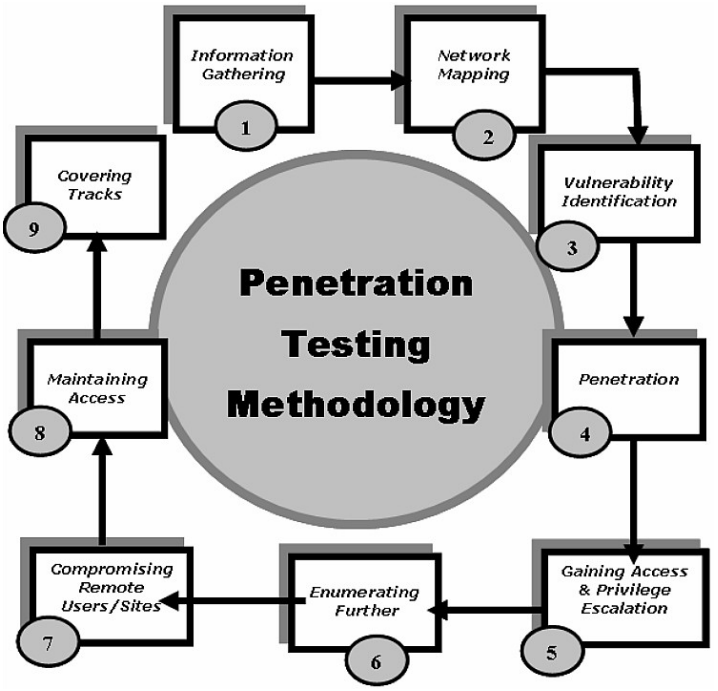
# Network Mapping




Como se ha visto, la misión de un firewall es lograr el cumplimiento de una política de control de acceso entre redes; es decir, permitir el paso de paquetes asociados a comunicaciones autorizadas y bloquear el resto. Uno de los primeros pasos en un proceso de auditoría o de un ataque es el Network Mapping en el que se trata de determinar:

- Los equipos activos.
- Los puertos abiertos y las aplicaciones ejecutándose en ellos.
- El sistema operativo y versiones de aplicaciones ejecutándose.
- La topología de la red (perímetro, redes internas, ...).

Las reglas definidas en los firewalls han de permitir las comunicaciones deseadas sin desvelar más información que la necesaria; por ejemplo, no permitiendo el acceso desde Internet a servicios que únicamente deben ser accesibles desde la Intranet. Los administradores de sistemas, analistas y atacantes disponen de diversas técnicas y herramientas para abordar el mapeado de la red y la verificación de las reglas de un firewall. Conocer estas técnicas y los protocolos empleados es importante para decidir si se estos protocolos se permiten/bloquean en los firewalls.

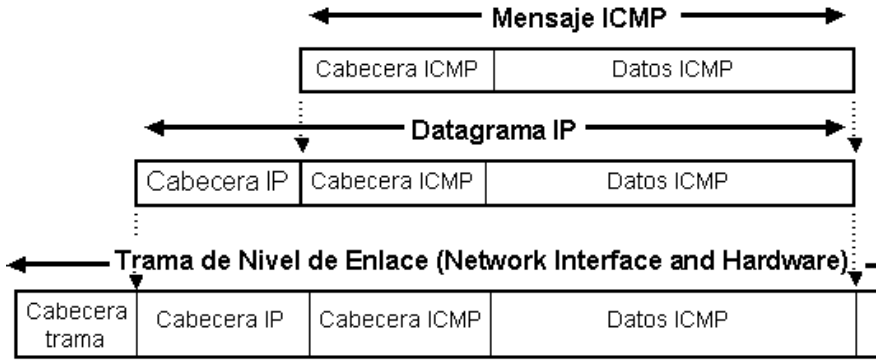
## ICMP (Internet Control Message Protocol)

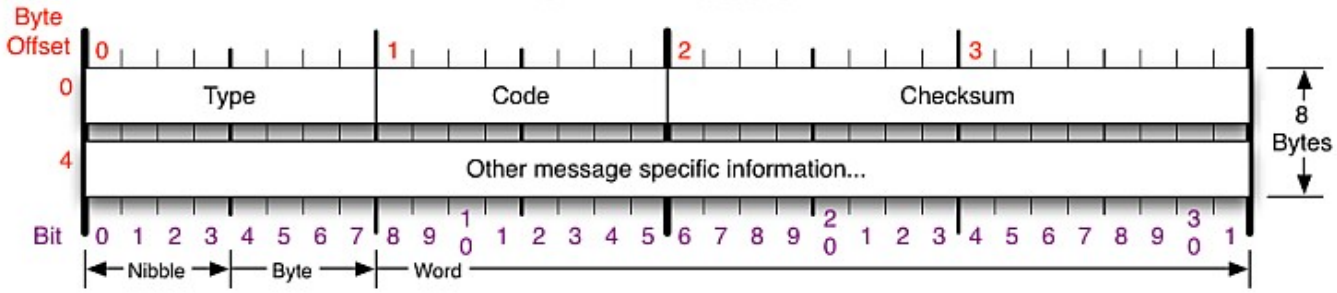
En el funcionamiento normal de una red se dan a veces situaciones extraordinarias que requieren enviar avisos especiales; por ejemplo, si un datagrama con el bit DF (Don’t Fragment) puesto a 1 no puede pasar por una determinada red, el router donde se produce el problema debe devolver un mensaje al host emisor indicándole lo sucedido. El mecanismo para indicar todos estos incidentes en Internet es el protocolo conocido como ICMP, que proporciona un descarte informado[1].

Entre las características más importantes de ICMP están:

- Los mensajes ICMP viajan por la red viajando sobre datagramas IP.
- Los mensajes ICMP están sujetos en los routers a las mismas reglas que cualquier otro datagrama (enrutamiento, fragmentación, tiempo de vida, etc.).
- Los mensajes ICMP son generados por el host o router que detecta el problema o situación extraordinaria, y van dirigidos al host o router que aparece en el campo dirección origen del datagrama que causó el problema.
- Los mensajes ICMP poseen una cabecera y un campo de datos:
  - Cabecera ICMP:
    - Tipo: tipo de mensaje ICMP (eco, respuesta de eco, destino inalcanzable, etc.).
    - Código: especifica el mensaje ICMP dentro de un tipo (destino inalcanzable por red destino desconocida, etc.).
    - Checksum: suma de comprobación de errores.
    - Datos específicos del tipo: datos opcionales para cada tipo de ICMP.
  - Datos ICMP: En el caso de mensajes de error y para facilitar la identificación del datagrama por parte del host emisor, se incluye la cabecera y los primeros ocho bytes de datos del datagrama original.

1 Esto no convierte a IP en fiable ya que no hay secuenciamiento ni retransmisión de datagramas IP.




**ICMP Message Types**

| Type | Code/Name |
|---|---|
| 0 | Echo Reply |
| 3 | Destination Unreachable |
| | 0 Net Unreachable |
| | 1 Host Unreachable |
| | 2 Protocol Unreachable |
| | 3 Port Unreachable |
| | 4 Fragmentation required, and DF set |
| | 5 Source Route Failed |
| | 6 Destination Network Unknown |
| | 7 Destination Host Unknown |
| | 8 Source Host Isolated |
| | 9 Network Administratively Prohibited |
| | 10 Host Administratively Prohibited |
| | 11 Network Unreachable for TOS |
| 3 | Destination Unreachable (continued) |
| | 12 Host Unreachable for TOS |
| | 13 Communication Administratively Prohibited |
| 4 | Source Quench |
| 5 | Redirect |
| | 0 Redirect Datagram for the Network |
| | 1 Redirect Datagram for the Host |
| | 2 Redirect Datagram for the TOS & Network |
| | 3 Redirect Datagram for the TOS & Host |
| 8 | Echo |
| 9 | Router Advertisement |
| 10 | Router Selection |
| 11 | Time Exceded |
| | 0 TTL Exceeded |
| | 1 Fragment Reassembly Time Exceeded |
| 12 | Parameter Problem |
| | 0 Pointer Problem |
| | 1 Missing a Required Operand |
| | 2 Bad Length |
| 13 | Timestamp |
| 14 | Timestamp Reply |
| 15 | Information Request |
| 16 | Information Reply |
| 17 | Address Mask Request |
| 18 | Address Mask Reply |
| 30 | Traceroute |

**Checksum**

Checksum of ICMP header

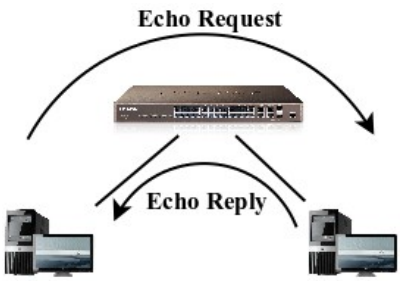
**RFC 792**

Please refer to RFC 792 for the Internet Control Message protocol (ICMP) specification.

Algunos de los mensajes ICMP se usan en el network mapping:

### Echo Request y Echo Reply

Usados por el comando ping para detectar si un destino determinado está operativo. Automatizar el proceso de hacer ping a varios equipos se conoce como barrido ping o ping sweep.




```
manuel@pcmanuel:~$ ping 192.168.0.100
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.
64 bytes from 192.168.0.100: icmp_req=1 ttl=255 time=0.627 ms
64 bytes from 192.168.0.100: icmp_req=2 ttl=255 time=0.615 ms
^C
--- 192.168.0.100 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.615/0.621/0.627/0.006 ms
```

Si al enviar un ICMP Echo Request se recibe como respuesta un Echo Reply se sabe que el equipo destino está operativo; sin embargo, el no recibir el Echo Reply no implica necesariamente que el equipo destino no esté operativo. Puede que no exista, o que esté apagado en ese momento, o que algún equipo intermedio prohíba el paso de mensajes ICMP o que el equipo esté configurado para no responder a este tipo de mensajes. Habrá que utilizar otras técnicas para descubrirlo.

### Destino Inalcanzable (Destination Unreachable)

Este mensaje se produce cuando no se puede entregar el datagrama en su destino por diversas situaciones. A modo de ejemplo:

- Cuando un router no encuentra en sus tablas ninguna ruta por la que pueda llegar a la dirección para la que va dirigido un datagrama.
- Cuando un router se encuentra con un datagrama que tiene puesto a 1 el bit DF (Don't Fragment) y que no cabe en la MTU de la red por la que ha de enviarlo.
- Cuando un router no puede reenviar un datagrama debido a que existe un filtro de paquetes en la ruta.

**Tiempo excedido (Time Exceeded)**

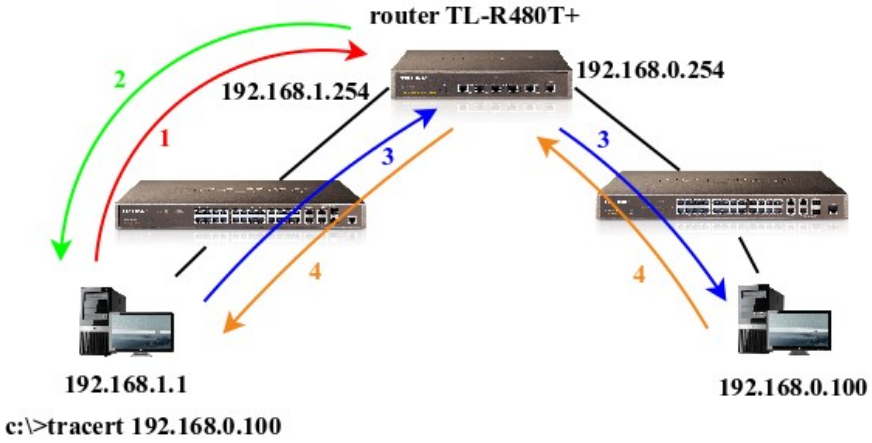
Se envía al emisor de un mensaje cuando:

```
▷ Ethernet II, Src: 08:00:09:72:74:e0 (08:00:09:72:74:e0), Dst: 00:00:b4:34:b7:fa (00:00:b4:34:b7:fa)
▷ Internet Protocol, Src: 168.156.1.33 (168.156.1.33), Dst: 134.39.89.236 (134.39.89.236)
▽ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 1 (Host unreachable)
    Checksum: 0xa7a2 [correct]
  ▷ Internet Protocol, Src: 134.39.89.236 (134.39.89.236), Dst: 10.0.0.1 (10.0.0.1)
```

- Cuando un router al decrementar el TTL de la cabecera IP llega a 0[2]. Esto puede ser síntoma de que se ha producido algún bucle en la red o que el valor del TTL utilizado es demasiado bajo para el diámetro de la red.
- Cuando el temporizador de reensamblado de un datagrama IP fragmentado llega expira.

El comando tracert/traceroute juega con los mensajes ICMP Echo Request, Echo Reply, Tiempo Excedido y Destino Inalcanzables para saber la ruta que sigue un paquete hasta llegar a su destino (traza de ruta). Tracert envía mensajes ICMP Echo Request al destino partiendo de un ttl=1 e incrementando su valor en sucesivos mensajes. Estos mensajes caducan al llegar al primer router, al segundo, …, de forma que los routers intermedios enviarán mensajes ICMP de Tiempo Excedido descubriendo de esta forma su existencia.

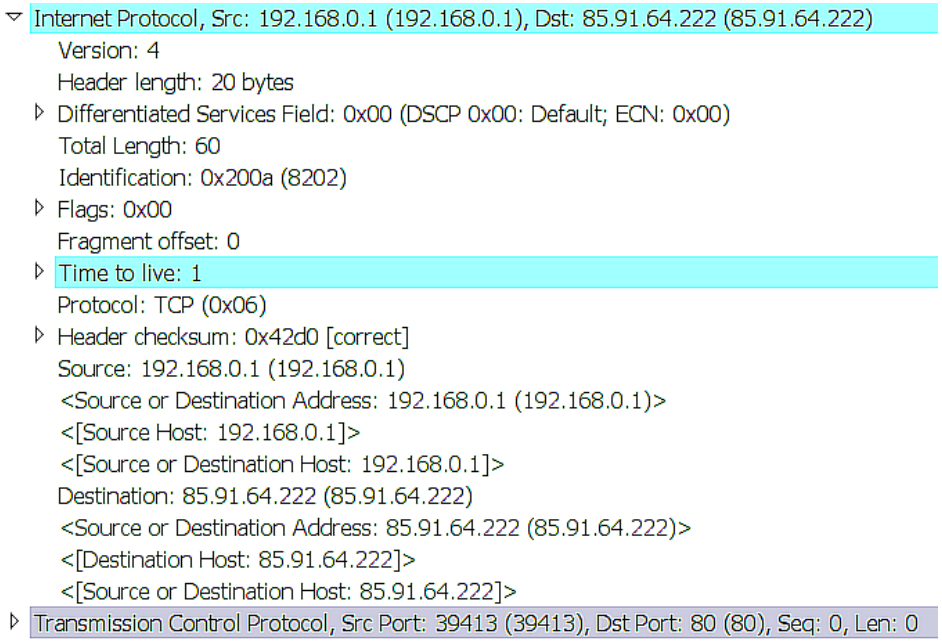
```
C:\>tracert 192.168.0.100

Traza a la dirección router.iesxulianmagarinos.edu.xunta.es [192.168.0.100]
sobre un máximo de 30 saltos:

  1    10 ms    <10 ms    <10 ms  brazilfw.iesxulianmagarinos.edu.xunta.es [192.168.1.254]
  2   <10 ms    <10 ms    <10 ms  router.iesxulianmagarinos.edu.xunta.es [192.168.0.100]
```




1. ICMP Echo Request con destino 192.168.0.100 y ttl=1 .
2. ICMP Time Exceeded → enviado por la interfaz cercana del router (192.168.1.254) .
3. ICMP Echo Request con destino 192.168.0.100 y ttl=2 (atraviesa el router).
4. ICMP Echo Reply procedente de 192.168.0.100.

Algunas implementaciones de traceroute permiten enviar mensajes TCP o UDP en vez de ICMP Echo Request (normalmente bloqueados en los firewalls). El funcionamiento es similar, ya que se envían los mensajes TCP o UDP[3] en datagramas IP con ttl=1, 2, 3, …

2 Los routers al recibir un paquete comprueban el checksum y el resultado de hacer TTL-1 antes de proceder a reenviar un paquete.
3 El puerto destino lo puede escoger el usuario para tratar de hacer el paquete de prueba lo más normal posible




## ARP (Address Resolution Protocol)

En el modelo del capas, el nivel de red usa al nivel de enlace para el transporte de sus paquetes de datos. El nivel de red usa direcciones IP y en cambio el nivel de enlace usa direcciones MAC para identificar los equipos. A partir de una dirección MAC no se puede deducir la dirección IP del equipo y viceversa. Los protocolos de resolución de direcciones ARP/RARP son responsables de convertir las direcciones IP a direcciones de red físicas y viceversa utilizando un sistema de pregunta-respuesta. Los mensajes ARP son:

- **ARP Request (solicitud ARP)**
  - Solicita la MAC asociada a una dirección IP.
  - Esta pregunta viaja en una trama broadcast de nivel 2 para llegar a todos los equipos del segmento de red al que está conectada la interfaz que envía el mensaje.
- **ARP Reply (respuesta ARP)**
  - El nodo cuya dirección IP coincida con la dirección IP solicitada en el ARP Request envía una respuesta ARP.
  - Este mensaje viaja en una trama unicast dirigida a la MAC del solicitante ARP.

Al ser ARP un protocolo de nivel de red, pero trabajando por debajo de IP, hace que la gran mayoría de los firewalls no bloqueen estos mensajes. De este hecho se aprovecha la técnica arp sweep o barrido arp, que permite hacer 'arping' a equipos dentro de una red local descubriendo aquellos que están operativos, aun cuando tengan el firewall activado (por ejemplo bloqueando los mensajes icmp).

```
[manuel@pcmanuel ~]$ ping -c 2 192.168.0.80
PING 192.168.0.80 (192.168.0.80) 56(84) bytes of data.
^C
--- 192.168.0.80 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 10C

[manuel@pcmanuel ~]$ sudo arping -c 2 -I enp2s0 192.168.0.80
ARPING 192.168.0.80 from 192.168.0.10 enp2s0
Unicast reply from 192.168.0.80 [08:00:27:82:E0:9D]  0.741ms
Unicast reply from 192.168.0.80 [08:00:27:82:E0:9D]  0.750ms
Sent 2 probes (1 broadcast(s))
Received 2 response(s)
```

# Port Scanning (Exploración de puertos)

Un port scanning es el proceso de conexión a puertos TCP y UDP del sistema objetivo para determinar qué servicios se están ejecutando. El identificar los puertos que están a la escucha permitirá:

- Descubrir equipos funcionando .
- Determinar qué aplicaciones se están ejecutando.
- Determinar el tipo de sistema operativo.

Al ser TCP/UDP los protocolos encargados de encapsular los datos de nivel de aplicación (datos de usuario) podremos superar las dos grandes limitaciones de las técnicas anteriores: el alcance limitado del arp sweep y el más que habitual bloqueo de los mensajes icmp en los routers-firewall.

Los servicios activos pueden llegar a permitir la entrada de un usuario no autorizado si están mal configurados o se trabaja con versiones de software con problemas de seguridad conocidos.

Aunque existen multitud de herramientas para hacer análisis de puertos, la más conocida por su potencia y funcionalidades es nmap. Nmap además de explorar puertos es capaz de detectar que sistema operativo corre en la máquina objetivo (fingerprinting). Para ello analiza los paquetes procedentes de la máquina objetivo y los compara con implementaciones conocidas de los fabricantes.

```
root@kaliLinux:~# nmap -sV -O -p 1-65535 192.168.56.92

Starting Nmap 6.25 ( http://nmap.org ) at 2013-06-25 23:28 CEST
Nmap scan report for 192.168.56.92
Host is up (0.0026s latency).
Not shown: 65529 closed ports
PORT     STATE SERVICE      VERSION
80/tcp   open  http         Apache httpd 1.3.17 ((Win32))
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 2003 or 2008 microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
1026/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 08:00:27:1B:18:51 (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows 2000|XP|2003
OS CPE: cpe:/o:microsoft:windows_2000::sp2 cpe:/o:microsoft:windows_2000::sp3 cpe:/o:microsoft:w
indows_2000::sp4 cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3 cpe:/o:micros
oft:windows_server_2003::- cpe:/o:microsoft:windows_server_2003::sp1 cpe:/o:microsoft:windows_se
rver_2003::sp2
OS details: Microsoft Windows 2000 SP2 - SP4, Windows XP SP2 - SP3, or Windows Server 2003 SP0 -
 SP2
Network Distance: 1 hop
```

# Seguridad por obscuridad: Port-Knocking y SPA

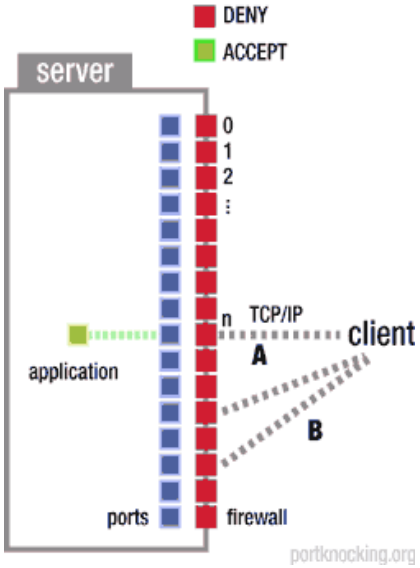
El principio de seguridad por obscuridad en el campo en que estamos trabajando se puede ver como un intento de hacer un sistema/servicio más seguro mediante su ocultación. Si un atacante no sabe de la existencia del sistema/servicio no podrá atacarlo. Dejando debates a un lado, lo que tenemos que tener claro es que ocultar algo, hace más díficil el trabajo del atacante, pero eso no quiere decir que nuestro sistema sea más seguro en el caso de ser descubierto. Colocar un servicio ssh para administrar un servidor de la DMZ desde Internet en un puerto raro, diferente al puerto por defecto, no impide que haciendo una exploración de puertos exhaustiva sea descubierto. Una vez que la ocultación no funciona, la seguridad del servicio dependerá de la versión y configuración del programa.

La idea de la seguridad por obscuridad sigue ahí y fue perfeccionada más allá de mover el servicio a puertos raros. Los sistemas que vamos a ver tratan de **ocultar un servicio** (tipicamente ssh) y **permitir el acceso al mismo bajo demanda**, enviando una secuencia especial de paquetes o un paquete especialmente formado. Un servicio oculto

de esta manera es desconocido para un atacante; ya que, el puerto del servicio está cerrado (más bien, está abierto pero bloqueado por el firewall) y por lo tanto no es susceptible de ser atacado.

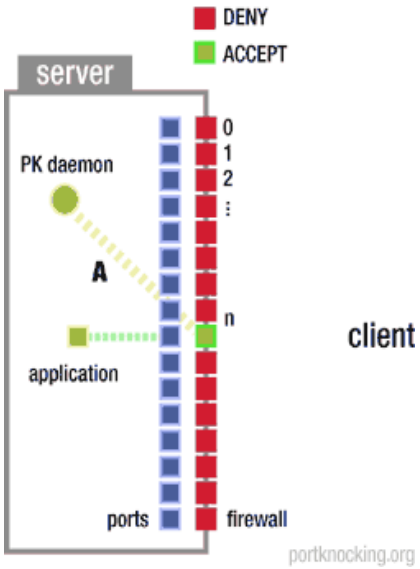
## Port-knocking

En port-knocking el punto de partida es un servicio activo (p.e. un servidor ssh escuchando en el puerto 22/tcp) pero bloqueado por el firewall. De esta forma, para cualquier intento de conexión o exploración de puertos el puerto aparece como cerrado. Sin embargo, un usuario que envíe una secuencia preestablecida de paquetes a unos puerrtos determinados hará que el firewall se reconfigure permitiendo el acceso al servicio. Es equivalente a golpear en una puerta con una secuencia especial de golpes pactada previamente, para que los de dentro al reconocerla abran la puerta. En la siguiente secuencia de imégenes aparece explicado el proceso:




1. El cliente no puede conectarse a la aplicación que escucha en el puerto n ni a otros puertos ya que el firewall los bloquea.


2. El cliente se conecta a una serie de puertos en una secuencia definida. Sigue sin poder establecer ninguna conexión al bloquearlas el firewall.

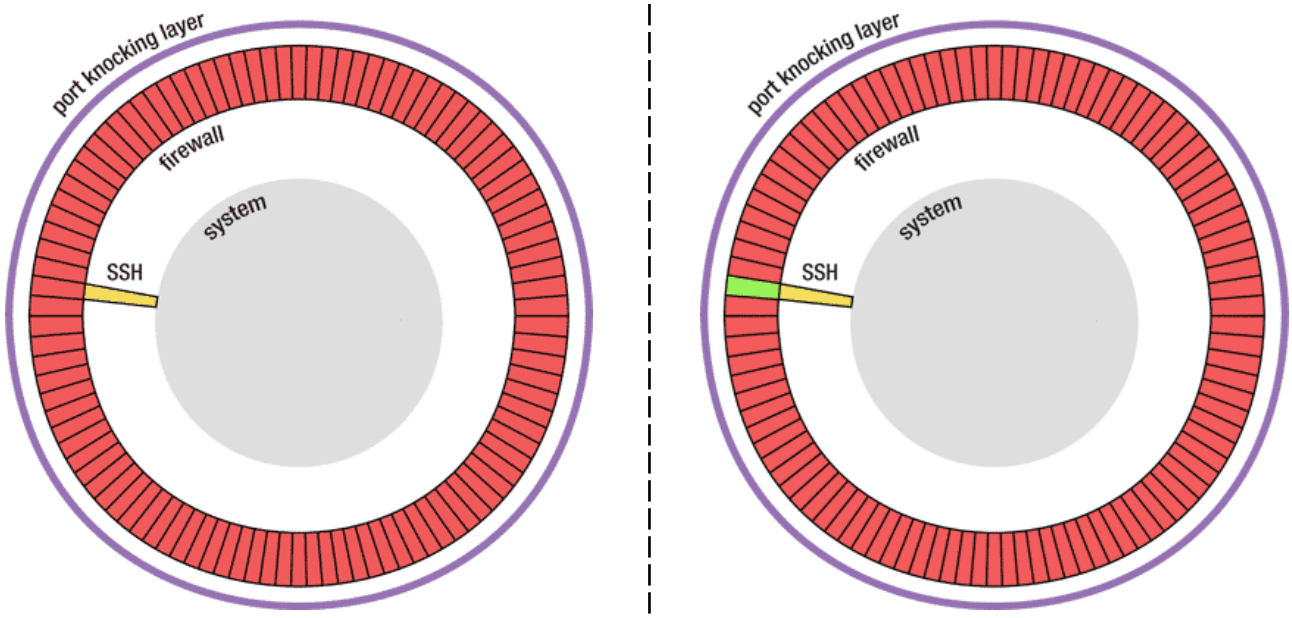



3. El servidor de port-knocking intercepta los intentos de conexión y los reconoce modificando las reglas del firewall para abrir el puerto n al cliente.


4. El cliente se conecta al puerto n accediendo al servicio deseado, autenticándose con los mecanismos normales asociados a ese servicio.

Fig. Port-knocking. Fuente: http://www.portknocking.org/

Port-knocking proporciona una capa más de seguridad:




Port-knocking es una capa más de seguridad, a mayores de la proporcionada por el firewall y las existentes detrás del firewall (autenticación de usuarios en ssh).

En el caso de darse la secuencia de puertos correcta, port-knocking permite acceder al servicio; y por tanto, al sistema de autenticación de ssh.

Fig. Port-knocking como una capa más de seguridad. Fuente: http://www.portknocking.org/

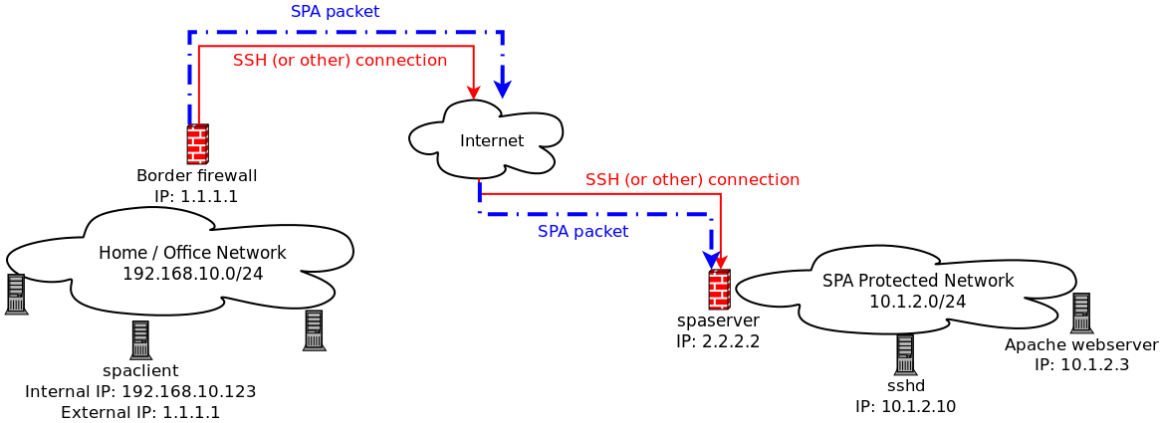
Port-knocking es interesante, pero puede presentar algunos problemas:

- Descubrimiento de la secuencia de paquetes, permitiendo que otro usuario abra el puerto do servicio.
- DoS (denegación de servicio), bien a propósito enviando un atacante paquetes al azar falsificando la IP del usuario legítimo para romper la secuencia de paquetes ou bien por azar; ya que, en una red TCP/IP los paquetes no tienen por que seguir la misma ruta, pudiendo llegar desordenados o no llegar algunos de ellos.
- NAT: varios equipos situados detrás de una misma IP pública pueden interferir entre ellos al acceder al mismo sistema al compartir la misma IP pública.

## SPA (Single Packet Authorization)

Con este sistema, el cliente envía un único paquete cifrado al equipo donde un servidor SPA estará a la escucha y se encargará de capturar y analizar el contenido del paquete. Si todo es correcto, el servidor SPA reconfigurará el firewall para permitir que el cliente inicie una conexión hacia el servicio deseado. Este sistema se ideó para superar las limitaciones del Port-knocking; ya que, se envía un único paquete cifrado y dentro del mismo va información sobre el equipo que se quiere conectar, el puerto que se desea abrir y una marca de tiempo.

En el siguiente esquema de http://www.cipherdyne.org/fwknop/ podemos ver un resumen del funcionamiento:



Para que *spaclient* pueda conectarse por ssh al servidor sshd, debe enviar un paquete SPA adecuado al firewall *spaserver*. *Spaserver* ejecuta una política de tráfico DROP y descarta los paquetes; sin embargo, el servicio SPA está 'sniffando' los paquetes y recoge el paquete SPA enviado por el cliente. Tras analizar el paquete, el servicio SPA procede a reconfigurar el *ruleset* del firewall para permitir que *spaclient* inicie una conexión ssh hacia el equipo sshd.

Fijarse en los siguientes puntos:

- El servicio SPA no está escuchando en ningún puerto; si no que captura el paquete del cliente, gracias a la librería libpacp (como haría wireshark o tcpdump). No podemos abrir conexiones con el servicio SPA, por que no hay puerto para conectarse con él. Lo que está haciendo es capturar paquetes como si fuese un sniffer.
- El servicio SPA reconfigura en tiempo real el firewall añadiendo las regras necesarias (filtrado y NAT) para que *spaclient* pueda iniciar una conexión ssh con el equipo sshd.
- Tras un tiempo determinado (normalmente unos pocos segundos) vuelve a reconfigurar el firewall cerrando el puerto para todo el mundo (incluido *spaclient*):
    - Si durante ese intervalo de tiempo *spaclient* no estableció una conexión con sshd, tendrá que comenzar de nuevo el proceso.
    - Si durante ese intervalo de tempo *spaclient* si estableció una conexión con sshd y después el servidor SPA reconfigura el firewall para impedir nuevas conexiones, la conexión *spaclient*-sshd establecida no se pierde al estar permitida por las regras ESTABLISHED, RELATED del ruleset.