**UF2: DOS, DDOS, BACKDOOR**

En la UF1 hemos aprendido que los protocolos son vulnerables debido a su arquitectura y funcionamiento los protocolos ICMP y ARP son utilizados en todas las redes locales del mundo y un buen conocimiento de estos protocolos puede ayudar a entender otros tipos de protocolos de red.

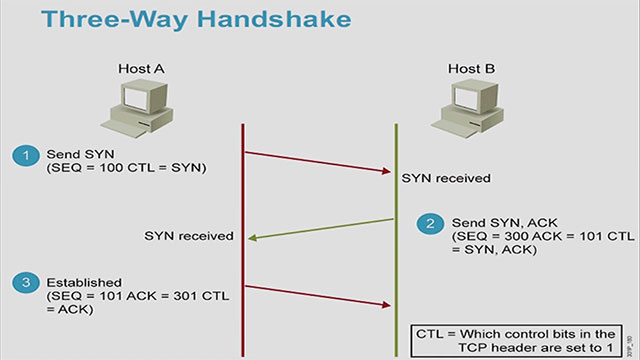
Previamente a utilizar y aprender otros tipos de ataques y en concreto los de denegación de servicios es necesario ver la estructura básica de intercambio de paquetes durante el proceso de conexión y sesión a un servicio de red.

**Three way handshake**

Esta negociación son la relación de pasos que sigue un host para iniciar sesión en un servicio TCP/UDP. El diagrama que tenemos a continuación nos enseña los pasos y los estados de cada paquete en cada momento hasta que se abre sesión.

Los estados del “flag” (bandera) de cada uno de estos paquetes TCP son los siguientes:

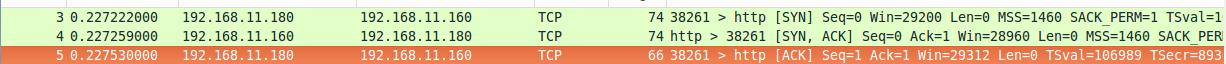
1. SYN: Sincronizacion de conexion
2. ACK: Acknowledge (acuse de recivo)



**Ejercicio three way handshake:**

Para entender el funcionamiento de “three way handshake” es necesario capturar la paqueteria durante el proceso de validación a un servicio de red por lo tanto debemos seguir los siguientes pasos:

1. Buscar un compañero para hacer la practica.
2. Instalar el servicio de “apache”.
3. El compañero se conecta al servicio apache desde el otro host.
4. Mientras el compañero se valida realizamos una captura con el “wireshark”.
5. Realizar una captura de pantalla del “three way handshake” capturado.

****

Con el resultado de esta practica podemos ver claramente el proceso y negociación del “three way handshake” cada vez que un usuario se conecta a un servicio sea del tipo que sea se completa esta operación.

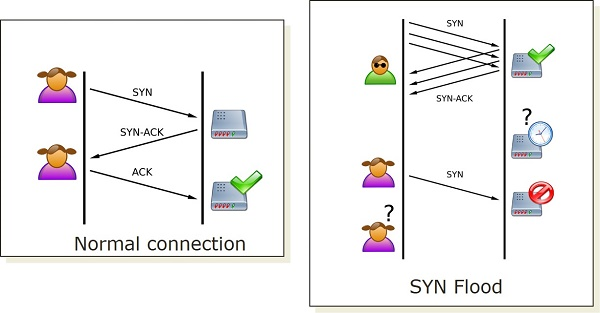
Un servidor recibe cientos o miles de paquetes de tipo SYN por lo tanto este es un proceso habitual y común en las redes.

Los hackers aprobechan esta vulnerabilidad.

¿QUE PASARIA SI UN HACKER INYECTA MILES DE PAQUETES DE TIPO “SYM” A UN SERVICIO DE RED?

A la práctica un ataque solo con inyección de paquetes de tipo SYM no es suficiente para atacar un servicio de red por lo tanto hay que analizar un poco más las estructura de los ataques de tipo DOS.

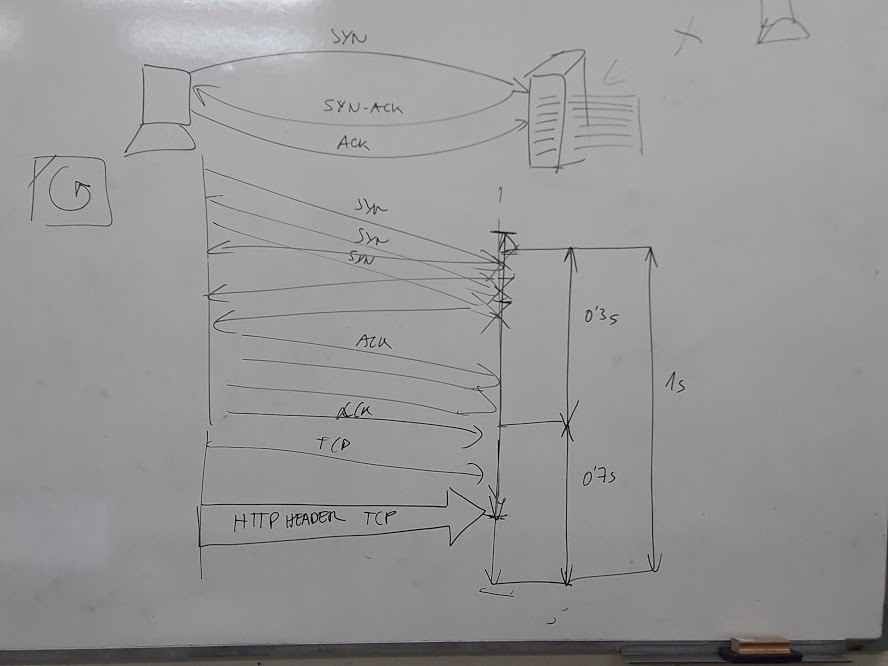
**Ataque de tipo DOS (Denial of Service)**



Los ataques de tipo DOS se basan íntegramente en la estructura de el three way handshake colapsando el servidor de peticiones de tipo SYN aparentemente válidas.

Como hemos visto en el apartado anterior es importante completar el SYN, SYN-ACK, ACK, pero no es suficiente con completar esta operación como hemos visto en la captura de el wireshark justo después de completar el three way handshake el cliente y el servicio intercambian paqueteria propia del protocolo utilizado en nuestro caso era “http”.

Cuando un hacker intenta realizar un DOS solo inyectando paquetes SYN el resultado será que el servicio aceptara estas peticiones y completara el three way handshake pero el cliente al no inyectar más carga del protocolo solicitado el servidor cerrará sesión por lo tanto no conseguiremos colapsar el servicio.

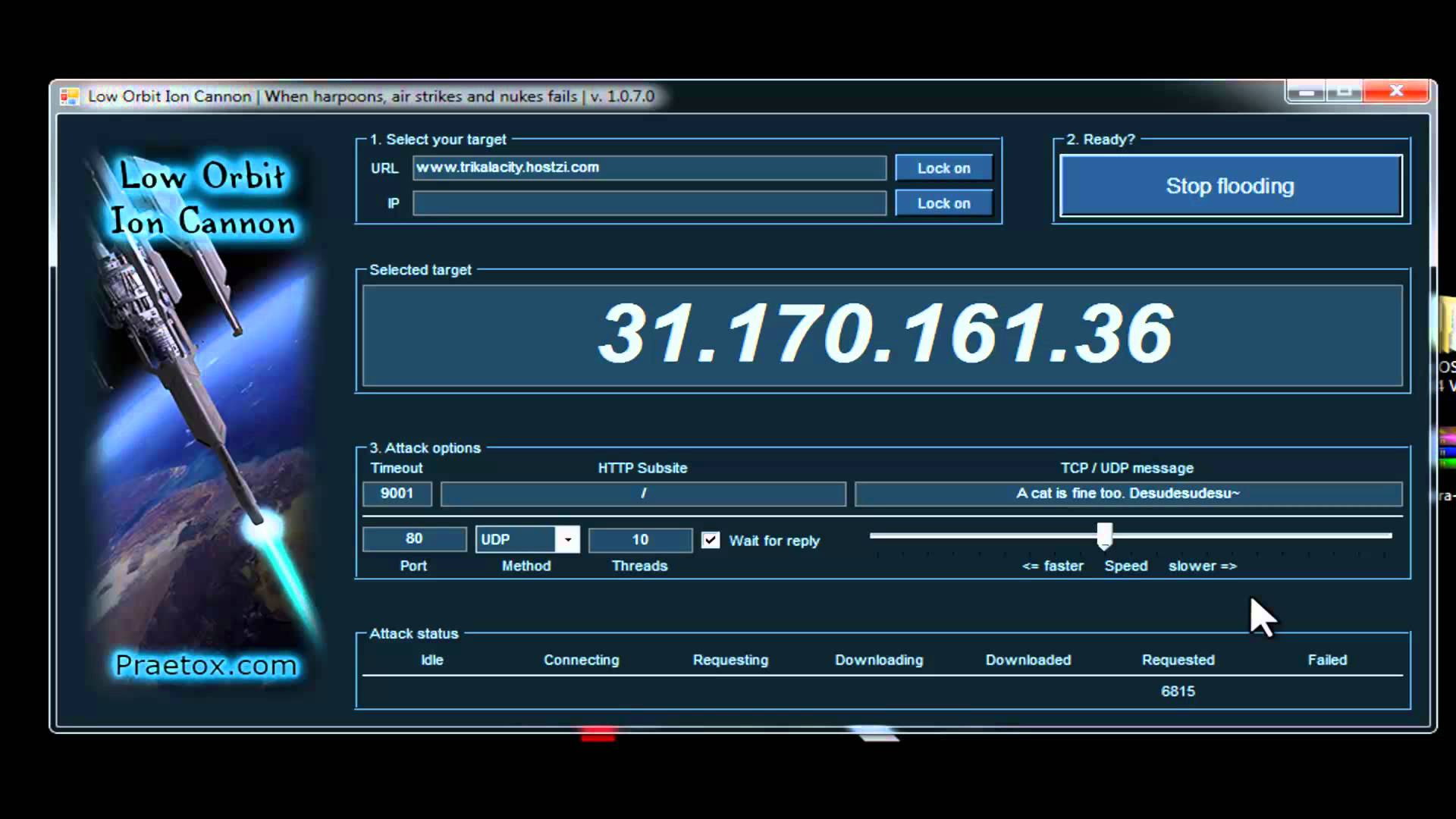


A partir de los conocimientos comentados anteriormente nos damos cuenta que es imprescindible inyectar paqueteria TCP justo después de completar el three way handshake en este diagrama se observa que aumentamos el tiempo de sesion inyectando una cabecera HTTP aparentemente valida esta cabecera tiene el siguiente formato.



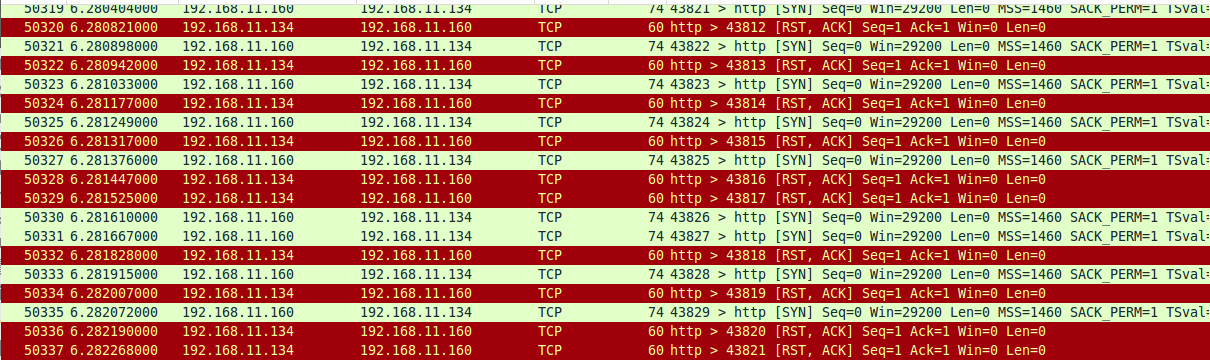
El formato de cabecera HTTP indica a los servicios HTTP que se desea que se el envíe al cliente la página web de índice este tipo de intercambio de cabeceras HTTP es común justo después de completar el three way handshake.

**Software en modo grafico que realizan ataques de tipo DOS (LOIC)**



El LOIC es una simple herramienta que nos permite inyectar paqueteria de tipo SYN y realizar un simple DOS, sus características son:

1. Se puede seleccionar la víctima.
2. Se puede especificar la URL o subdominio de la web a atacar.
3. Se puede especificar el puerto.
4. Se puede especificar el número de procesos asociados al ataque es decir, número de sentencias recurrentes que inyectan paqueteria contra el host víctima.
5. Un dial que indica la latencia de envio de informacion de esa sentencia recurrente.



LOIC es una herramienta visual y fácil de utilizar pero tiene ciertas restricciones de diseño y ademas el codigo opaco (compilado). Una alternativa muy utilizada por los hackers son los scripts programados por lenguajes como por ejemplo PYTHON estos scripts son potentes, manipulables y multiplataforma.

Los scripts de DOS se pueden encontrar en internet y ejecutarlos en un entorno linux. Los scripts mas conocidos son:

1. Hammer Master
2. Thor Hammer

Estos 2 scripts no solo inyectan paqueteria para completar el Three Way Handshake, si no que ademas inyectan HTTP en distintos formatos para simular una conexión web de usuario. La inyección de una cabecera HTTP válida puede indicar a la víctima que un host real se conecta al servicio web, por lo tanto el tiempo de sesion de usuario en el servidor dura mas tiempo que es el objetivo del DOS.

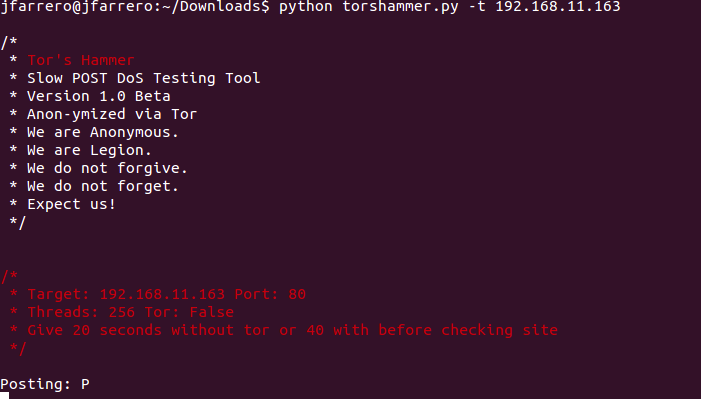
**Hammer Master**

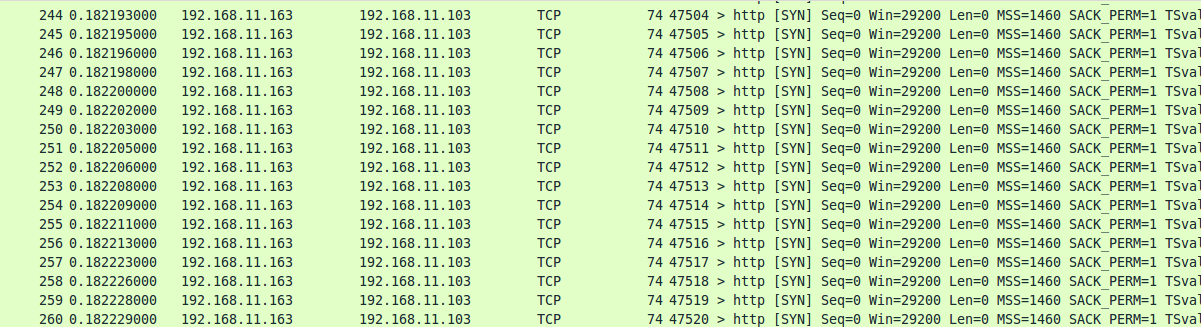
Este script realiza un ataque más elaborado que el LOIC en concreto no solo realiza el Three Way Handshake sino que además simula una peticion HTTP como si fuera un cliente solicitando la web legitima, en concreto este script solo es util para servicios web pero el método siempre es el mismo para todos los protocolos.

**Caracteristicas:**

1. La lista de los Users Agent estan integradas en codigo pero esta lista solo contiene 7 cabeceras.
2. El cuerpo restante de la petición se encuentra alojado en un fichero en el mismo directorio denominado HEADERS.

Este script accepta algunos parámetros a la hora de ejecutarse como por ejemplo: Número de puerto y procesos. La efectividad de Hammer Master esta limitada y para que sea realmente efectivo se deben realizar algunas modificaciones.



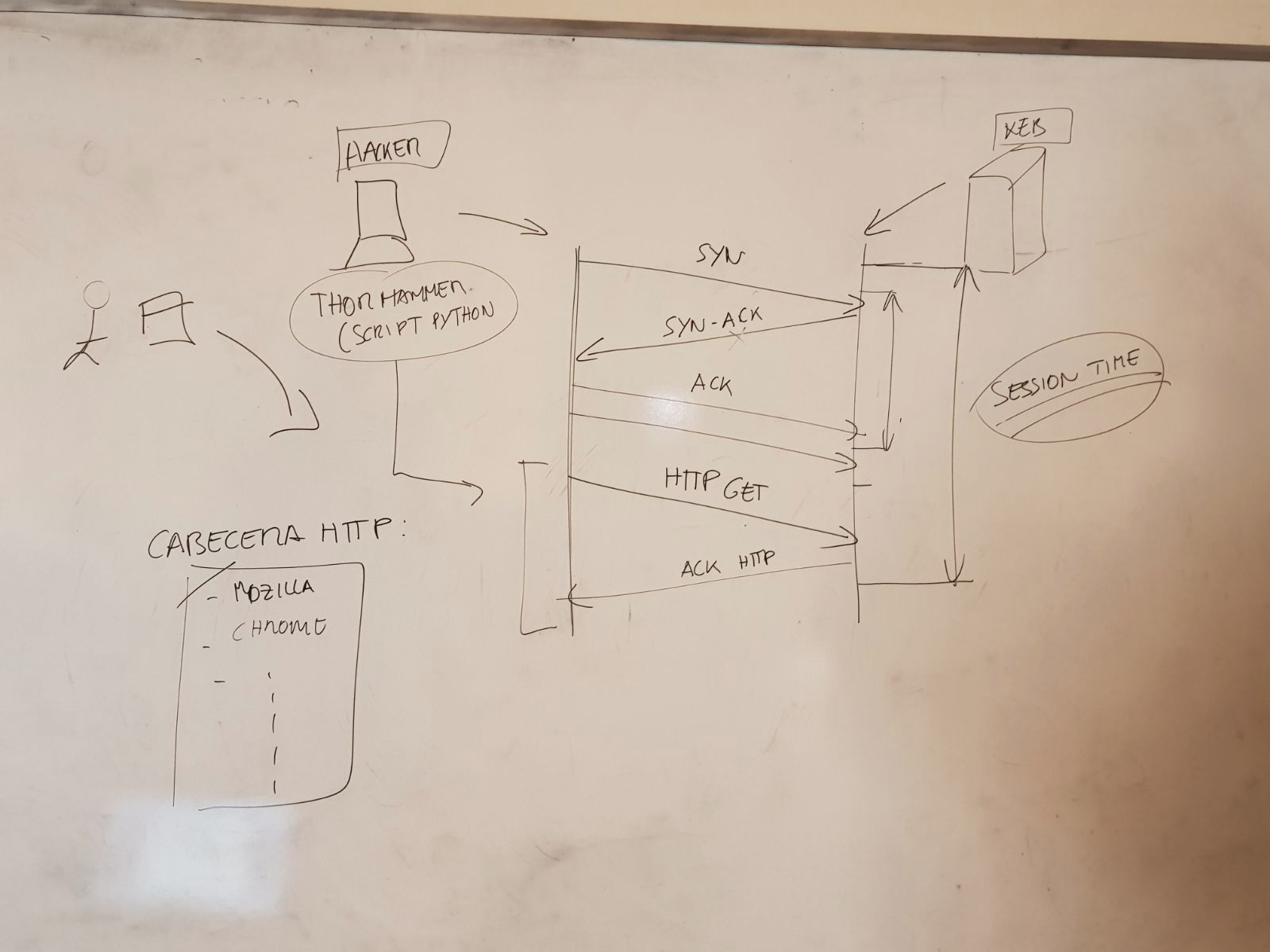


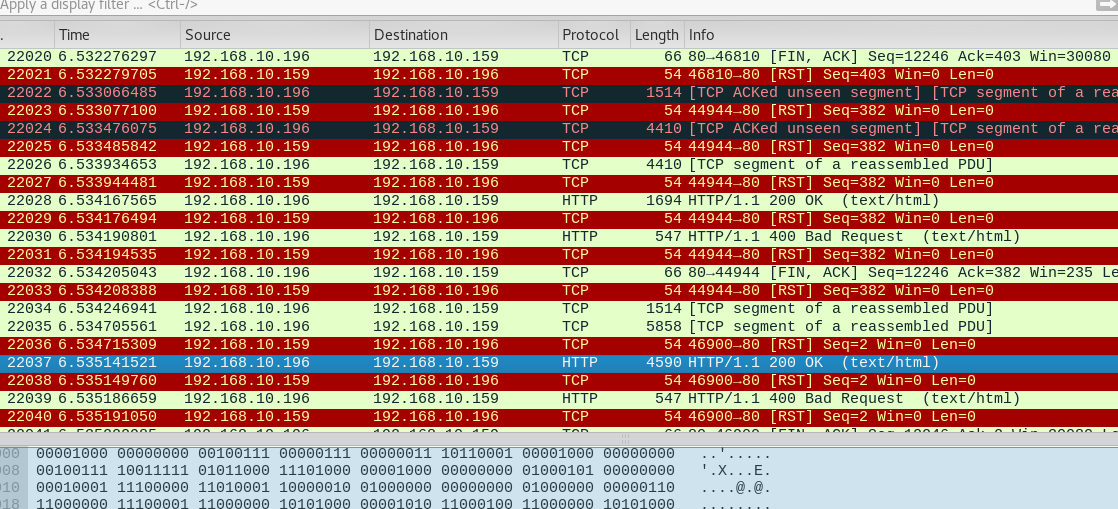
**Thor Hammer**

Este script sigue el mismo patrón que Hammer Master pero por su diseño es mas efectivo. Este script en concreto es una buena base para crear un ataque potente que garantizaria un tiempo de sesión más prolongado para hacer el ataque mas potente.

**Caracteristicas:**

1. Thor Hammer tiene una lista de “user agent” mas extensa.
2. El cuerpo de la cabezera esta integrado en codigo a diferencia del Hummer Master
3. Este tambien soporta multiples procesos simulando varias conexiones simultaneas.

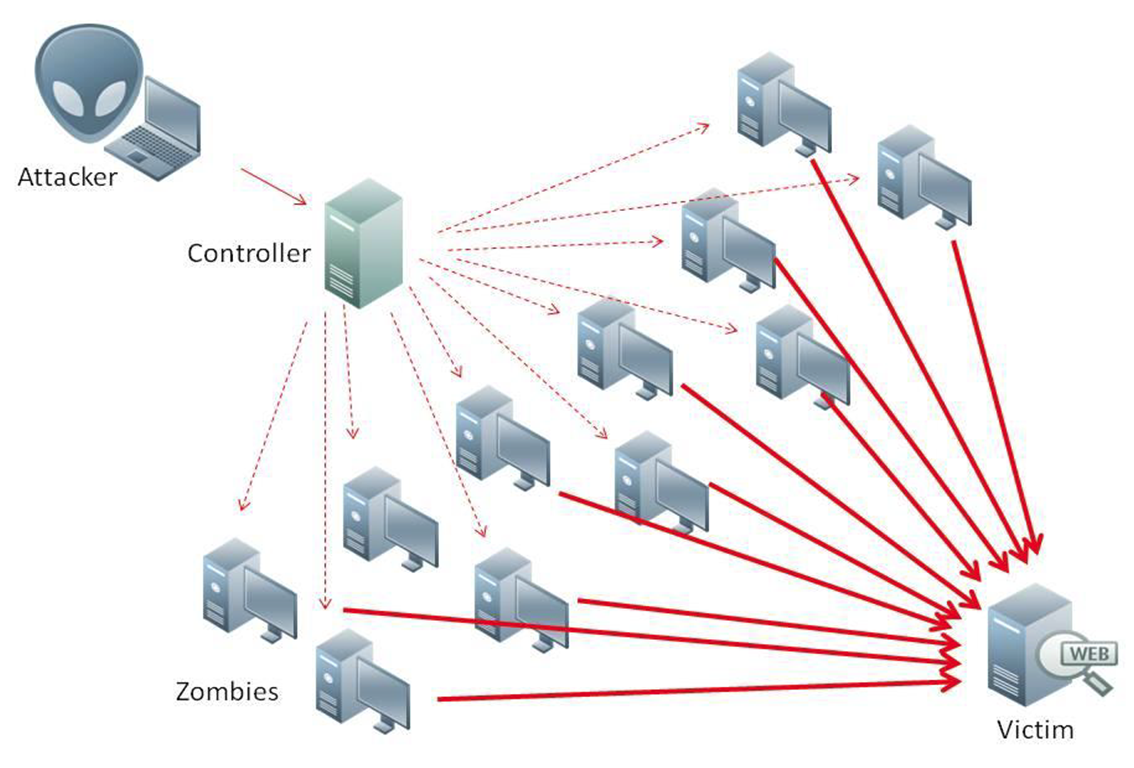




Conclusión: Tanto el LOIC como el HAMMER MASTER y el THOR HAMMER son scripts para ataques DOS. Por el solo hecho de intentar realizar ataques utilizando estas técnicas resulta peligroso y poco efectivo ya que el hacker debe tener una ip alcanzable ya que sino nunca se completaria el Three Way Handshake.

Lo que sí es cierto es que los dos scripts de Phyton son una muy buena base para diseñar los ataques DOS.

**DDOS**

****

La infraestructura de un ataque DDOS es compleja y requiere una planificación importante por parte del hacker los pasos para un ataque DDOS son los siguientes:

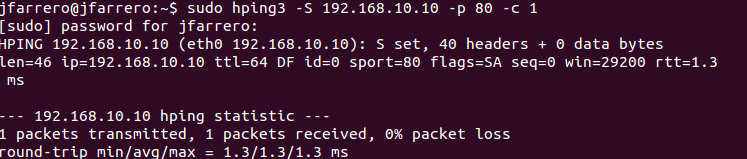
1. Obtener el conocimiento necesario a nivel de programación y sistemas.
2. Crear o aprovechar un codigo de algun juego o aplicación que llame la atención de los usuarios (Zombies).
3. Obtener el maximo de “Zombies” possible.
4. Una vez tenemos una lista de “Zombies” importante es el momento de diseñar la relación de servicios de tipo pasarela (Servidores VPS repartidos por todo el mundo), todos estos servicios son los que tienen contacto directo con los “Zombies”.
5. Crear un algoritmo para las pasarelas que recogen la señal del hacker por un lado y la pasan a los “Zombies” por el otro lado.
6. Crear al algoritmo que implementa el hacker y que indica la ip de la víctima el cuerpo etc. Esta señal la envía el hacker a los servidores VPS y estos la rebotan a los zombies.
7. Los “Zombies” reciben señal e inician cada uno de ellos un simple DOS.

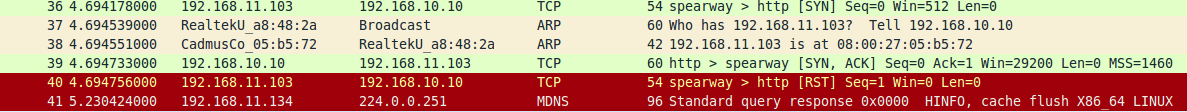
**HPING3: Packet Forging**

En el apartado anterior hemos visto los ataques DOS y comprendemos que el codigo de el ataque es capaz de montar paquetes en función del destino y número de puertos como por ejemplo: un paquete TCP con un flat de tipo SYN. En este apartado vamos a ver una herramienta simple que nos permite crear paquetes de una forma facil, en este caso hablamos del HPING3.

Los 3 modos que soporta HPING3 son los siguientes:

1. Modo parametros y opciones (bash).



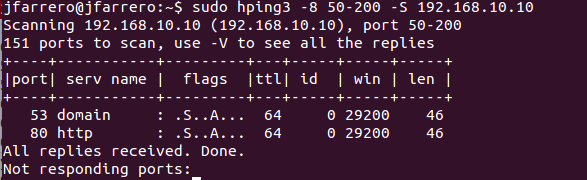


Como podemos observar este modo nos permite forjar un paquete con un Flag de tipo SYN y enviarlo al destino devolviendo como se ve en la imagen de “Wireshark” SYN/ACK y justo después un RST por lo tanto después de enviar el ACK el servicio del puerto 80 nos cierra sesion con RST al no completar el Three Way Handshake.

1. Modo comandos.



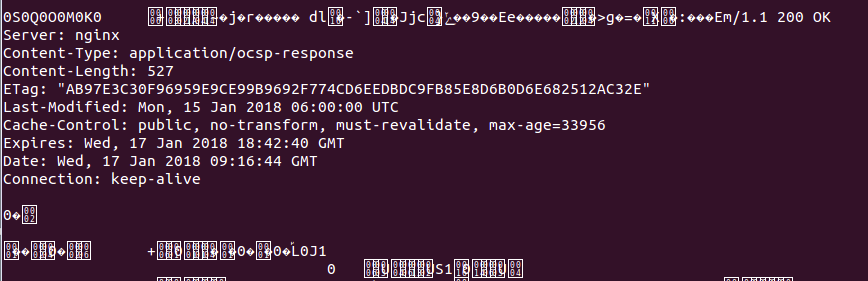
1. Modo rutinas prediseñadas.



**Ejercicio:**

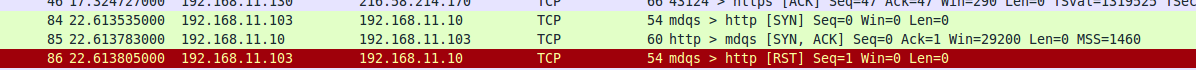
1. Captura con el sniffer de HPING3 todas las cabeceras HTTP en nuestra interfaz de red.





1. Crea en modo comandos un paquete de tipo IP/TCP con flag SYN y envialo verifica con Wireshark que se a forjado correctamente.





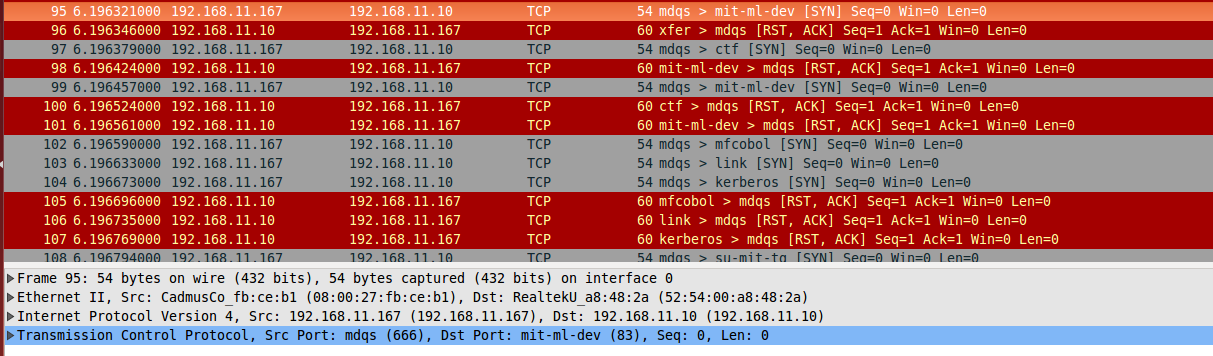
Con este ejercicio hemos aprendido a forjar un paquete y encapsularlo. El paquete IP/TCP permite editar la ip de origen de destino y tiempo de vida del paquete también el puerto origen puerto destino y el estado del flag.

**Ejercicio:**

HPING tambien permite el modo creacion de sentenciàs iterativas en modo comandos.

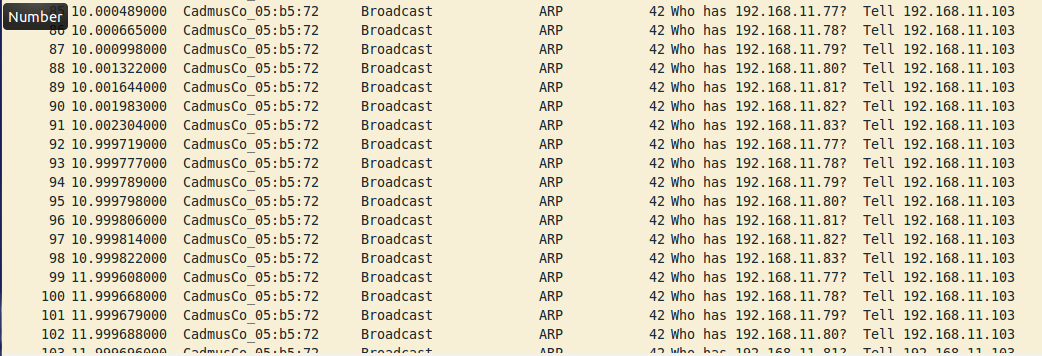
La sentencia FOR nos permitira la encapsulacion repetitiva de un paquete y su siguiente envio.

Una sentencia de ejemplo seria la siguiente:



Como se puede comprobar en las imagenes esta sentencia se a generado paquetes TCP/IP de tipo SYN contra los puertos del 50 hasta el 99 por lo tanto se han forjado 49 paquetes contra 49 puertos distintos como resultado el destino a respondido con paquetes RA o R.

Por la estructura de las sentenciàs FOR estàs se pueden anidar unas dentro de otras:



Como se puede observar en este ejemplo hemos anidado dos sentencias iterativas es decir, por cada iteración del bucle exterior se ejecuta en su totalidad el bucle interior.

En la imagen obtenida con Wireshark vemos múltiples paquetes de tipo ARP que indican que se intenta enviar un paquete a un host desconocido debido a que la sentencia HPING esta forjando paquetes de forma recurrente.

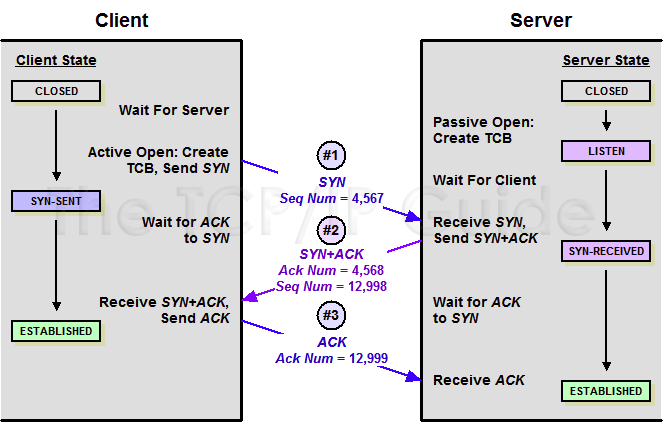
**Ataques de prediccion ISN**

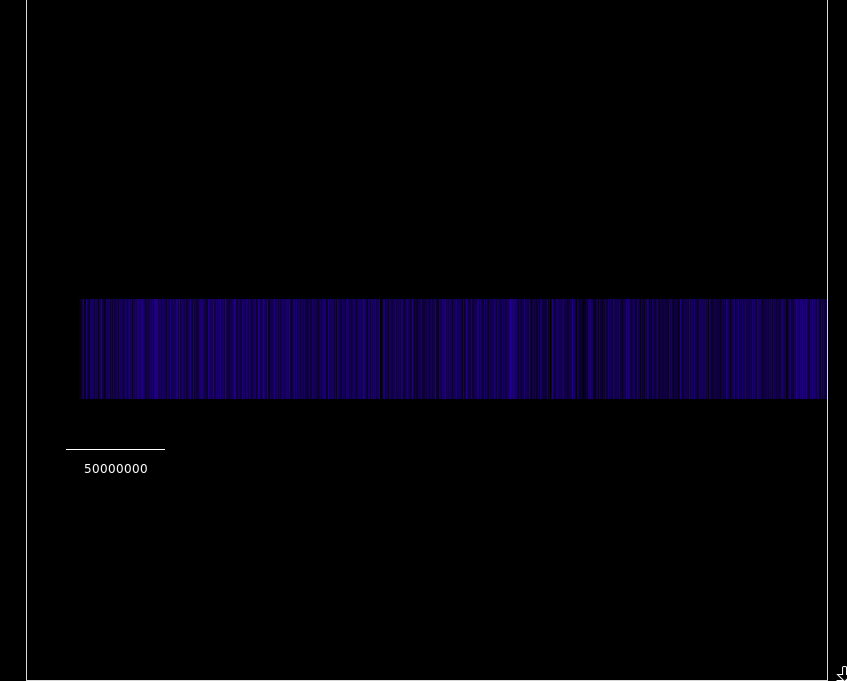
Una de las funciones de HPING es que puede ejecutar algoritmos desarrollados por los mismos programadores estos ficheros se encuentran en el directorio LIB. Uno de ellos nos permite muestrear el rango de los valores de secuencia de paquetes TCP.

Los ISN (Initial Sequence Number) este numero es generado por el cliente y el servicio y se va incrementando para identificar la conexión a lo largo de una comunicación este número va cambiando.

*Número de secuencia (32 bits): Identifica el byte del flujo de datos enviado por el emisor TCP al receptor TCP que representa el primer byte de datos del segmento.*

*Cuando una conexión está siendo establecida el flag SYN se activa y el campo del número de secuencia contiene el ISN (initial sequence number) elegido por el host para esa conexión. El número de secuencia del primer byte de datos será el ISN+1 ya que el flag SYN consume un número de secuencia.*





Como se puede observar en la imagen resultante del ISN spectrogram, el rango de bandas oscila entre el 0 y el valor maximo propuesto como parametro en el ejecutable, en este caso 500.000. Las bandas nos indican a lo largo del tiempo los valores que va tomando el ISN. Si se deja trabajar el script durante un periodo largo de tiempo las bandas irán cambiando de color indicando que ese valor sea repetido cuando las bandas tienden al blanco significa que más agrupacion numerica tienen por lo tanto esa franja numérica permite tener más probabilidad de acierto a la hora de predecir el número.

Conclusión: Realizar un High Jacking utilizando prediccion numerica es una tarea extremadamente complicada y hay otras técnicas más sencillas y efectivas.

**Temario Examen UF2**

1. **DOS: LOIC, Thor-Hammer, Hammer-Master.**
2. **Three Way Handshake**
3. **DDOS**
4. **HPING3**
5. **ISN**