|  |
| --- |
| CMPT 489 |
| Assignment 1 |
|  |

|  |
| --- |
| Marcelo Ollin Paco Zepeda  301180252 |

***Table of Contents***

[**Part One: Passive Information Gathering** 3](#_Toc19913168)

[**1.1: Open Source Intelligence** 3](#_Toc19913169)

[Task 1: 3](#_Toc19913170)

[**1.2: DNS Server Interrogating** 6](#_Toc19913171)

[**1.2.1: Basics** 6](#_Toc19913172)

[Task 2: 6](#_Toc19913173)

[Task 3: 6](#_Toc19913174)

[**1.2.2: Understanding Hierarchy** 6](#_Toc19913175)

[Task 4: 6](#_Toc19913176)

[Task 5: 7](#_Toc19913177)

[**Part Two: Active Information Gathering** 7](#_Toc19913178)

[**2.1: Network Mapping and Port Scanning** 7](#_Toc19913179)

[Task 6: 7](#_Toc19913180)

[Task 7: 8](#_Toc19913181)

[Task 8: 9](#_Toc19913182)

[Task 9: 10](#_Toc19913183)

[Task 10: 10](#_Toc19913184)

[**2.2 Vulnerability Scanning** 13](#_Toc19913185)

[Task 11: 13](#_Toc19913186)

# **Part One: Passive Information Gathering**

## **1.1: Open Source Intelligence**

#### Task 1:

Select a technology company that you have never heard of. Perform a thorough passive information gathering on the selected company and present your results in a brief report. Include your methodology and rationale in information gathering. Report the results of using at least two open source intelligence tools such as Maltego, theHarvester in your information gathering process.

**Note:** For the passive information gather section, I chose the technology company **eventbase** as the target.

Using Maltego, I was able to construct a graph containing information on the company eventbase. Figure 1 provides a high-level view of all the information gathered using basic transforms.

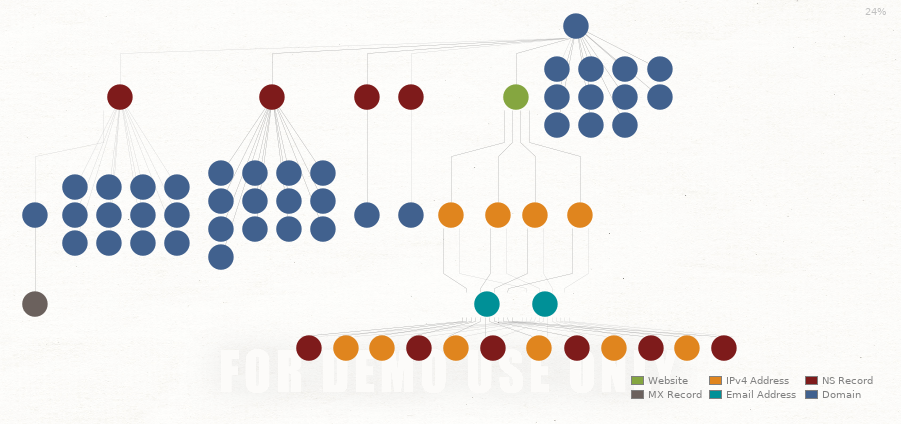
****

Figure 1

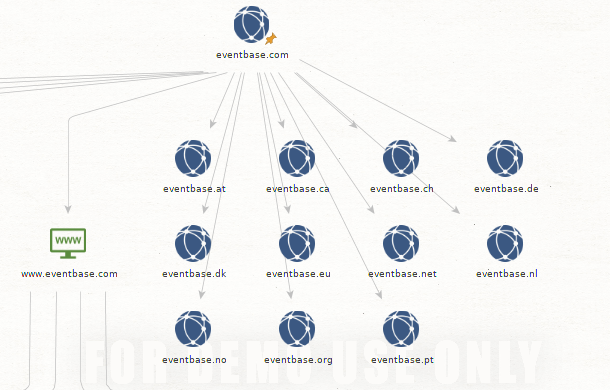


Figure 2

In figure 2, I used the “to website” transform, to look up the website for eventbase and verify that it actually exists. We can tell that the website does in fact exists since the transform ran successfully and a new node in the shape of a green monitor appeared on the graph.

Moreover, using the “to domain” transform, I was able to get all the sub-domains that belong to eventbase. We can see that they have different domains for different geographical regions.

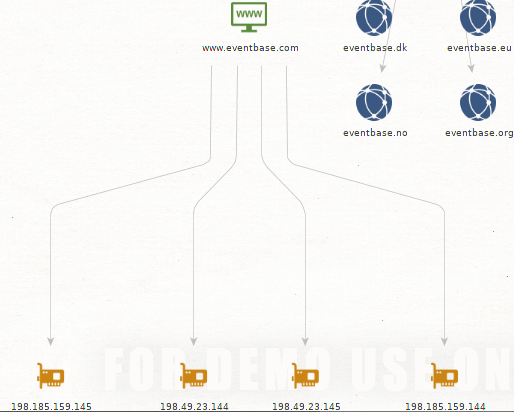
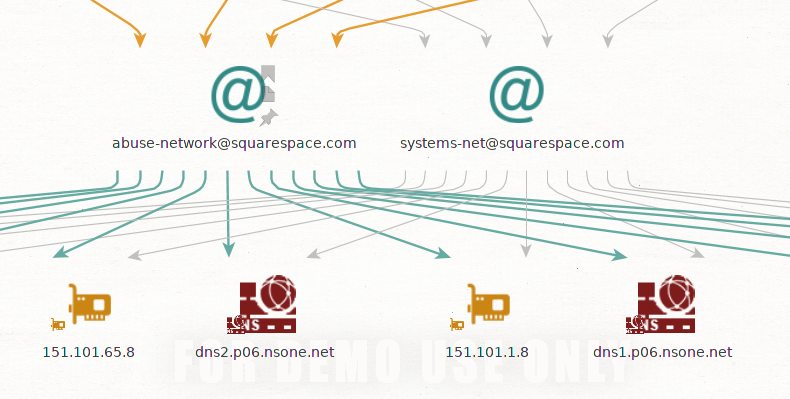
Figure 3 uses the transform “to IP” to obtain the IP addresses of the host website. The transform returns four IP addresses that belong to the eventbase domain. From here, the next step I took was to look at the email addresses that belonged to the four IP addresses the transform returned. The four IP addresses mapped back to two email addresses. The result is show in figure 4 below.

Figure 4

Figure 3

To gather more information about eventbase, I used theHarverster tool to find a series of employee emails, hosts and a list of employee’s names and their role at the company.

Using the following theHarverster command:

theharvester -d eventbase.com -l 100 -b google -h results.html

I got information such as employee emails and hosts that were found using google as a data source.

[+] Emails found:

------------------

info@eventbase.com

kevin.lorch@eventbase.com

marketing@eventbase.com

jobs@eventbase.com

sapsupport@eventbase.com

sarah.blackmore@eventbase.com

last@eventbase.com

john@eventbase.com

support@eventbase.com

event\_tech@eventbase.com

[+] Hosts found in search engines:

------------------------------------

Total hosts: 4

[-] Resolving hostnames IPs...

free.eventbase.com:100.24.231.193

live.eventbase.com:52.201.101.201

rsa1-webservice.eventbase.com:184.73.41.244

www.eventbase.com:198.185.159.144\

Editing theHarvester command to use LinkedIn as a data source instead got us a list of employee names and their role at eventbase:

theharvester -d eventbase.com -l 100 -b linkedin -h results.html

Users from Linkedin:

-------------------

Ben West - Co-Founder - Eventbase

Sanit Jain - Senior Product Manager - Eventbase

Miriam Trotscha - Account Manager - Eventbase

Stacey Louie - Intermediate Accountant - Eventbase

Angela Stogre - Vice President Finance - Eventbase

April Andrews - Senior Account Manager - Eventbase

Chris Seto - Backend Software Developer - Pixieset

Hannah Coffey - Product Manager - Eventbase

Stephanie Forbes - Producer - Eventbase

Vivian Lau - Associate Producer - Eventbase

Savannah Boyd - Producer - Eventbase

Sharon Chong - Android Developer - VRIFY

Won Ng - Chief Operating Officer - Canalyst

Tavis Paquette - Lead DevOps Engineer - Eventbase

Curtis Strome - Product Manager - Lendesk

Kosta S. - Python Developer - Eventbase

AJ Brigden - Android Developer - Eventbase

Michelle Osborne - Producer - Eventbase

Shayne J. - Lead QA Engineer - Evolve Biologix

Callum Davies - Associate Tech Lead - Eventbase

Kevin Chen - Mobile Developer - Eventbase

Eugene Chong - Technical Lead - Eventbase

Brendan DeBrincat - IT Systems - Eventbase

James Kelly - Quality Assurance Analyst - Eventbase

Sukwhan Chung - Associate Producer - Eventbase

Sophie Donnison - Producer - Eventbase

Christina Looker - Exploratory QA - HSBC

Abhiraj Bhatia - Foundation QA Lead - Eventbase

Himani Dutta - Quality Assurance Analyst - Eventbase

Will Nguyen - Frontend Web Developer - Eventbase

Jenna Cho - Producer - Eventbase

Kyle Wang - Technical Product Specialist - Eventbase

Lovedeep Malik - Front End Developer - Eventbase

Vanessa Lai - Events - Eventbase

Jas Rowinski - Devops Engineer - Eventbase

Andrew Whitman - Technical Lead - Eventbase

Ashli Ahn - UI Designer - Eventbase

Jay Tollefson - Solutions Architect - Eventbase

Luke Basso - Senior Software Engineer - realtor.com

Nicole Farley - Producer - Eventbase

Jesse Scott - Senior Android Engineer - Mojio

Stephen Tynan - Producer - Eventbase

Jared Zecchel - UI Designer - Eventbase

Angela Stogre - Vice President Finance - Eventbase

Lyndsay Imrie - Account Executive - Eventbase

Kasey Sherwood - Co-Founder - irevu

Jomar Santos - Web Developer - Eventbase

#### 

## **1.2: DNS Server Interrogating**

### **1.2.1: Basics**

#### Task 2:

Using dig find the IP address of [www.sfu.com](http://www.sfu.com/). What is the IP address?

dig www.sfu.com

; <<>> DiG 9.11.5-P4-5.1+b1-Debian <<>> www.sfu.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18640

;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;www.sfu.com.   IN    A

;; ANSWER SECTION:

www.sfu.com.   86400    IN CNAME    www.sfu.ca.

www.sfu.ca.   250    IN A   142.58.102.68 🡸 **This is the IP Address**

;; Query time: 126 msec

;; SERVER: 10.0.0.1#53(10.0.0.1)

;; WHEN: Tue Sep 17 12:59:01 PDT 2019

;; MSG SIZE  rcvd: 80

#### Task 3:

The returned answer from the previous task includes a CNAME part. What does this mean?

**CNAME** means canonical name, which is the properly denoted host name of a computer or a network server.

### **1.2.2: Understanding Hierarchy**

#### Task 4:

Run a query to ask a root server about **mail.sfu.ca** without using recursion (Hint use the @ for directing the query to a specific root server). What command did you use? What is the result of the query?

The following command was used to as a root server about mail.sfu.ca:

dig +norecurse @a.root-servers.net mail.sfu.ca

This is the result of the query:

; <<>> DiG 9.11.5-P4-5.1+b1-Debian <<>> +norecurse @a.root-servers.net mail.sfu.ca

; (2 servers found)

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12812

;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 9

;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 1472

;; QUESTION SECTION:

;mail.sfu.ca.   IN    A

;; AUTHORITY SECTION:

ca.   172800    IN NS c.ca-servers.ca.

ca.   172800    IN NS j.ca-servers.ca.

ca.   172800    IN NS x.ca-servers.ca.

ca.   172800    IN NS any.ca-servers.ca.

;; ADDITIONAL SECTION:

c.ca-servers.ca.    172800 IN A 185.159.196.2

j.ca-servers.ca.    172800 IN A 198.182.167.1

x.ca-servers.ca.    172800 IN A 199.253.250.68

any.ca-servers.ca.    172800 IN A 199.4.144.2

c.ca-servers.ca.    172800 IN AAAA    2620:10a:8053::2

j.ca-servers.ca.    172800 IN AAAA    2001:500:83::1

x.ca-servers.ca.    172800 IN AAAA    2620:10a:80ba::68

any.ca-servers.ca.    172800 IN AAAA   2001:500:a7::2

;; Query time: 93 msec

;; SERVER: 198.41.0.4#53(198.41.0.4)

;; WHEN: Tue Sep 17 12:54:52 PDT 2019

;; MSG SIZE  rcvd: 293

#### Task 5:

The answer to the previous task will not give you the IP address of **mail.sfu.ca**. Instead follow the “path” down in the hierarchy of the nameservers to find the address of **mail.sfu.ca** without using recursion. What commands did you use? What is the IP you found?

Commands used to find the IP address of **mail.sfu.ca** in order:

dig +norecurse @a.root-servers.net mail.sfu.ca

dig +norecurse @any.ca-servers.ca mail.sfu.ca

dig +norecurse @whistler.sfu.ca mail.sfu.ca

The last command gives us the IP address of mail.sfu.ca in the answer section of the command output:

;; ANSWER SECTION:

mail.sfu.ca.   300    IN A   142.58.225.1 🡸 **This is the IP Address**

# **Part Two: Active Information Gathering**

## **2.1: Network Mapping and Port Scanning**

#### Task 6:

What is the IP address of the local network in the form of IP/netmask? What command did you use to find this?

The following is the IP address of the local network in the form of IP/netmask:

Network:   10.0.0.0/24

For this task, I used **ifconfig** to get IP address assigned to my computer by the local network. Then used **ipcalc** to get the IP address of the local network in the form of IP/netmask

ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500

     inet 10.0.0.16  netmask 255.255.255.0  broadcast 10.0.0.255

ipcalc 10.0.0.16

Address:   10.0.0.16       00001010.00000000.00000000. 00010000

Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000

Wildcard:  0.0.0.255       00000000.00000000.00000000. 11111111

=>

Network:   10.0.0.0/24      00001010.00000000.00000000. 00000000

HostMin:   10.0.0.1       00001010.00000000.00000000. 00000001

HostMax:   10.0.0.254       00001010.00000000.00000000. 11111110

Broadcast: 10.0.0.255       00001010.00000000.00000000. 11111111

Hosts/Net: 254               Class A, Private Internet

#### Task 7:

Perform a **full ping** scan in the local network using Nmap and identify all potential targets. Report the results of the scan and point the IPs of the potential target machines. What commands did you use to scan the network?

Command used to scan the network:

nmap 10.0.0.0/24

The IPs of the potential target machines are pointed at with 🡸 **Target IP Address** in the result output below:

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-20 12:27 PDT

Nmap scan report for pfSense.localdomain (10.0.0.1) 🡸 **Target IP Address**

Host is up (0.00048s latency).

Not shown: 998 filtered ports

PORT   STATE SERVICE

53/tcp open  domain

80/tcp open  http

MAC Address: 08:00:27:34:4F:F8 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.0.0.15 🡸 **Target IP Address**

Host is up (0.00045s latency).

Not shown: 987 closed ports

PORT  STATE SERVICE

135/tcp   open msrpc

139/tcp   open netbios-ssn

445/tcp   open microsoft-ds

554/tcp   open rtsp

2869/tcp  open icslap

5357/tcp  open wsdapi

10243/tcp open  unknown

49152/tcp open  unknown

49153/tcp open  unknown

49154/tcp open  unknown

49155/tcp open  unknown

49156/tcp open  unknown

49157/tcp open  unknown

MAC Address: 08:00:27:6C:52:84 (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.0.0.17 🡸 **Target IP Address**

Host is up (0.00068s latency).

Not shown: 997 closed ports

PORT STATE SERVICE

135/tcp open  msrpc

139/tcp open  netbios-ssn

445/tcp open  microsoft-ds

MAC Address: 08:00:27:13:52:EB (Oracle VirtualBox virtual NIC)

Nmap scan report for 10.0.0.16

Host is up (0.000012s latency).

Not shown: 999 closed ports

PORT STATE SERVICE

111/tcp open  rpcbind

Nmap done: 256 IP addresses (4 hosts up) scanned in 10.63 seconds

#### Task 8:

Perform a TCP SYN scan on a specific target using Nmap. Report the result. What command did you use to perform the scan? Perform a TCP full scan in a specific target **different** than the target you used for TCP SYN scan. Report the result. What command did you use to perform the scan? What is the difference between this method of scanning and the one that you used for TCP SYN scan?

Command used to do a TPC SYN scan on target machine with IP 10.0.0.17:

nmap -sS 10.0.0.17

Result output:

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-20 12:33 PDT

Nmap scan report for 10.0.0.17

Host is up (0.00044s latency).

Not shown: 997 closed ports

PORT STATE SERVICE

135/tcp open  msrpc

139/tcp open  netbios-ssn

445/tcp open  microsoft-ds

MAC Address: 08:00:27:13:52:EB (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.39 seconds

Command used to do TCP full scan on target machine with IP 10.0.0.15:

nmap -sT 10.0.0.15

Result output:

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-20 13:55 PDT

Nmap scan report for 10.0.0.15

Host is up (0.024s latency).

Not shown: 987 closed ports

PORT  STATE SERVICE

135/tcp   open msrpc

139/tcp   open netbios-ssn

445/tcp   open microsoft-ds

554/tcp   open rtsp

2869/tcp  open icslap

5357/tcp  open wsdapi

10243/tcp open  unknown

49152/tcp open  unknown

49153/tcp open  unknown

49154/tcp open  unknown

49155/tcp open  unknown

49156/tcp open  unknown

49157/tcp open  unknown

MAC Address: 08:00:27:6C:52:84 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.85 seconds

The main difference between performing a TCP SYN scan vs a TPC full scan, is that a TCP full scan is easily detectible since the target host logs will show a bunch of connection and error messages for the services which accept() the connection just have it immediately shut down. Where as a TCP SYN is harder to trace since fewer sites log it.

#### Task 9:

Perform two full port scanning in two different targets separately. Report the results. Can you infer the operating system from these results? If yes, indicate how. If not explain why.

Full scan on target machine with IP 10.0.0.15:

nmap -p0-65535 10.0.0.15

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-20 15:43 PDT

Nmap scan report for 10.0.0.15

Host is up (0.00022s latency).

Not shown: 65523 closed ports

PORT  STATE SERVICE

135/tcp   open msrpc

139/tcp   open netbios-ssn

445/tcp   open microsoft-ds

554/tcp   open rtsp

2869/tcp  open icslap

5357/tcp  open wsdapi

10243/tcp open  unknown

49152/tcp open  unknown

49153/tcp open  unknown

49154/tcp open  unknown

49155/tcp open  unknown

49156/tcp open  unknown

49157/tcp open  unknown

MAC Address: 08:00:27:6C:52:84 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 18.20 seconds

Full scan on target machine with IP 10.0.0.17:

nmap -p0-65535 10.0.0.17

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-20 15:46 PDT

Nmap scan report for 10.0.0.17

Host is up (0.00030s latency).

Not shown: 65533 closed ports

PORT STATE SERVICE

135/tcp open  msrpc

139/tcp open  netbios-ssn

445/tcp open  microsoft-ds

MAC Address: 08:00:27:13:52:EB (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 17.16 seconds

You can infer the type operating system the target machines are running from this output (ie. Windows). However, you cannot tell what version of the operating system they are running.

#### Task 10:

There are different ways to identify a target’s operating system. Using Nmap show **two** different ways to do that. Execute these for both of the target machines. In total there should be **four** results (two for the first target and two for the second). Report the results and associate the IPs with the operating systems.

The two commands used to identify the target’s OS:

nmap -sV [IP Address]

nmap -p0-65535 -A -T4 [IP Address]

Result output on target machine with IP address 10.0.0.15:

nmap -sV 10.0.0.15

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-20 15:53 PDT

Nmap scan report for 10.0.0.15

Host is up (0.00034s latency).

Not shown: 987 closed ports

PORT  STATE SERVICE  VERSION

135/tcp   open msrpc    Microsoft Windows RPC

139/tcp   open netbios-ssn  Microsoft Windows netbios-ssn

445/tcp   open microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)

554/tcp   open rtsp?

2869/tcp  open http     Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

5357/tcp  open http     Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

10243/tcp open  http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

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

49157/tcp open  msrpc Microsoft Windows RPC

MAC Address: 08:00:27:6C:52:84 (Oracle VirtualBox virtual NIC)

Service Info: Host: ADMIN-PC; OS: Windows; 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 129.24 seconds

nmap -p0-65535 -A -T4 10.0.0.15

Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-20 15:57 PDT

Nmap scan report for 10.0.0.15

Host is up (0.00097s latency).

Not shown: 65523 closed ports

PORT  STATE SERVICE  VERSION

135/tcp   open msrpc    Microsoft Windows RPC

139/tcp   open netbios-ssn  Microsoft Windows netbios-ssn

445/tcp   open microsoft-ds Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)

554/tcp   open rtsp?

|\_rtsp-methods: ERROR: Script execution failed (use -d to debug)

2869/tcp  open http     Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

5357/tcp  open http     Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

|\_http-server-header: Microsoft-HTTPAPI/2.0

|\_http-title: Service Unavailable

10243/tcp open  http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

|\_http-server-header: Microsoft-HTTPAPI/2.0

|\_http-title: Not Found

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

49157/tcp open  msrpc Microsoft Windows RPC

MAC Address: 08:00:27:6C:52:84 (Oracle VirtualBox virtual NIC)

Device type: general purpose|media device

Running: Microsoft Windows 2008|10|7|8.1, Microsoft embedded

OS CPE: cpe:/o:microsoft:windows\_server\_2008::sp2 cpe:/o:microsoft:windows\_10 cpe:/h:microsoft:xbox\_one cpe:/o:microsoft:windows\_7::- cpe:/o:microsoft:windows\_7::sp1 cpe:/o:microsoft:windows\_8 cpe:/o:microsoft:windows\_8.1

OS details: Microsoft Windows Server 2008 SP2 or Windows 10 or Xbox One, Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1

Network Distance: 1 hop

Service Info: Host: ADMIN-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:

|\_clock-skew: mean: 2h19m59s, deviation: 4h02m29s, median: 0s

|\_nbstat: NetBIOS name: ADMIN-PC, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:6c:52:84 (Oracle VirtualBox virtual NIC)

| smb-os-discovery:

|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)

|   OS CPE: cpe:/o:microsoft:windows\_7::sp1:professional

|   Computer name: admin-PC

|   NetBIOS computer name: ADMIN-PC\x00

|   Workgroup: WORKGROUP\x00

|\_  System time: 2019-09-20T15:59:19-07:00

| smb-security-mode:

|   account\_used: guest

|   authentication\_level: user

|   challenge\_response: supported

|\_  message\_signing: disabled (dangerous, but default)

| smb2-security-mode:

|   2.02:

|\_ Message signing enabled but not required

| smb2-time:

|   date: 2019-09-20T22:59:19

|\_  start\_date: 2019-09-20T17:34:49

TRACEROUTE

HOP RTT ADDRESS

1   0.97 ms 10.0.0.15

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

Nmap done: 1 IP address (1 host up) scanned in 211.78 seconds

Shortened result output on target machine with IP address 10.0.0.17:

nmap -sV 10.0.0.17

...

PORT STATE SERVICE  VERSION

135/tcp open  msrpc Microsoft Windows RPC

139/tcp open  netbios-ssn Microsoft Windows netbios-ssn

445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds

...

nmap -p0-65535 -A -T4 10.0.0.17

...

Running: Microsoft Windows XP|2003

OS CPE: cpe:/o:microsoft:windows\_xp cpe:/o:microsoft:windows\_server\_2003

OS details: Microsoft Windows XP SP2 or SP3, or Windows Server 2003

...

MAC: 08:00:27:13:52:eb (Oracle VirtualBox virtual NIC)

| smb-os-discovery:

|   OS: Windows XP (Windows 2000 LAN Manager)

|   OS CPE: cpe:/o:microsoft:windows\_xp::-

|   Computer name: admin-2bdbd2ba8

|   NetBIOS computer name: ADMIN-2BDBD2BA8\x00

|   Workgroup: WORKGROUP\x00

|\_  System time: 2019-09-20T16:02:58-07:00

...

IP address 10.0.0.15 has operating system: Windows 7

IP address 10.0.0.17 has operating system: Windows XP

## **2.2 Vulnerability Scanning**

#### Task 11:

Perform an advanced scan on the Windows XP target machine. Report the critical vulnerabilities of the system. Which of these could be used directly to exploit and gain access to the target system and which to gain more info or perform a denial of service attack according to your opinion?

There are five critical vulnerabilities on the Windows XP target machine:

* **MS08-067**: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Execution (958644) (ECLIPSEDWING) (uncredentialed check)
* **MS09-001**: Microsoft Windows SMB Vulnerabilities Remote Code Execution (958687) (uncredentialed check)
* **MS17-010**: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
* Unsupported Windows OS (remote)
* Microsoft Windows XP Unsupported Installation Detection

In my opinion, **MS08-067, MS09-001** and **MS17-010** can be used to exploit and gain access to the target system. However, I believe that only **MS17-010** can be used to gather more information on the target system. As for denial of service attacks, **MS09-001** and **MS17-010** can be used.