DNS Spoofing

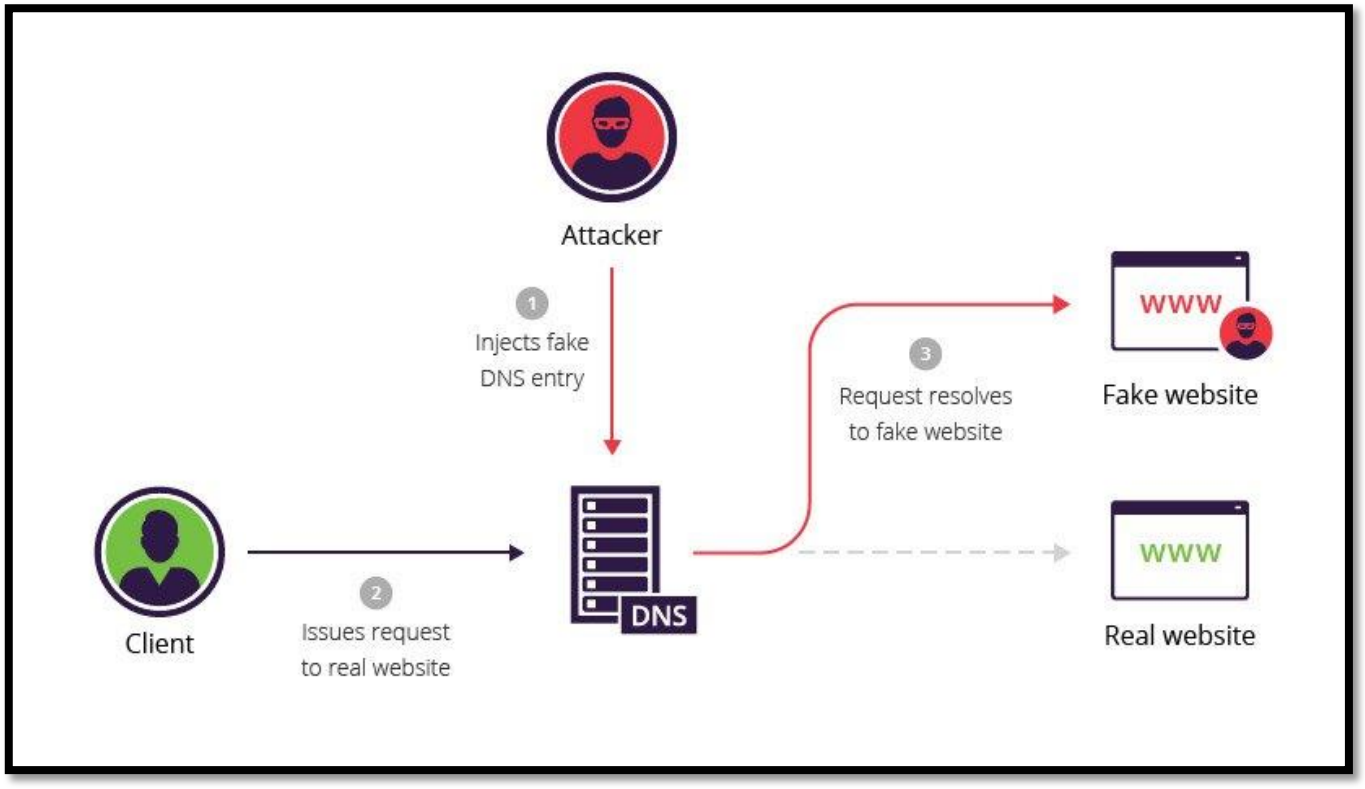
What is DNS Spoofing?

Domain Name Server (DNS) spoofing (a.k.a. DNS cache poisoning) is an attack in which altered DNS records are used to redirect online traffic to a fraudulent website that resembles its intended destination.

Once there, users are prompted to login into (what they believe to be) their account, giving the perpetrator the opportunity to steal their access credentials and other types of sensitive information. Furthermore, the malicious website is often used to install worms or viruses on a user's computer, giving the perpetrator long-term access to it and the data it stores.

Methods for executing a DNS spoofing attack include:

1. Man in the middle (MITM) – The interception of communications between users and a DNS server in order to route users to a different/malicious IP address.
2. DNS server compromise – The direct hijacking of a DNS server, which is configured to return a malicious IP address.



DNS cache poisoning example

The following example illustrates a DNS cache poisoning attack, in which an attacker (IP 192.168.3.300) intercepts a communication channel between a client (IP 192.168.1.100) and a server computer belonging to the website www.estores.com (IP 192.168.2.200).

In this scenario, a tool (e.g., arpspoof) is used to dupe the client into thinking that the server IP is 192.168.3.300. At the same time, the server is made to think that the client's IP is also 192.168.3.300.

Such a scenario would proceed as follows:

1. The attacker uses arpspoof to issue the command: arpspoof 192.168.1.100 192.168.2.200. This modifies the MAC addresses in the server's ARP table, causing it to think that the attacker's computer belongs to the client.
2. The attacker once again uses arpspoof to issue the command: arpspoof 192.168.2.200 192.168.1.100, which tells the client that the perpetrator's computer is the server.
3. The attacker issues the Linux command: echo 1> /proc/sys/net/ipv4/ip_forward. As a result, IP packets sent between the client and server are forwarded to the perpetrator's computer.
4. The host file, 192.168.3.300 estores.com is created on the attacker's local computer, which maps the website www.estores.com to their local IP.
5. The perpetrator sets up a web server on the local computer's IP and creates a fake website made to resemble www.estores.com.
6. Finally, a tool (e.g., dnsspoof) is used to direct all DNS requests to the perpetrator's local host file. The fake website is displayed to users as a result and, only by interacting with the site, malware is installed on their computers.

DNS spoofing mitigation using domain name server security (DNSSEC)

DNS is an unencrypted protocol, making it easy to intercept traffic with spoofing. What's more, DNS servers do not validate the IP addresses to which they are redirecting traffic.

DNSSEC is a protocol designed to secure your DNS by adding additional methods of verification. The protocol creates a unique cryptographic signature stored alongside your other DNS records, e.g., A record and CNAME. This signature is then used by your DNS resolver to authenticate a DNS response, ensuring that the record wasn't tampered with.

While DNSSEC can help protect against DNS spoofing, it has a number of potential downsides, including:

1. Lack of data confidentiality – DNSSEC authenticates, but doesn't encode DNS responses. As a result, perpetrators are still able to listen in on traffic and use the data for more sophisticated attacks.

2. Complex deployment – DNSSEC is often misconfigured, which can cause servers to lose the security benefits or even deny access to a website altogether.

3. Zone enumeration – DNSSEC uses additional resource records to enable signature validation. One such record, NSEC, is able to verify the non-existence of a DNS zone. It can also be used to walk through a DNS zone to gather all existing DNS records—a vulnerability called zone enumeration. Newer versions of NSEC, called NSEC3 and NSEC5, publish hashed records of hostnames, thereby encrypting them and preventing zone enumeration.