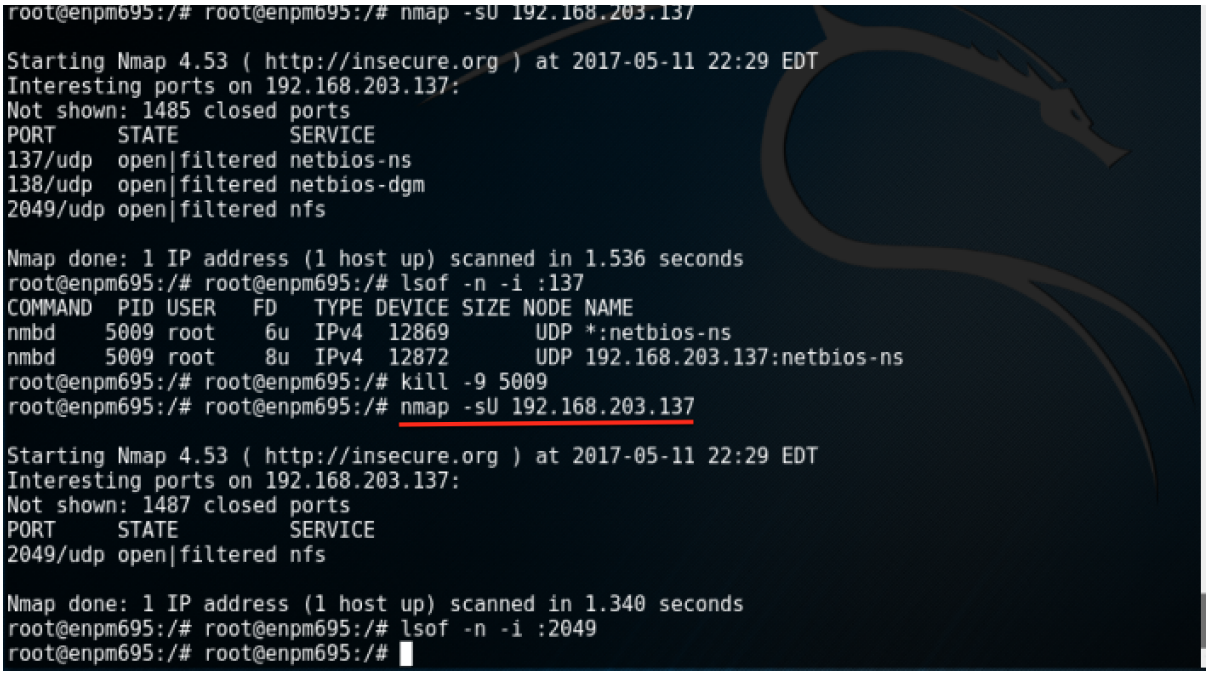




nmap shows the result of open tcp ports.



nmap shows the result of open udp ports.



Task 3: Identify the operating system (5 points)
We are able find the operating system running using the command **“uname –a”**

Task 4: Define the attack surface of the hardened system and web server application (10 points)

The hardened system has all services disabled except the below which have some vulnerabilities that can be used as an attack surface for entry point into the system.

Secure Shell
HTTP/HTTPS
Mail

Secure shell is vulnerable to brute force attacks specially by probing IPs on port 22 which is the default listening port for SSH.

SMTP service may be vulnerable to an open mail relay which is an SMTP server configured in such a way that it allows anyone on the Internet to send e-mail through it, not just mail destined to or originating from known users. By processing mail that is neither for nor from a local user, an open relay makes it possible for an unscrupulous sender to route large volumes of spam. In effect, the owner of the server -- who is typically unaware of the problem -- donates network and computer resources to the sender's purpose.

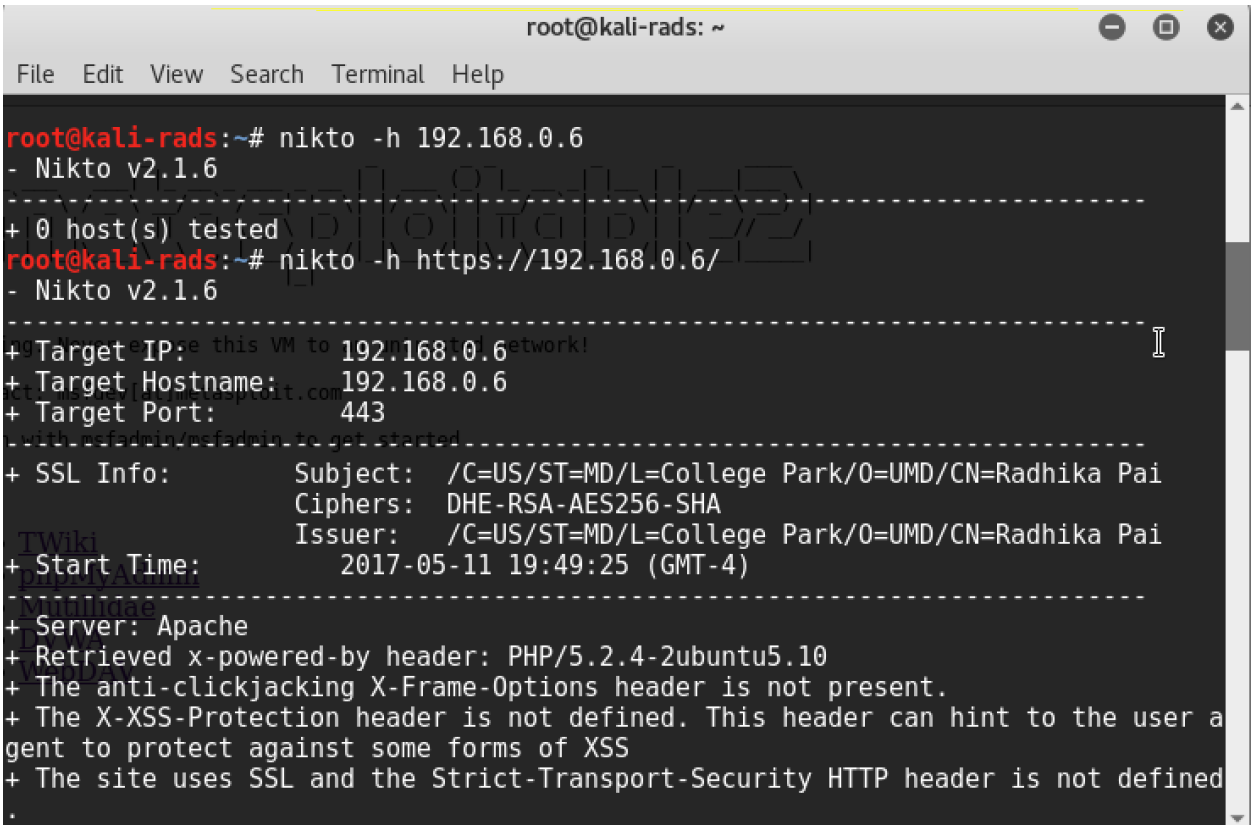
Apache web server is running on TCP ports 80 and 443 which are very popular and can be discovered by attackers easily. When port 80 is open, web traffic flows unencrypted and attackers can easily intercept the information on the network via man-in-middle-attacks. Thus it is preferred to use port 443 because the traffic is encrypted. Also, opening them from a restricted IP reduces the attack surface. However, open SSL port 443 is vulnerable to heartbleed attacks which allows an attacker to capture passwords and other confidential information via the SSL port 443. Also, a hacker can push bots or take control of the server and in turn reach the internal network. Once they have control of the web server they might be able to leverage that access to gain control of other machines. It very much depends on which other internal servers that machine can communicate with.

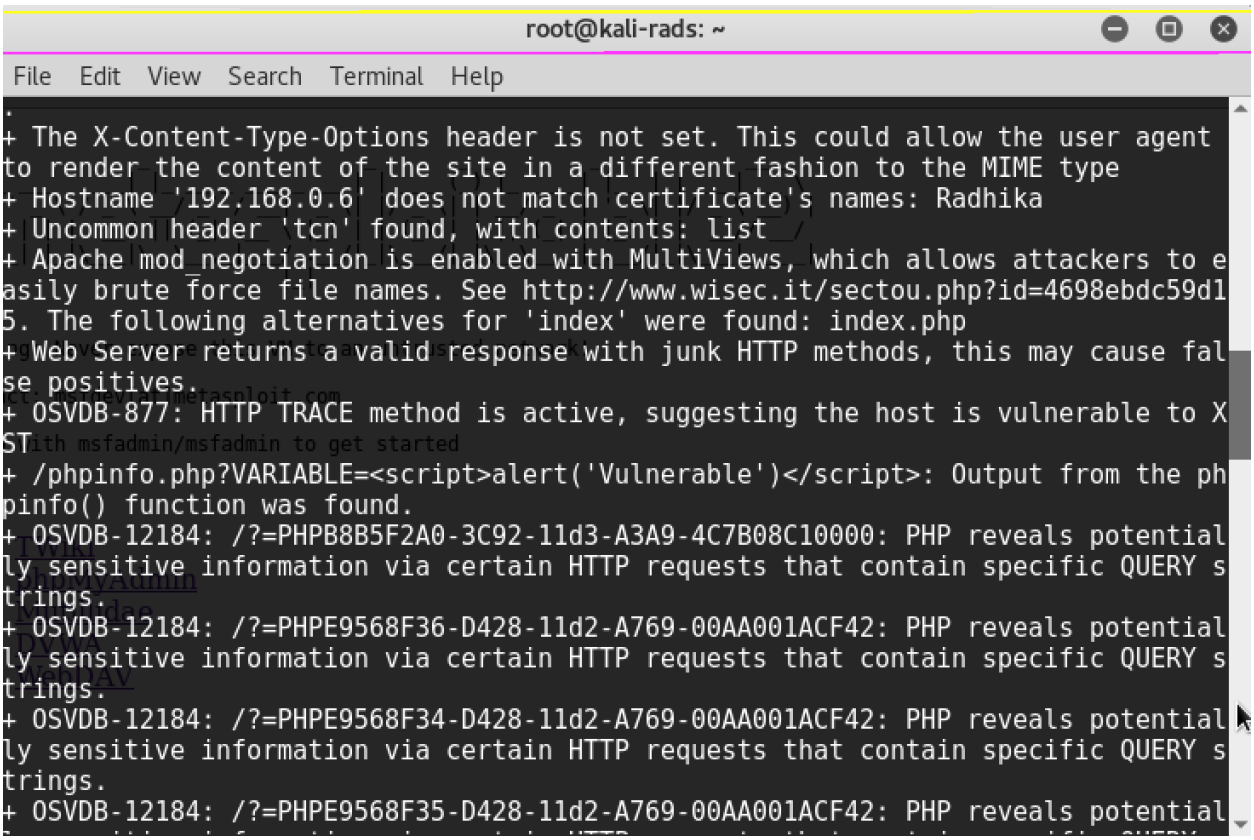
Although the website is hardened by hosting on https and removing the vulnerable applications, a scan using nikto has discovered some vulnerabilities which can be used as attack surfaces by the attackers:

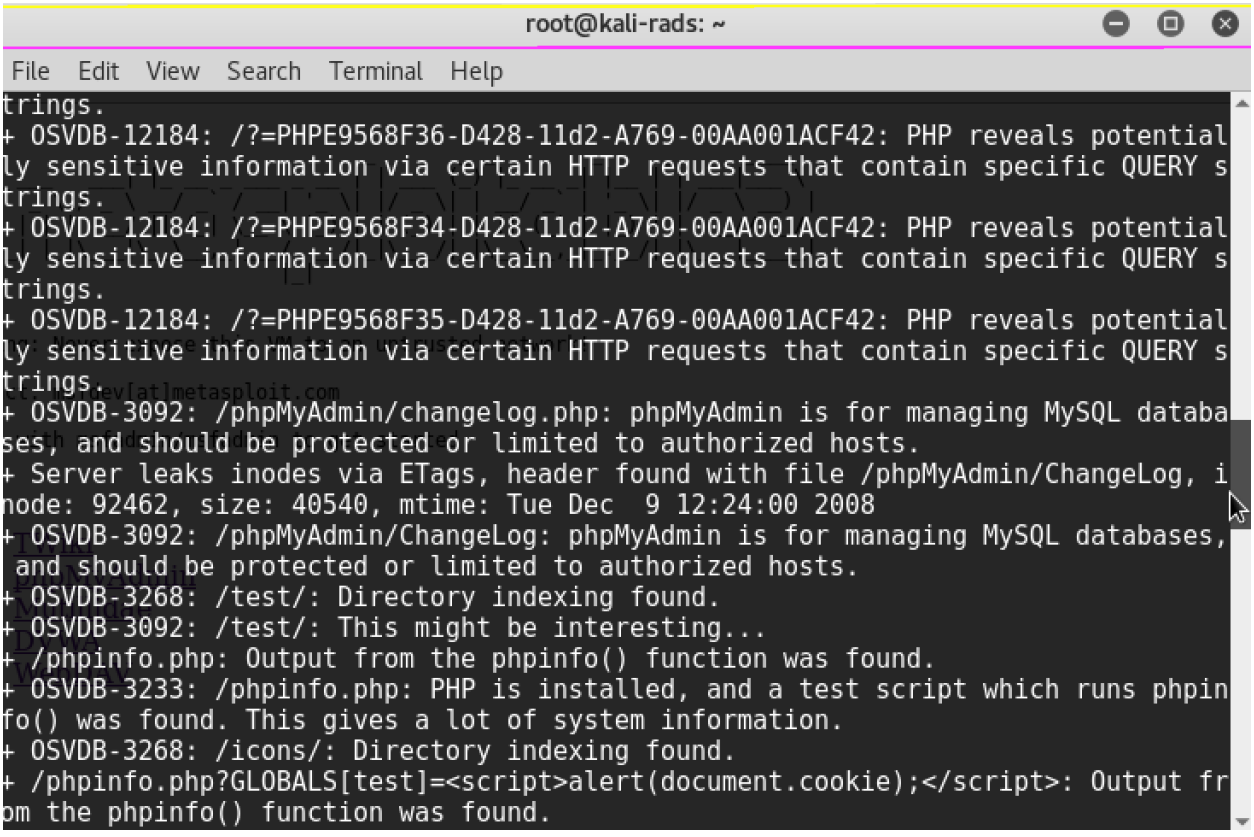
1. Web server is vulnerable to click-jacking attacks due to absence of X-frame options

header.

2. It is also vulnerable to XSS attacks due to absence of XSS protection header.
3. Moreover, Apache/2.2.8 being used appears to be outdated as the current versions in use are atleast Apache/2.4.12 which is comparatively less vulnerable to attacks.
4. The server leaks inodes via Etags with header found with file /phpMyAdmin/ChangeLog.
5. Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.




root@kali-rads: ~
File Edit View Search Terminal Help
.
+ 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
+ Hostname '192.168.0.6' does not match certificate's names: Radhika
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to e
asily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d1
5. The following alternatives for 'index' were found: index.php
+ Web Server returns a valid response with junk HTTP methods, this may cause fal
se positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to X
ST
+ /phpinfo.php?VARIABLE=<script>alert('Vulnerable')</script>: Output from the ph
pinfo() function was found.
+ OSVDB-12184: /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potential


root@kali-rads: ~
File Edit View Search Terminal Help
trings.
+ OSVDB-12184: /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-12184: /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potential
ly sensitive information via certain HTTP requests that contain specific QUERY s
trings.
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databa
ses, and should be protected or limited to authorized hosts.
+ Server leaks inodes via ETags, header found with file /phpMyAdmin/ChangeLog, i
node: 92462, size: 40540, mtime: Tue Dec 9 12:24:00 2008
+ OSVDB-3092: /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases,
 and should be protected or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpin
fo() was found. This gives a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ /phpinfo.php?GLOBALS[test]=<script>alert(document.cookie);</script>: Output fr
om the phpinfo() function was found.

Tools used:

Nmap
Metasploit
Nikto
John the ripper

References:

1. http://www.codechewing.com/library/disable-remove-website-apache/
2. https://corenumb.wordpress.com/2013/03/04/metasploitable-2-ftp-exploitation-vsftpd-backdoor-session-1/
3. https://www.acunetix.com/vulnerabilities/web/smtp-open-mail-relay
4. https://www.owasp.org/index.php/Top_10_2013-Top_10
5. https://samsclass.info/120/proj/p17-https.html
6. http://ask.xmodulo.com/turn-off-server-signature-apache-web-server.html