Man-in-the-Middle Detection

we will cover evidence of three chained MITM techniques

- ARP Spoofing (network interception) Address Resolution Protocol responsible for finding the MAC "who has this IP"
- DNS Spoofing (redirection)
- SSL Stripping (credential capture)

They intercept by exploiting weaknesses in network protocols or by using techniques like ARP, DNS, or IP spoofing

Common Types of MITM Attacks

- Packet sniffing
- Session hijacking
- SSL stripping
- DNS spoofing
- IP spoofing
- Rogue Wi-Fi access point

Once attackes established a MITM position, they control the data stream

Example:

- can infect a browser exploit
- malware dropper
- remote access trojan (RAT)
-

Detecting ARP Spoofing

In ARP spoofing, attacker send fakes ARP reply

Why ARP spoofing works.

ARP has no authentication.
any device can send unsolicited "is-at"

Indicators of the Attack

- Duplicate MAC-to-IP Mappings
- Unsolicited ARP Replies → high volume of ARP replies without matching requests
- Abnormal ARP traffic Volume → A large number of ARP packets in short intervals
- Unusual Traffic Routing → Traffic rerouted through the attacker's MAC
- Gateway Redirection Patterns → Multiple destination MACs for the same gateway IP
- ARP Probe/Reply Loops → same pattern

Questions (Wireshark)

1) How many ARP packets from the gateway MAC Address were observed?

arp && arp.src.proto_ipv4 == 192.168.10.1 && eth.src == 02:aa:bb:cc:00:01

2) What MAC address was used by the attacker to impersonate the gateway?

arp.opcode == 2 && arp.src.proto_ipv4 == 192.168.10.1

3) How many gratuitous arp replies were observed for 192.168.10.1?

arp.isgratuitous && arp.src.proto_ipv4 == 192.168.10.1

A suspicious host sends many unsolicited (gratuitious) ARP replies

Repeated gratuitious ARPs can indicate an attacker maintaining their poison state

4) How many unique mac addresses claimed the same IP (192.168.10.1)?

arp.opcode == 2 && arp.src.proto.ipv4 == "192.168.10.1"

5) How many ARP spoofing packet were observed in total from the attacker?

arp.opcode == 2 && arp.src.proto.ipv4 == "192.168.10.1" && eth.src == 02.fe.fe.fe.55.55

Unmasking DNS Spoofing

How it works:

1) The Victim tries to visit their bank at "domain"
2) Attacker, who is already on the local network, intercepts the victim's DNS query
3) Attackers quickly sends a fake DNS response to the victim.
4) Victims's computer cache that DNS, when they try to connect it unknowingly connect to the attacker's server, which might host replica of the ~~domain~~ website.

Indicators:

- Multiple DNS responses for the same query
- DNS response from an unexpected source → doesn't match any configured resolver
- Suspicously short TTL (Time-to-live) values
- Unsolicited DNS responses

Question

1) How many DNS responses were observed for the domain corp-login.acme-corp.local?

dns.flags.response == 1 && dns.qry.name == "name"

2) How many DNS requests were observed from the IPs other than 8.8.8.8?

dns.flags.response == 1 && ip.src == 8.8.8.8 && dns.qry.name == "name"

and subtracted

3) what IP did the attacker's forged DNS response return for the domain?

192.168.10.55

Spotting SSL Stripping in Action

- to remove or prevent TLS encryption

This causes the client to use HTTP not HTTPS

How it works:

→ The victim initiates an HTTPS request to a website
→ Attacker ~~intere~~ intercepts to request
→ The attacker connects to the website over HTTPS but relays the responses to the victim
→ The victim unknowingly interacts over HTTP

Indicators:

- Initial request vs response
  ↳ initial request → 443 https
  immediately shift → 80 https

- Redirect / Link Rewriting
- Certificate Errors

Questions

1) How many POST requests were observed for our domain corp-login.acme-corp.local?

http && ip.dst == 192.168.10.55

2) What's the password of the victim found in the plaintext after successful ssl stripping attack?

Secret123!