Executive Summary

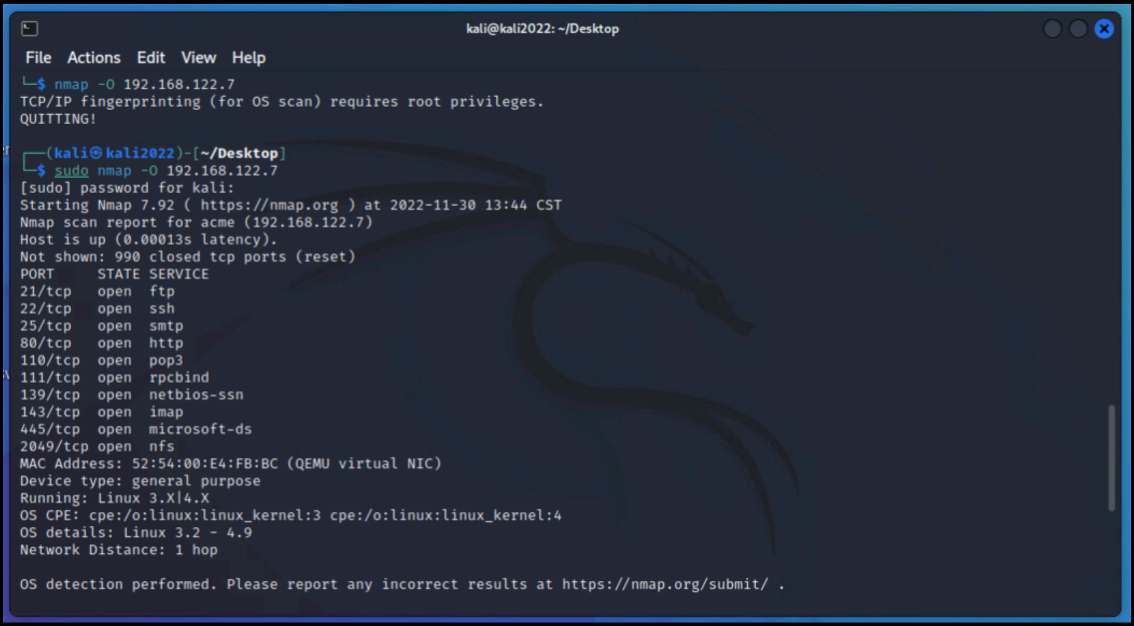
This summary reports on the findings that came from a pentest conducted on the ACME box located within the Netlabs Virtual Machine. Testing on the ACME machine began on 11/28/2022 and was terminated on 12/5/2022. Tests were conducted against the security controls that are currently in place on the machine. Methods used for testing follow the standard testing protocol for a black box machine-no information was provided to the tester. Testing was done in a safe manner (in an offline environment) that did not put any employees or data at risk.

The findings from the report place the system in high severity of risk. Exploitations on the system were found that could cause elevated privileges, loss of data, or system being halted. A main issue found was access to the database or framework which houses the information and structure of data on the system. Login info, credentials, and emails sent to the system were uncovered with basic reconnaissance techniques. Access into critical servers was also another outcome of the penetration test. The implications of these found vulnerabilities is that attackers will be able to infiltrate the system and elevate their access to obtain access to unauthorized data with the ability to modify it. All tests were done within the scope of the Netlabs Ubuntu Environment. Immediate action is recommended for the remediation and repair of access controls in the environment. Suggestions and best practices to follow for securing the machine will be discussed at the end of this report. Below, the findings will be explained in greater detail.

Initial Recon Findings and Intel Gathering

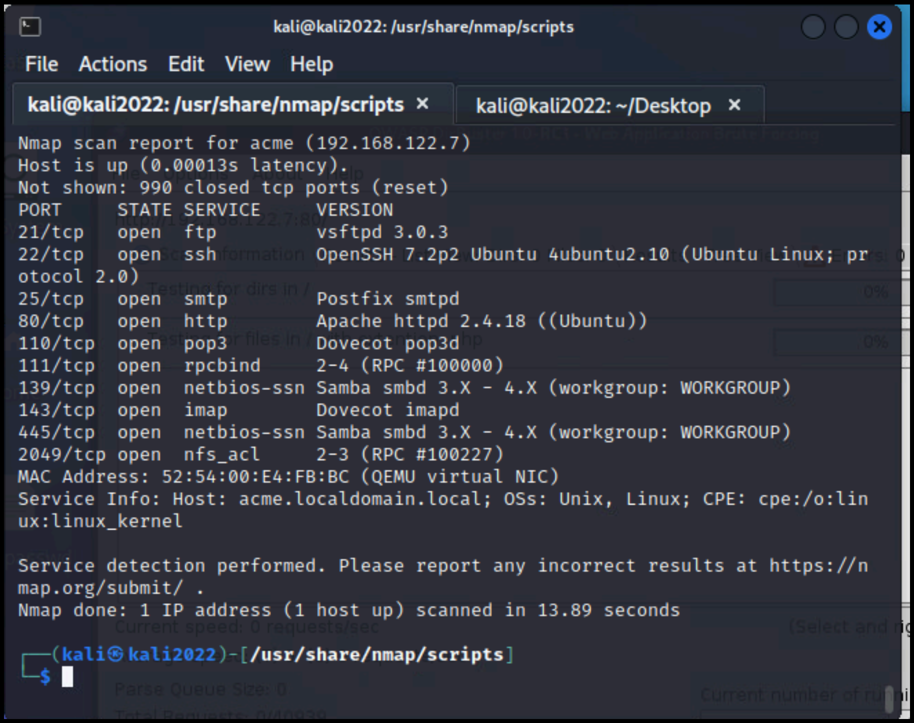
Using nmap to find open ports, versions, os versions, and script data.

The first step in this pentest was gathering data on what ports and services are open and vulnerable to get a better understanding of steps and strategies for the rest of the test. *Nmap -O 192.168.122.7* was used to find the open ports of the acme box. The findings are shown below.



```
└─$ nmap -O 192.168.122.7
TCP/IP fingerprinting (for OS scan) requires root privileges.
QUITTING!

┌──(kali㉿kali2022)-[~/Desktop]
└─$ sudo nmap -O 192.168.122.7
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-30 13:44 CST
Nmap scan report for acme (192.168.122.7)
Host is up (0.00013s latency).
Not shown: 990 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
110/tcp  open  pop3
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
143/tcp  open  imap
445/tcp  open  microsoft-ds
2049/tcp open  nfs
MAC Address: 52:54:00:E4:FB:BC (QEMU virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

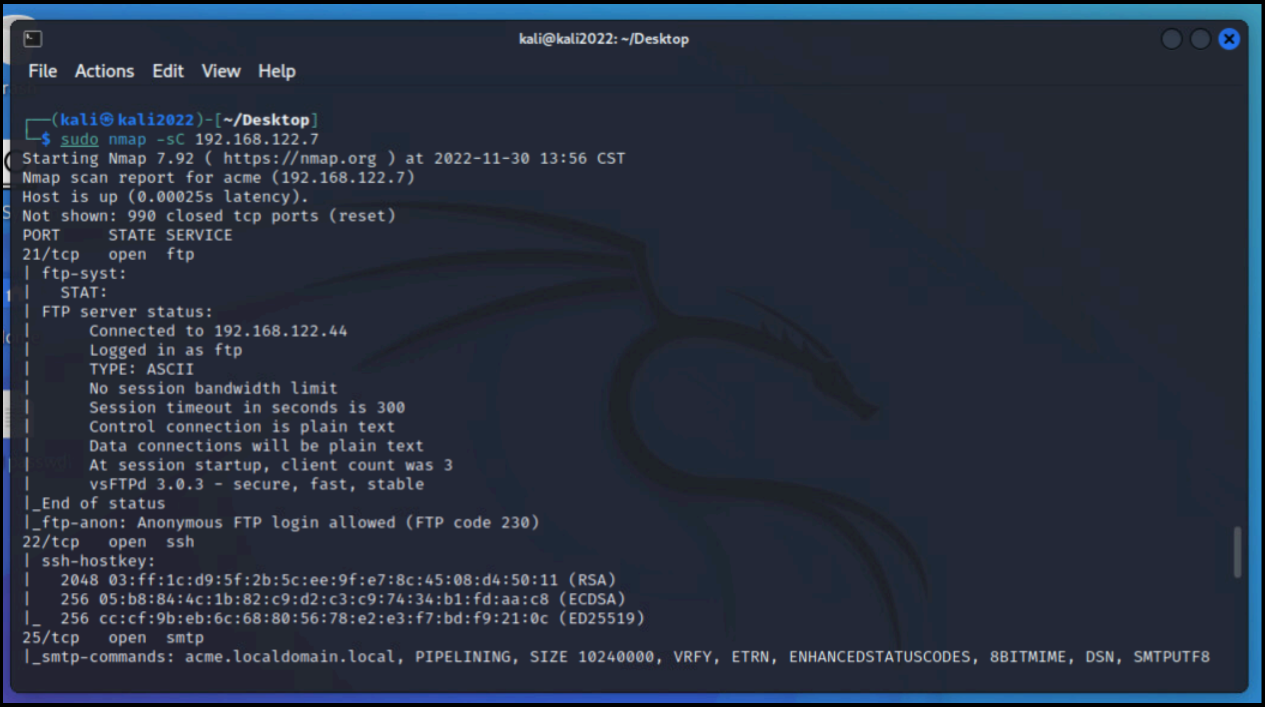
Next, command *nmap -sV 192.168.122.7* was used to get a deeper look into what versions the servers are running on the open ports from above. This data is a crucial step in setting up what ports will be tested for vulnerabilities in the coming tests. The finding of this command is listed below.


kali@kali2022: /usr/share/nmap/scripts
File Actions Edit View Help
kali@kali2022: /usr/share/nmap/scripts ×
kali@kali2022: ~/Desktop ×

```
Nmap scan report for acme (192.168.122.7)
Host is up (0.00013s latency).
Not shown: 990 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 3.0.3
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; pr
otocol 2.0)
25/tcp   open  smtp        Postfix smtpd
80/tcp   open  http        Apache httpd 2.4.18 ((Ubuntu))
110/tcp  open  pop3        Dovecot pop3d
111/tcp  open  rpcbind     2-4 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp  open  imap        Dovecot imapd
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
2049/tcp open  nfs_acl     2-3 (RPC #100227)
MAC Address: 52:54:00:E4:FB:BC (QEMU virtual NIC)
Service Info: Host: acme.localdomain.local; OSs: Unix, Linux; CPE: cpe:/o:lin
ux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.89 seconds

┌──(kali㉿kali2022)-[/usr/share/nmap/scripts]
└─$ █
```

Valuable information that can be taken away from the data above the versions of each of those services as well as the OS type: Linux.

Nmap -sC 192.168.122.7 was used to run the basic set of scripts and find anything important that stands out. Anonymous login is allowed which is a little flag and allows remote users to gain info on the system. The ssh hostkeys are also found but are not yet useful for anything yet. The results of the scan are below.



```
┌──(kali㉿kali2022)-[~/Desktop]
└─$ sudo nmap -sC 192.168.122.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-30 13:56 CST
Nmap scan report for acme (192.168.122.7)
Host is up (0.00025s latency).
Not shown: 990 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.122.44
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp   open  ssh
| ssh-hostkey:
|   2048 03:ff:1c:d9:5f:2b:5c:ee:9f:e7:8c:45:08:d4:50:11 (RSA)
|   256 05:b8:84:4c:1b:82:c9:d2:c3:c9:74:34:b1:fd:aa:c8 (ECDSA)
|_  256 cc:cf:9b:eb:6c:68:80:56:78:e2:e3:f7:bd:f9:21:0c (ED25519)
25/tcp   open  smtp
|_smtp-commands: acme.localdomain.local, PIPELINING, SIZE 10240000, VRFY, ETRN, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8
```

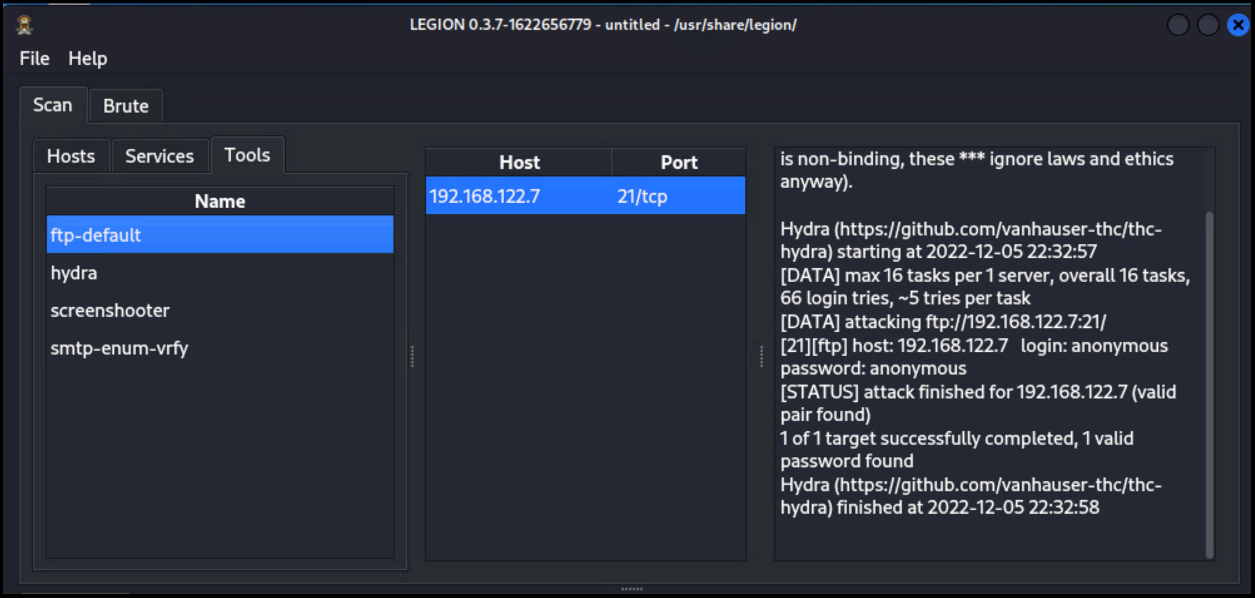
After gathering that information, I decided I would begin with the ftp port and work my way down till I could get everything from the other valuable ports: ssh, http, netbios, smb, and nfs.

Testing Outcomes

(Highest risk)
Access into ftp server method 2: Using legion tool

Access into the fpt server was found through running a hardmode legion scan on the IP 192.168.122.7 (ACME BOX)

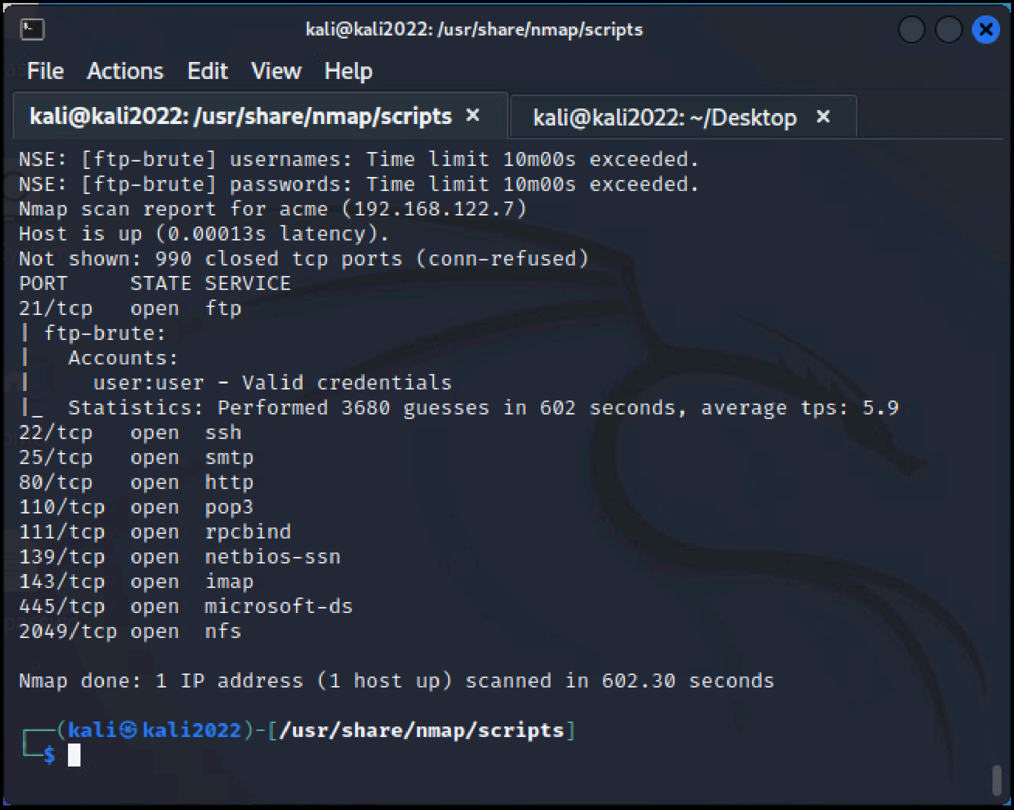
Legion was able to obtain the ftp username: anonymous and password: anonymous from the integrated password brute force hydra.



```
is non-binding, these *** ignore laws and ethics
anyway).

Hydra (https://github.com/vanhauser-thc/thc-
hydra) starting at 2022-12-05 22:32:57
[DATA] max 16 tasks per 1 server, overall 16 tasks,
66 login tries, ~5 tries per task
[DATA] attacking ftp://192.168.122.7:21/
[21][ftp] host: 192.168.122.7   login: anonymous
password: anonymous
[STATUS] attack finished for 192.168.122.7 (valid
pair found)
1 of 1 target successfully completed, 1 valid
password found
Hydra (https://github.com/vanhauser-thc/thc-
hydra) finished at 2022-12-05 22:32:58
```

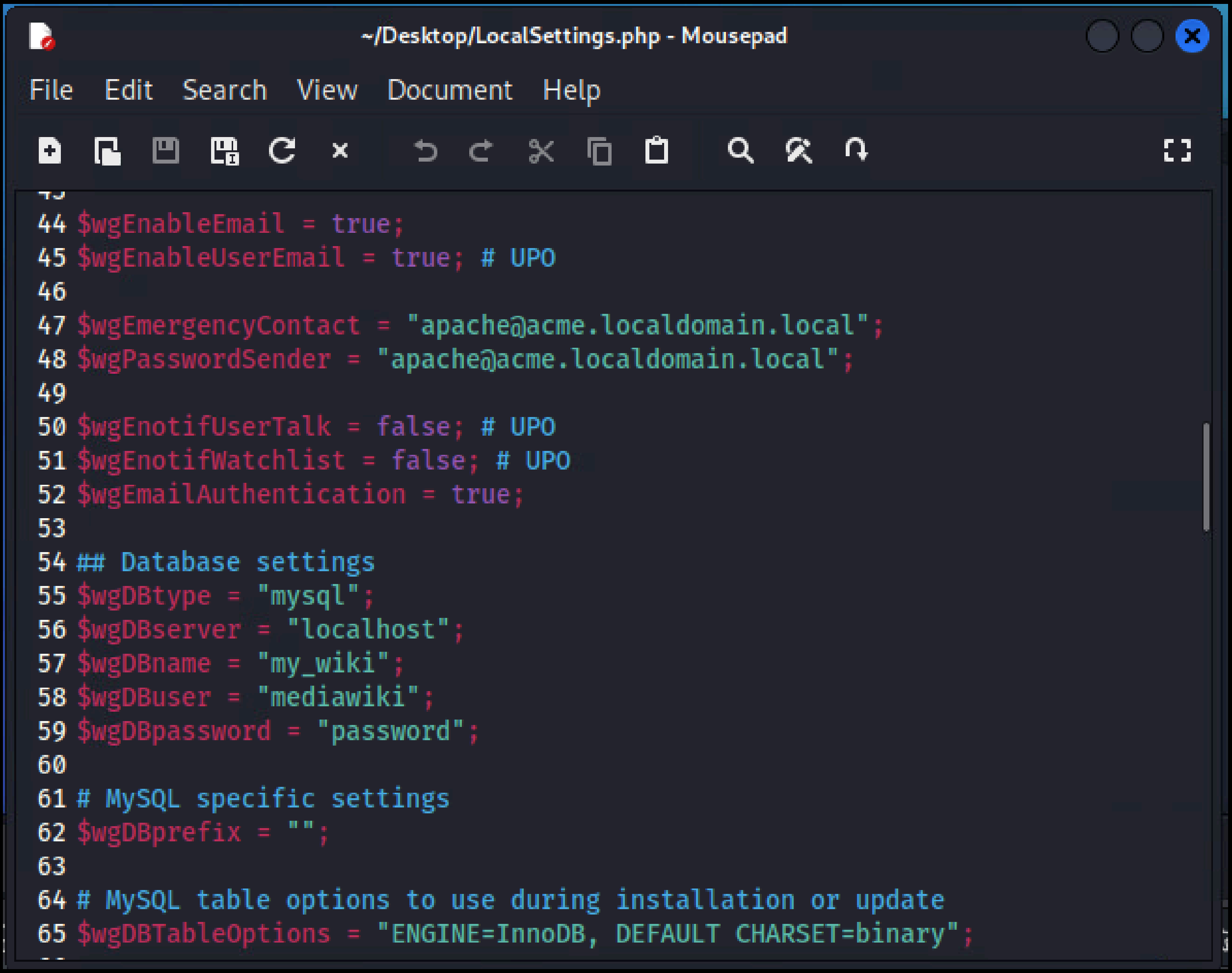
Access of ftp server method 1: nmap scans and nmap scripts

Another method that granted me ftp server access was through running nmap scripts. I went through the series of nmap scripts located at usr/share/nmap/ and that contained the string ftp. After running the ftp bruteforce script I was able to obtain the login credentials. Username: user and Password: user. This output can be seen below.


kali@kali2022: /usr/share/nmap/scripts
File Actions Edit View Help
kali@kali2022: /usr/share/nmap/scripts ×
kali@kali2022: ~/Desktop ×

```
NSE: [ftp-brute] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for acme (192.168.122.7)
Host is up (0.00013s latency).
Not shown: 990 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
| ftp-brute:
|   Accounts:
|     user:user - Valid credentials
|_  Statistics: Performed 3680 guesses in 602 seconds, average tps: 5.9
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
110/tcp  open  pop3
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
143/tcp  open  imap
445/tcp  open  microsoft-ds
2049/tcp open  nfs

Nmap done: 1 IP address (1 host up) scanned in 602.30 seconds

┌──(kali㉿kali2022)-[/usr/share/nmap/scripts]
└─$ 
```

The next step after finding this was to log into the ftp server. I used the following commands ftp->open->192.168.122.7->dir->. This led me to find a Mail file as well as a login for the database. Database name:my_wiki and password: password. **This find signifies that the ftp server is vulnerable and able to be accessed to find sensitive data.** In this case I was able to find the credentials for the login which will be used later. This is a high risk vulnerability because of the access it gives to the database on the server.

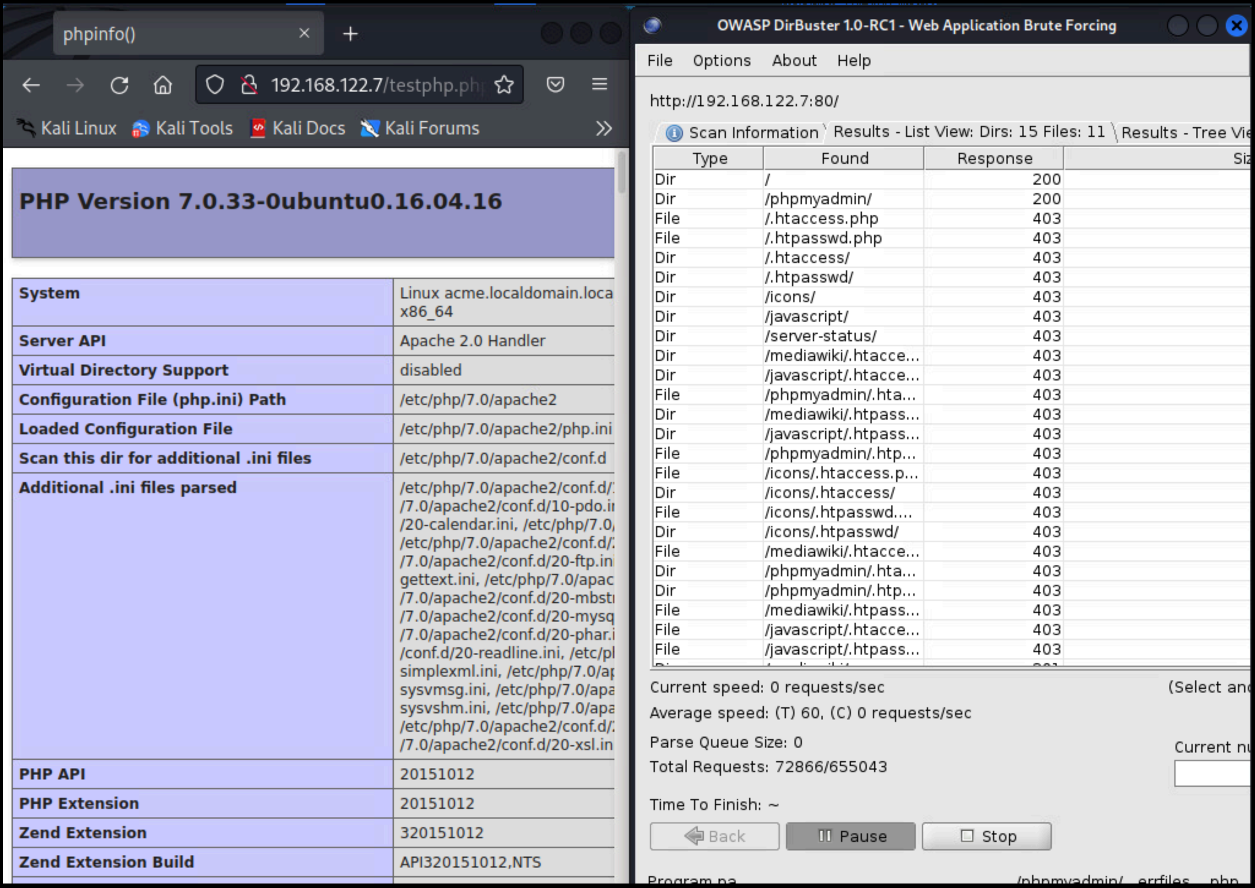


```
$wgEnableEmail = true;
$wgEnableUserEmail = true; # UPO

$wgEmergencyContact = "apache@acme.localdomain.local";
$wgPasswordSender = "apache@acme.localdomain.local";

$wgEnotifUserTalk = false; # UPO
$wgEnotifWatchlist = false; # UPO
$wgEmailAuthentication = true;

## Database settings
$wgDBtype = "mysql";
$wgDBserver = "localhost";
$wgDBname = "my_wiki";
$wgDBuser = "mediawiki";
$wgDBpassword = "password";

# MySQL specific settings
$wgDBprefix = "";

# MySQL table options to use during installation or update
$wgDBTableOptions = "ENGINE=InnoDB, DEFAULT CHARSET=binary";
```

PHP login credential from files on ftp server

Finding and logging into the php server

Because the http server is open, the next test was on the http port. I used the tool *dirbuster* to find any web applications that could be linked to the acme server. I ran dirbuster on the http://192.168.122.7:80 domain to find if there are any pages that can be accessed.



PHP Version 7.0.33-0ubuntu0.16.04.16

System	Linux acme.localdomain.loca x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/ /7.0/apache2/conf.d/10-pdo.i /20-calendar.ini, /etc/php/7.0, /etc/php/7.0/apache2/conf.d/ /7.0/apache2/conf.d/20-ftp.in gettext.ini, /etc/php/7.0/apac /7.0/apache2/conf.d/20-mbst /7.0/apache2/conf.d/20-mysq /7.0/apache2/conf.d/20-phar.i /conf.d/20-readline.ini, /etc/pl simplexml.ini, /etc/php/7.0/a sysvmsg.ini, /etc/php/7.0/apa sysvshm.ini, /etc/php/7.0/apa /etc/php/7.0/apache2/conf.d/ /7.0/apache2/conf.d/20-xsl.in
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS

http://192.168.122.7:80/

Type	Found	Response	Siz
Dir	/	200	
Dir	/phpmyadmin/	200	
File	/.htaccess.php	403	
File	/.htpasswd.php	403	
Dir	/.htaccess/	403	
Dir	/.htpasswd/	403	
Dir	/icons/	403	
Dir	/javascript/	403	
Dir	/server-status/	403	
Dir	/mediawiki/.htacce...	403	
Dir	/javascript/.htacce...	403	
File	/phpmyadmin/.hta...	403	
Dir	/mediawiki/.htpass...	403	
Dir	/javascript/.htpass...	403	
File	/phpmyadmin/.htp...	403	
File	/icons/.htaccess.p...	403	
Dir	/icons/.htaccess/	403	
File	/icons/.htpasswd....	403	
Dir	/icons/.htpasswd/	403	
File	/mediawiki/.htacce...	403	
Dir	/phpmyadmin/.hta...	403	
Dir	/phpmyadmin/.htp...	403	
File	/mediawiki/.htpass...	403	
File	/javascript/.htacce...	403	
File	/javascript/.htpass...	403	

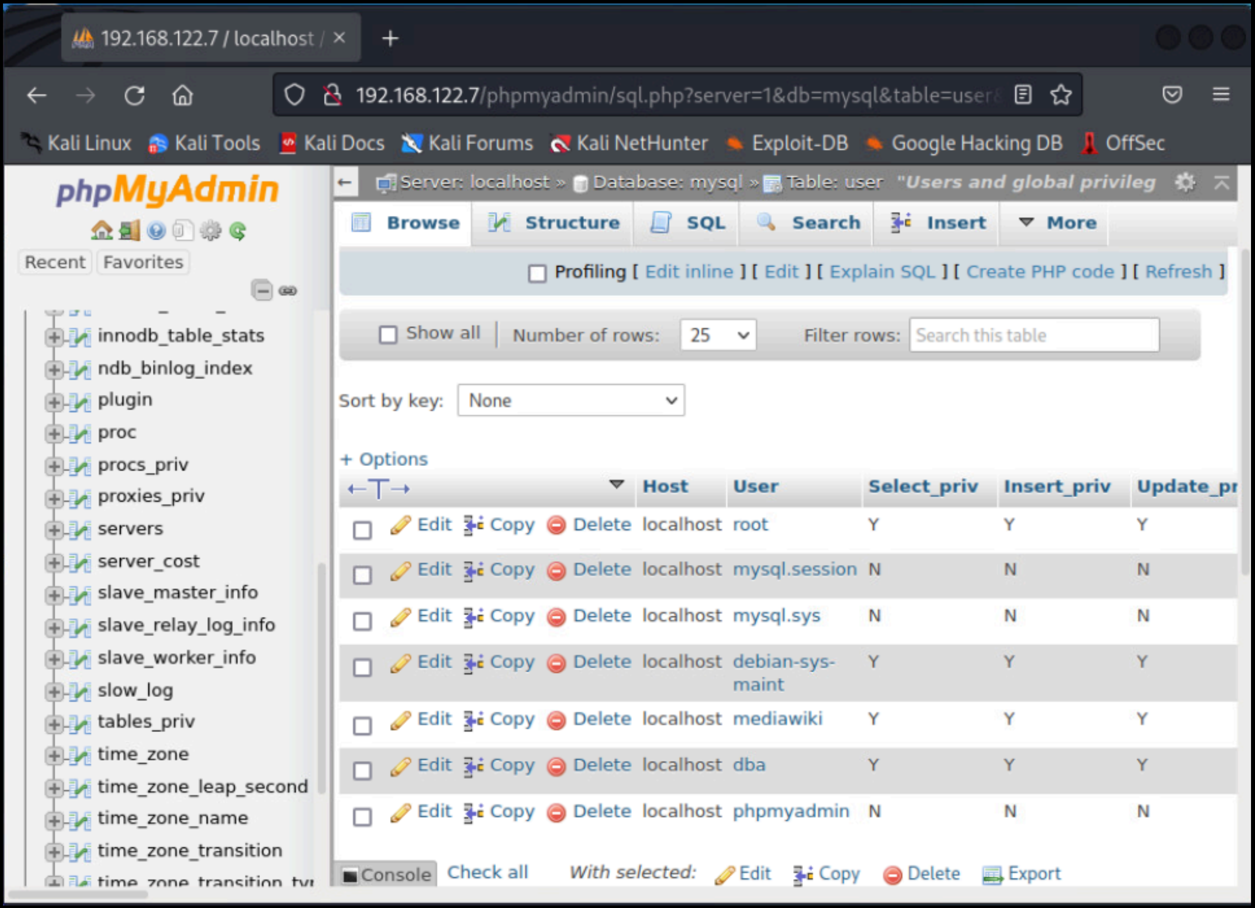
```
Current speed: 0 requests/sec                    (Select an
Average speed: (T) 60, (C) 0 requests/sec
Parse Queue Size: 0                              Current nu
Total Requests: 72866/655043
Time To Finish: ~
```

Dirbuster output

The result of this test was ability to find php versions which is a low to mild threat because by knowing the version, attackers can access a vulnerability for that version and gain access through it. I was also able to find the pages that I can access (ones with a response message of 200). I then navigated to http://192.168.122.7/phpmyadmin which led to me the php login page.

Unauthorized access into php login and database access (severe risk)

From the credential found above in the ftp server files, I used the login *username: my_wiki password: password* to access the database. This is considered a vulnerability because of the ability to see the rest of the users and edit user access. The image below shows the ability for attackers to remove or add users as they like.



192.168.122.7/phpmyadmin/sql.php?server=1&db=mysql&table=user

Server: localhost » Database: mysql » Table: user "Users and global privileg

Browse
Structure
SQL
Search
Insert
More
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
Check all
With selected: Edit Copy Delete Export
Console

innodb_table_stats
ndb_binlog_index
plugin
proc
procs_priv
proxies_priv
servers
server_cost
slave_master_info
slave_relay_log_info
slave_worker_info
slow_log
tables_priv
time_zone
time_zone_leap_second
time_zone_name
time_zone_transition
time_zone_transition_typ

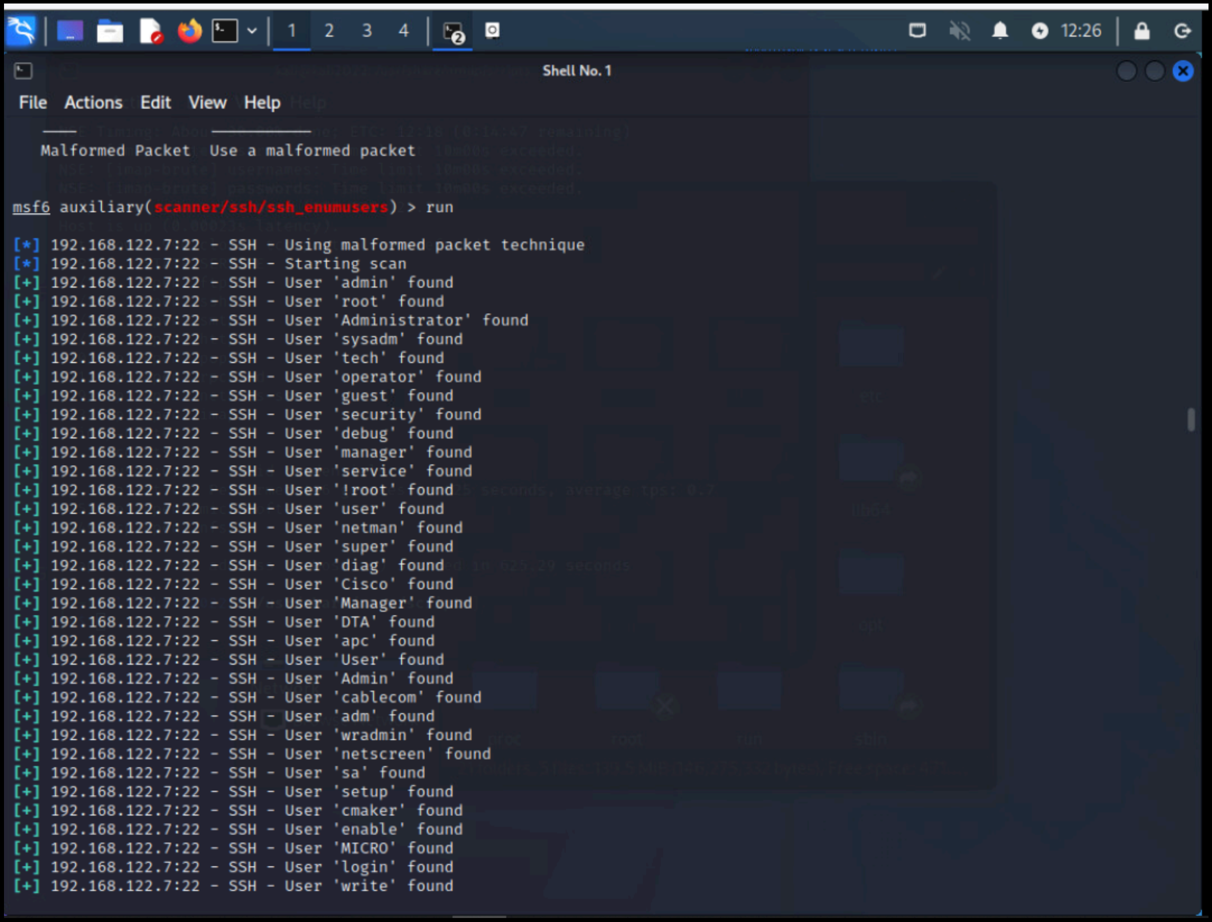
	Host	User	Select_priv	Insert_priv	Update_pr
□ Edit Copy Delete	localhost	root	Y	Y	Y
□ Edit Copy Delete	localhost	mysql.session	N	N	N
□ Edit Copy Delete	localhost	mysql.sys	N	N	N
□ Edit Copy Delete	localhost	debian-sys-maint	Y	Y	Y
□ Edit Copy Delete	localhost	mediawiki	Y	Y	Y
□ Edit Copy Delete	localhost	dba	Y	Y	Y
□ Edit Copy Delete	localhost	phpmyadmin	N	N	N

In the database the following sensitive data was found (No application of this data thus far):

- Username:41646d696e
- password : hashed password pbkdf2:sha512:30000:64:iHkfLaJBXzORWfwf3mH1Ww==:aKlrjsPW2uxVtVyMCwDo4qIFAANBHi4R44owGKrmG6zl1gSFfzentBM7Uw/GnpN8Sg+WQMAJqWrJDnPDLc4ACg==

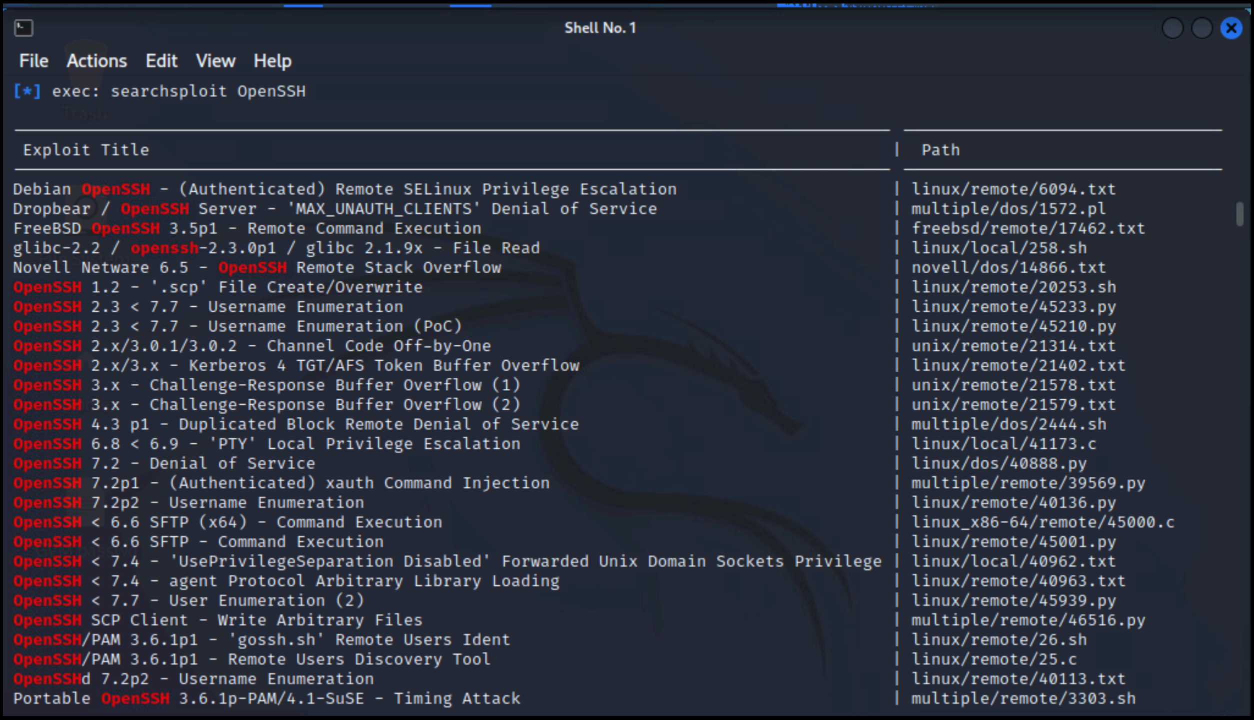
SSH and SMB client login

The next avenue that was tested was the ssh login. First metasploit was used to scan for all possible usernames that can be used for ssh login the result found over 50 logins shown below. This was done using the username enumeration scanner. From this valid usernames such as admin, root, tech, guest, and user were found



```
   ----             ----------
   Malformed Packet  Use a malformed packet

msf6 auxiliary(scanner/ssh/ssh_enumusers) > run

[*] 192.168.122.7:22 - SSH - Using malformed packet technique
[*] 192.168.122.7:22 - SSH - Starting scan
[+] 192.168.122.7:22 - SSH - User 'admin' found
[+] 192.168.122.7:22 - SSH - User 'root' found
[+] 192.168.122.7:22 - SSH - User 'Administrator' found
[+] 192.168.122.7:22 - SSH - User 'sysadm' found
[+] 192.168.122.7:22 - SSH - User 'tech' found
[+] 192.168.122.7:22 - SSH - User 'operator' found
[+] 192.168.122.7:22 - SSH - User 'guest' found
[+] 192.168.122.7:22 - SSH - User 'security' found
[+] 192.168.122.7:22 - SSH - User 'debug' found
[+] 192.168.122.7:22 - SSH - User 'manager' found
[+] 192.168.122.7:22 - SSH - User 'service' found
[+] 192.168.122.7:22 - SSH - User '!root' found
[+] 192.168.122.7:22 - SSH - User 'user' found
[+] 192.168.122.7:22 - SSH - User 'netman' found
[+] 192.168.122.7:22 - SSH - User 'super' found
[+] 192.168.122.7:22 - SSH - User 'diag' found
[+] 192.168.122.7:22 - SSH - User 'Cisco' found
[+] 192.168.122.7:22 - SSH - User 'Manager' found
[+] 192.168.122.7:22 - SSH - User 'DTA' found
[+] 192.168.122.7:22 - SSH - User 'apc' found
[+] 192.168.122.7:22 - SSH - User 'User' found
[+] 192.168.122.7:22 - SSH - User 'Admin' found
[+] 192.168.122.7:22 - SSH - User 'cablecom' found
[+] 192.168.122.7:22 - SSH - User 'adm' found
[+] 192.168.122.7:22 - SSH - User 'wradmin' found
[+] 192.168.122.7:22 - SSH - User 'netscreen' found
[+] 192.168.122.7:22 - SSH - User 'sa' found
[+] 192.168.122.7:22 - SSH - User 'setup' found
[+] 192.168.122.7:22 - SSH - User 'cmaker' found
[+] 192.168.122.7:22 - SSH - User 'enable' found
[+] 192.168.122.7:22 - SSH - User 'MICRO' found
[+] 192.168.122.7:22 - SSH - User 'login' found
[+] 192.168.122.7:22 - SSH - User 'write' found
```

The next step after was trying to find the password for the login. Because I already knew the ssh version from the initial recon: OpenSSH 7.2p2 I ran metasploit to find ssh exploits. Most of them applied to other versions but I was able to run a couple shown below.

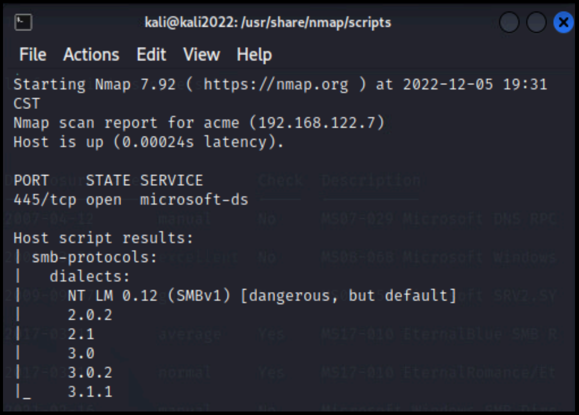


```
[*] exec: searchsploit OpenSSH

 ----------------------------------------------------------------------------------------  ---------------------------------
 Exploit Title                                                                            |  Path
 ----------------------------------------------------------------------------------------  ---------------------------------
Debian OpenSSH - (Authenticated) Remote SELinux Privilege Escalation                      | linux/remote/6094.txt
Dropbear / OpenSSH Server - 'MAX_UNAUTH_CLIENTS' Denial of Service                        | multiple/dos/1572.pl
FreeBSD OpenSSH 3.5p1 - Remote Command Execution                                          | freebsd/remote/17462.txt
glibc-2.2 / openssh-2.3.0p1 / glibc 2.1.9x - File Read                                    | linux/local/258.sh
Novell Netware 6.5 - OpenSSH Remote Stack Overflow                                        | novell/dos/14866.txt
OpenSSH 1.2 - '.scp' File Create/Overwrite                                                | linux/remote/20253.sh
OpenSSH 2.3 < 7.7 - Username Enumeration                                                  | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)                                            | linux/remote/45210.py
OpenSSH 2.x/3.0.1/3.0.2 - Channel Code Off-by-One                                         | unix/remote/21314.txt
OpenSSH 2.x/3.x - Kerberos 4 TGT/AFS Token Buffer Overflow                                | linux/remote/21402.txt
OpenSSH 3.x - Challenge-Response Buffer Overflow (1)                                      | unix/remote/21578.txt
OpenSSH 3.x - Challenge-Response Buffer Overflow (2)                                      | unix/remote/21579.txt
OpenSSH 4.3 p1 - Duplicated Block Remote Denial of Service                                | multiple/dos/2444.sh
OpenSSH 6.8 < 6.9 - 'PTY' Local Privilege Escalation                                      | linux/local/41173.c
OpenSSH 7.2 - Denial of Service                                                           | linux/dos/40888.py
OpenSSH 7.2p1 - (Authenticated) xauth Command Injection                                   | multiple/remote/39569.py
OpenSSH 7.2p2 - Username Enumeration                                                      | linux/remote/40136.py
OpenSSH < 6.6 SFTP (x64) - Command Execution                                              | linux_x86-64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execution                                                    | linux/remote/45001.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading                                  | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)                                                      | linux/remote/45939.py
OpenSSH SCP Client - Write Arbitrary Files                                                | multiple/remote/46516.py
OpenSSH/PAM 3.6.1p1 - 'gossh.sh' Remote Users Ident                                       | linux/remote/26.sh
OpenSSH/PAM 3.6.1p1 - Remote Users Discovery Tool                                         | linux/remote/25.c
OpenSSHd 7.2p2 - Username Enumeration                                                     | linux/remote/40113.txt
Portable OpenSSH 3.6.1p-PAM/4.1-SuSE - Timing Attack                                      | multiple/remote/3303.sh
```

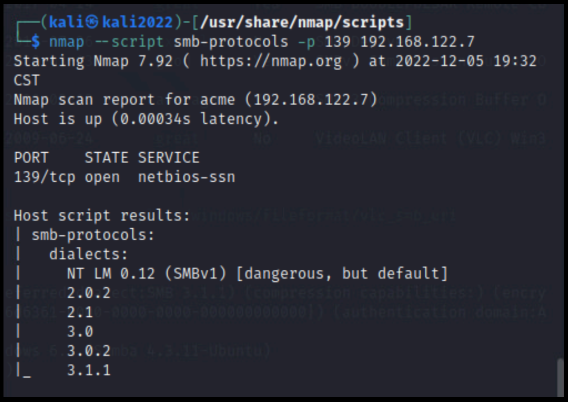
SSH exploits

None of these exploits resulted in any keys so I decided to move onto testing ports 139 and 445 SMB clients.

The first step taken was finding the SMB version by running an nmap script for port 445 and 139 the result of this scan were possible version of the SMB client for both ports and are shown below:

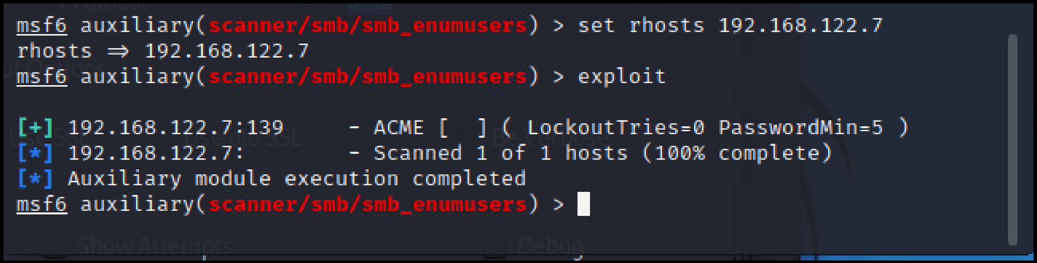

kali@kali2022: /usr/share/nmap/scripts
File Actions Edit View Help

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-05 19:31
CST
Nmap scan report for acme (192.168.122.7)
Host is up (0.00024s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-protocols:
|   dialects:
|     NT LM 0.12 (SMBv1) [dangerous, but default]
|     2.0.2
|     2.1
|     3.0
|     3.0.2
|_    3.1.1
```



```
┌──(kali㉿kali2022)-[/usr/share/nmap/scripts]
└─$ nmap --script smb-protocols -p 139 192.168.122.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-05 19:32
CST
Nmap scan report for acme (192.168.122.7)
Host is up (0.00034s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn

Host script results:
| smb-protocols:
|   dialects:
|     NT LM 0.12 (SMBv1) [dangerous, but default]
|     2.0.2
|     2.1
|     3.0
|     3.0.2
|_    3.1.1
```

The potential username for the SMB client was found using the metasploit scanner for enumerating smb users. The result was finding a user: ACME with a password length of at least 5 characters:

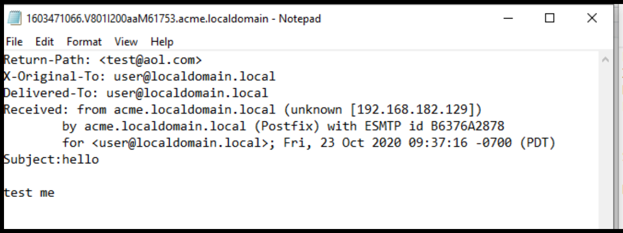


```
msf6 auxiliary(scanner/smb/smb_enumusers) > set rhosts 192.168.122.7
rhosts => 192.168.122.7
msf6 auxiliary(scanner/smb/smb_enumusers) > exploit

[+] 192.168.122.7:139     - ACME [  ] ( LockoutTries=0 PasswordMin=5 )
[*] 192.168.122.7:        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_enumusers) > 
```

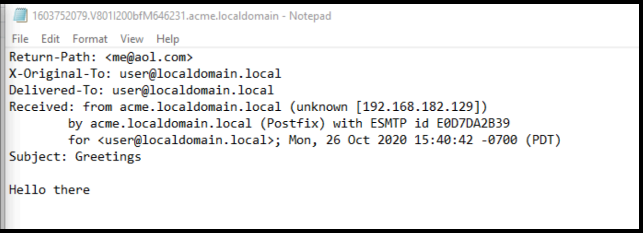
This finding could have been used more extensively to try to obtain the SMB login.

Additional finds:

- Emails in ftp server


1603471066.V801I200aaM61753.acme.localdomain - Notepad
File Edit Format View Help

```
Return-Path: <test@aol.com>
X-Original-To: user@localdomain.local
Delivered-To: user@localdomain.local
Received: from acme.localdomain.local (unknown [192.168.182.129])
        by acme.localdomain.local (Postfix) with ESMTP id B6376A2878
        for <user@localdomain.local>; Fri, 23 Oct 2020 09:37:16 -0700 (PDT)
Subject:hello

test me
```


1603752079.V801I200bfM646231.acme.localdomain - Notepad
File Edit Format View Help

```
Return-Path: <me@aol.com>
X-Original-To: user@localdomain.local
Delivered-To: user@localdomain.local
Received: from acme.localdomain.local (unknown [192.168.182.129])
        by acme.localdomain.local (Postfix) with ESMTP id E0D7DA2B39
        for <user@localdomain.local>; Mon, 26 Oct 2020 15:40:42 -0700 (PDT)
Subject: Greetings

Hello there
```

Final Summary

This report shed light on the vulnerabilities and ways that attackers might try to breach the system. Implications of breaches were outlined with possible data loss, access control, or loss of server functionality that can come from these vulnerabilities.

To remediate these issues the fpt server can be strengthened using an encrypted ftp server or FTPS. To secure the database and access into the php server, the php version should be updated constantly and sanitization should be used in the login to prevent sql injection. Cross site scripting should also be mitigate by not allowing remote code execution