REDES

NAT, VPN, DMZ

REDES

NAT

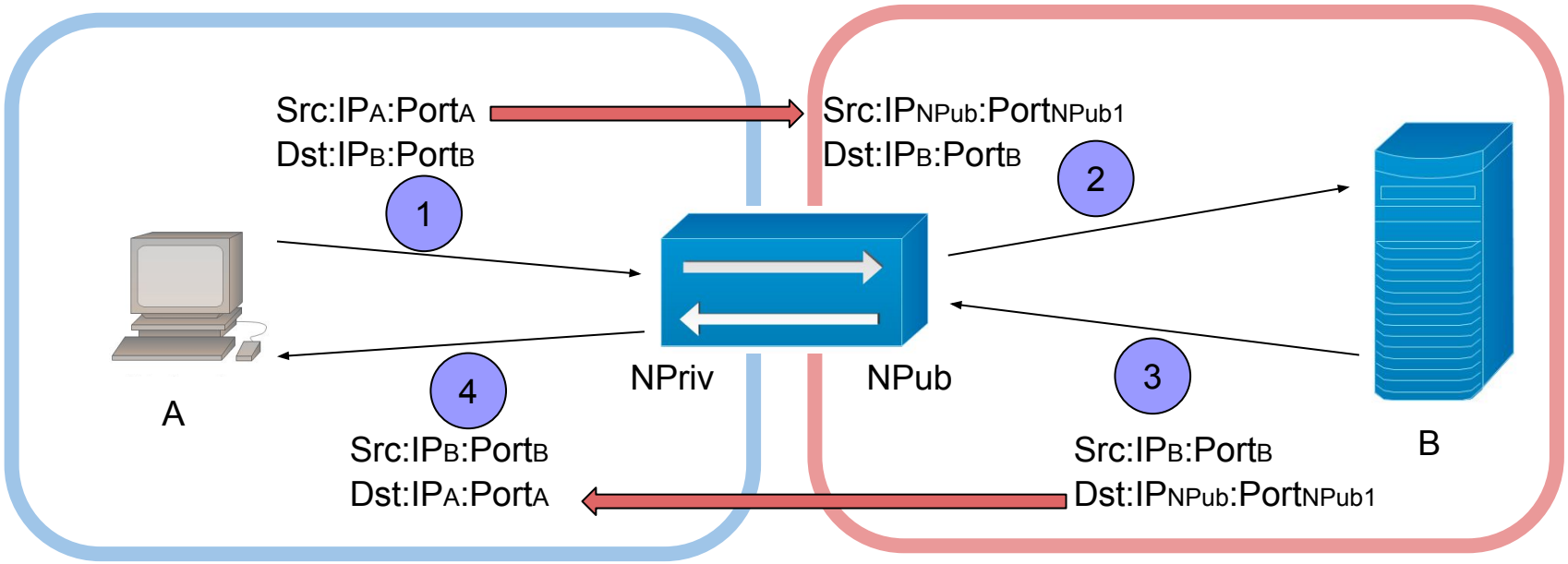
NAT

- No está estandarizado en los detalles (RFC 1631 es muy general)
- El NAT tradicional no acepta conexiones entrantes a menos que hayan sido salientes.
- Fue diseñado para el concepto cliente/servidor
- NAT no desaparece con IPv6 porque siguen existiendo los firewalls y las redes públicas y privadas

SNAT (Source NAT)

En algunos casos, el NAT unicamente cambia la source IP y mantiene el puerto, pero esto asume que se cuenta con tantas IP publicas como dispositivos que se quieren conectar a internet. Este no es el caso de las redes de hogar (en donde hay una sola IP publica, y se va cambiando el puerto).

- Utilizado cuando un dispositivo en una red privada contacta a otro en una red pública
- Típicamente el de la red pública es un servidor de algún tipo



En TCP hace connection tracking (sigue el estado de la conexion). En UDP, usa timeouts.
Mientras exista esta conexion (o no haya ocurrido timeout), usa el mismo puerto e IP para el mismo source host.

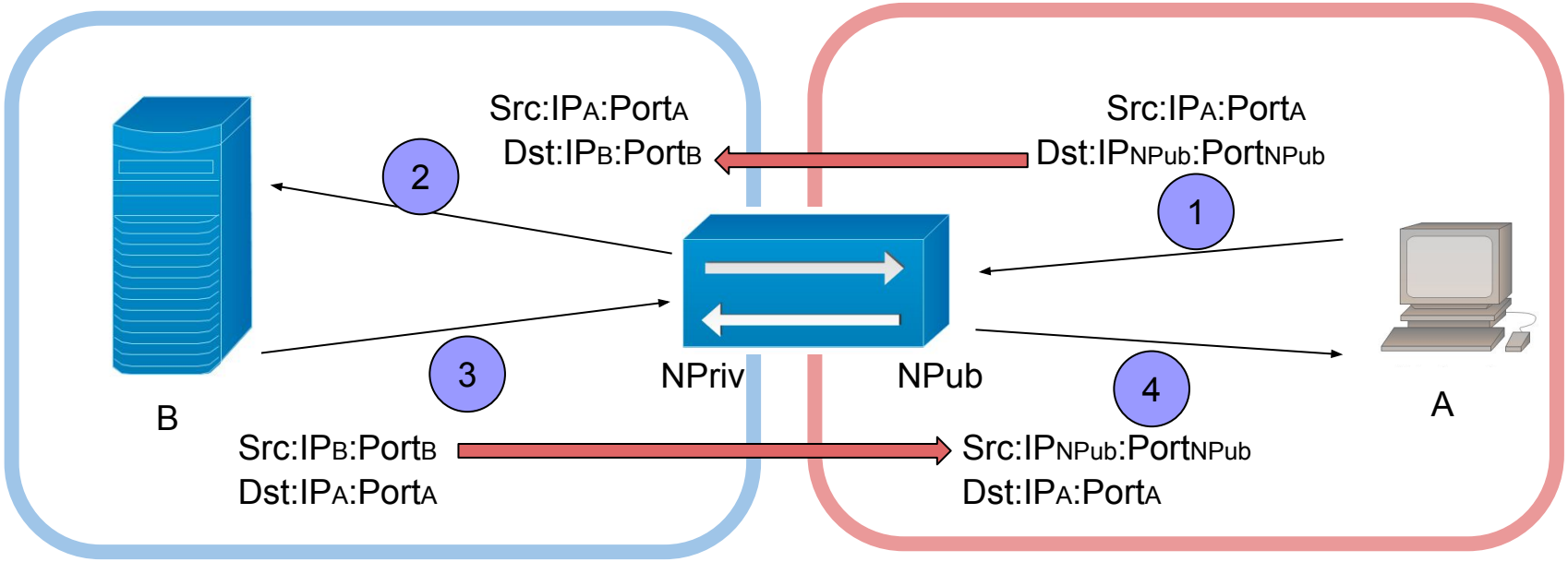
IPA:PortA <-> IPNPub:PortNPub1 | IPB:PortB

Puede cambiar el puerto, IP o ambas del origen.

DNAT (Destination NAT)

- Utilizado cuando un dispositivo en una red pública quiere iniciar una conexión hacia uno que se encuentra en una red privada
- Típicamente el de la red privada es un servidor de algún tipo

(Tengo un servidor interno y quiero accederlo desde afuera)



IPB:PortB <-> IPNPub:PortNPub | *

En este caso, hay que configurar MANUALMENTE en el NAT la asociacion de IP-puerto que quiero mapear con un IP-puerto de la red interna.
Esto tambien se llama port-forwarding.

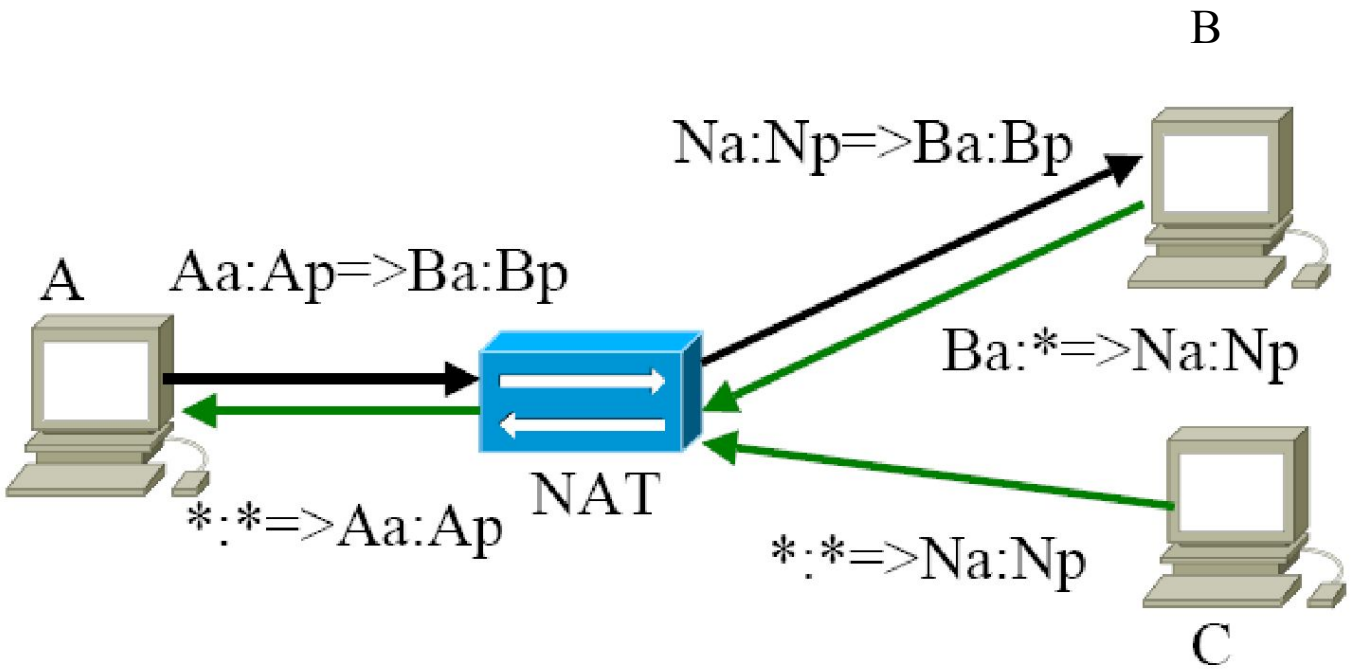
Tipos de NAT

- Tipos de NAT (SNAT)
  - Full Cone
  - IP Restricted NAT
  - Port restricted NAT
  - Symmetric

Los 3 primeros difieren del ultimo en que pasa cuando alguien externo a la red manda un paquete al router que hace NAT.
Full Cone es cuando cualquier IP-puerto que me mande algo a la IP-puerto que SNAT habia abierto, el paquete pasa.
Por ejemplo, mando algo desde mi laptop a un servidor con IP 8.8.8.8. La IP 7.7.7.7 podria contestar a ese IP-puerto del cual salio el paquete en mi router. El tema aca es que si se corta la conexion con el servidor original (8.8.8.8), entonces 7.7.7.7 no me puede mandar mas. Por este motivo, generalmente se usa con UDP.
IP Restricted NAT permite que la misma IP publica me pase paquetes desde cualquier puerto, pero siempre desde la misma IP.
Port Restricted NAT solo permite que me siga mandando paquetes la misma IP desde el mismo puerto.
El Symmetric es el mas seguro, y el que mas se usa (muy parecido a port-restriced)

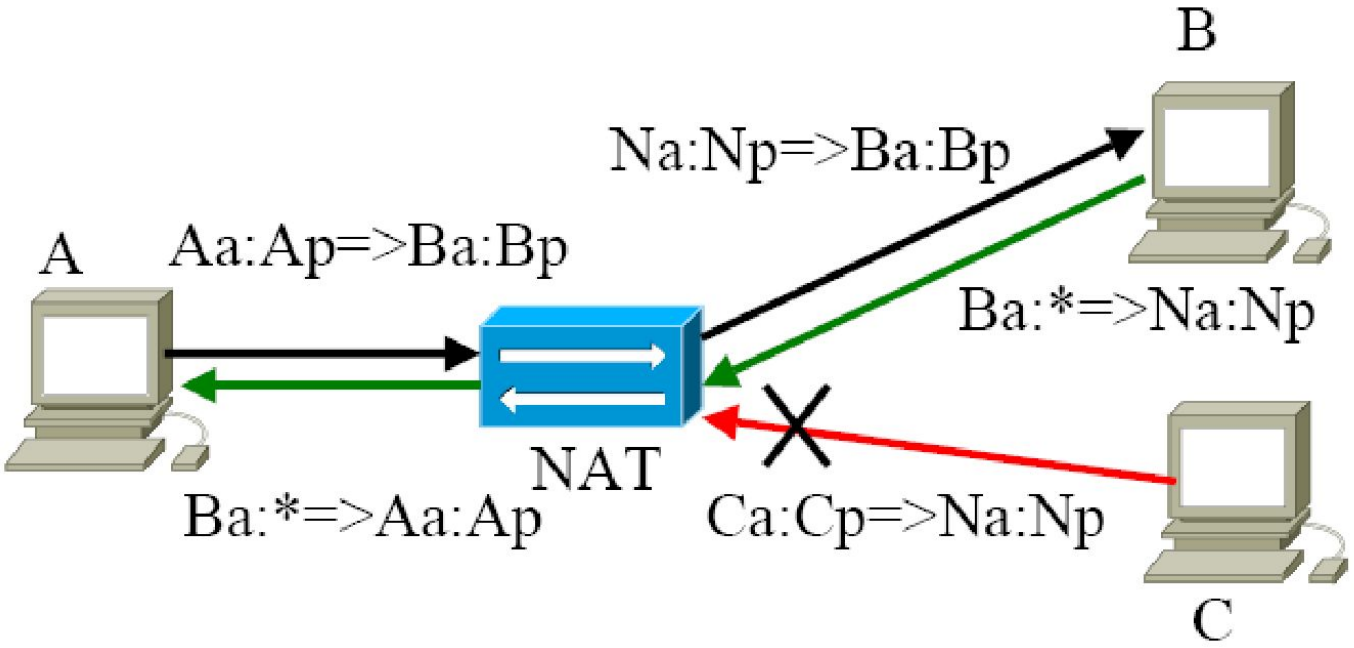
NAT – Full Cone (Static NAT)

- a=address, p=port
- Muy poco restrictivo para conexiones entrantes
- Mapeo estático para servidores



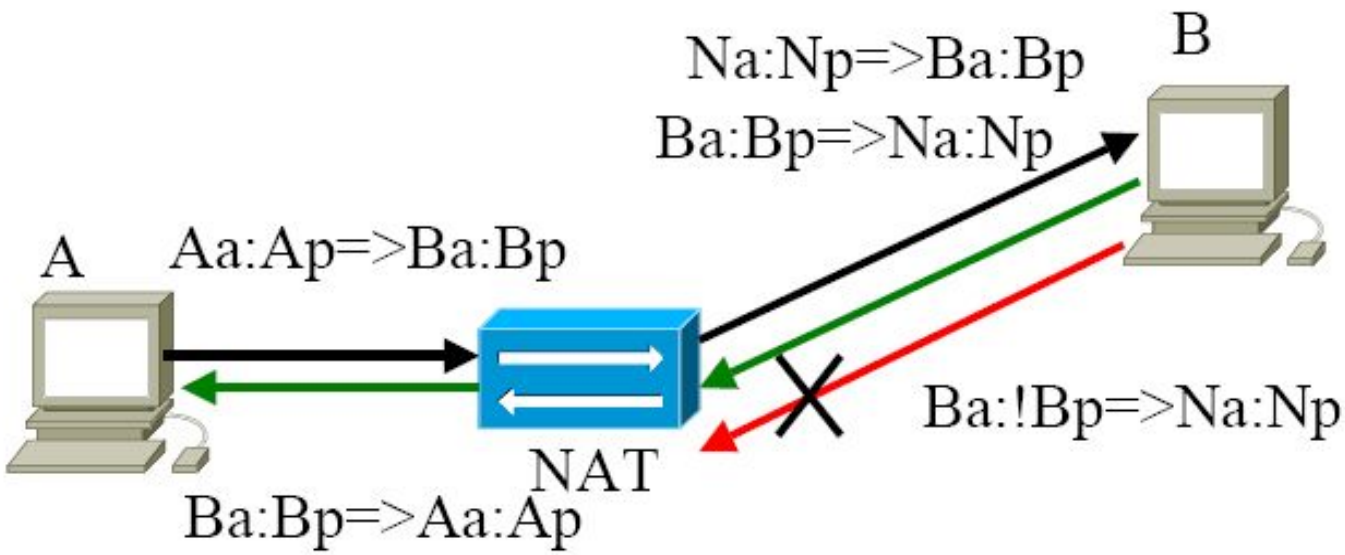
NAT – IP Restricted

- a=address, p=port
- Solo restringe la IP entrante no el puerto



NAT – Port Restricted

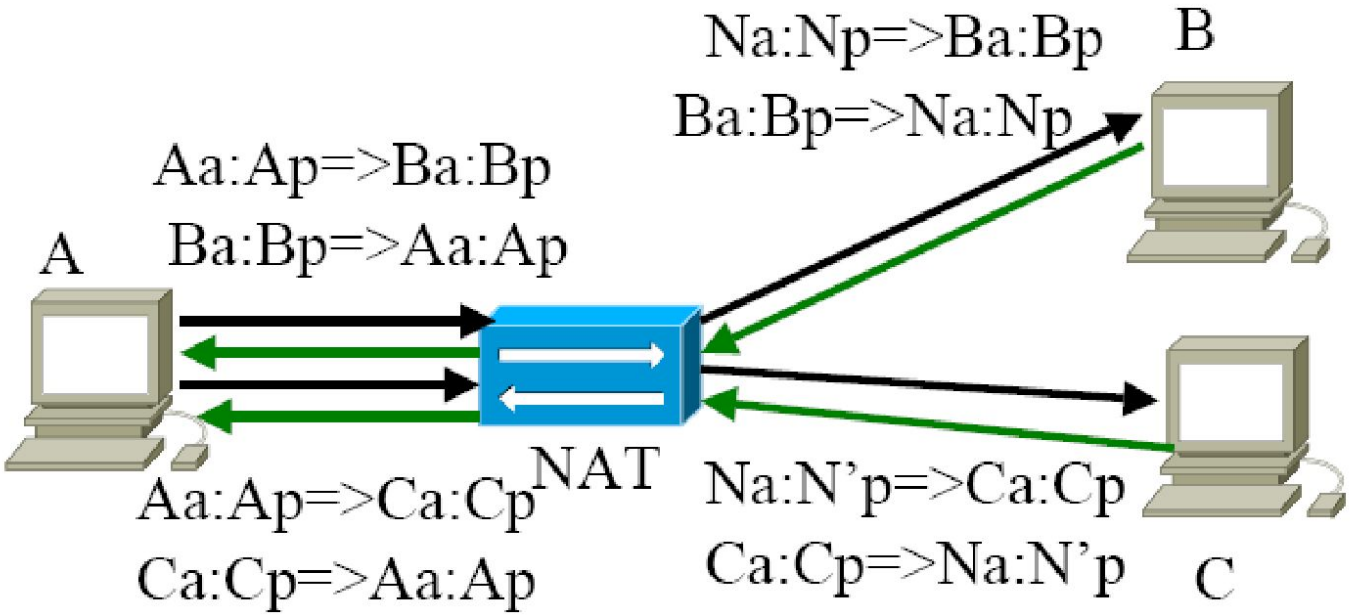
- a=address, p=port
- Restringe la IP entrante y el puerto



NAT – Symmetric

- a=address, p=port
- Cada nueva conexión genera un nuevo puerto

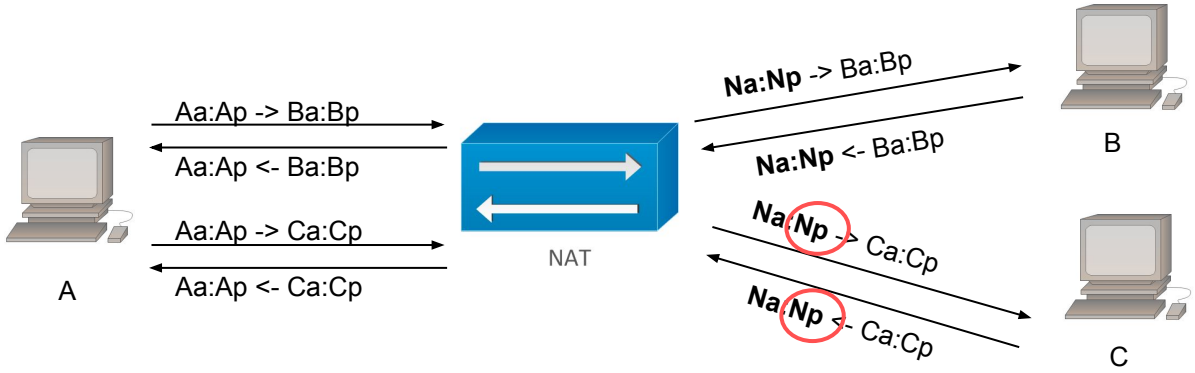
Cada vez que un host en una red privada quiera hacer una nueva conexion, el NAT utiliza un nuevo puerto.
Es decir, symmetric abre un nuevo puerto por cada destino, y ademas es port-restricted.



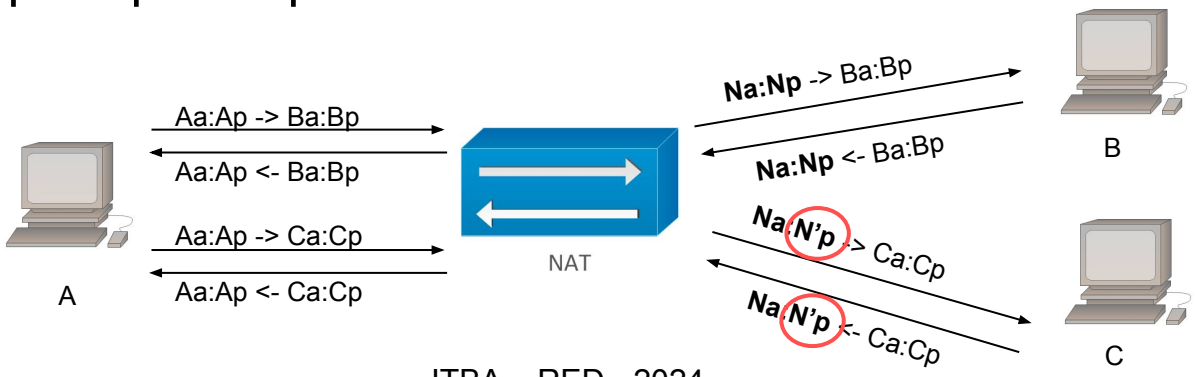
NAT – Symmetric vs Port Restricted

- En Full Cone, IP Restricted y Port Restricted la asociación es de Aa:Ap a Na:Np para todos los Xa:Xp que se conecten desde ese mismo Aa:Ap. Trata de respetar que Np = Ap

Basicamente, trata de usar el mismo puerto para NATear al mismo host de una erd



- En Symmetric para cada Xa:Xp que se conecte desde Aa:Ap se elige un Na:Np distinto por más que el Aa:Ap sea el mismo. Nunca es Np = Ap excepto de casualidad.



Symmetric es mucho mas seguro porque el servidor B no le puede decir al servidor malicioso D "fijate que A tiene este puerto abierto".
La desventaja es que terminas usando muchos mas puertos (entonces en muchos casos necesitas mas IP).

NAT Traversal

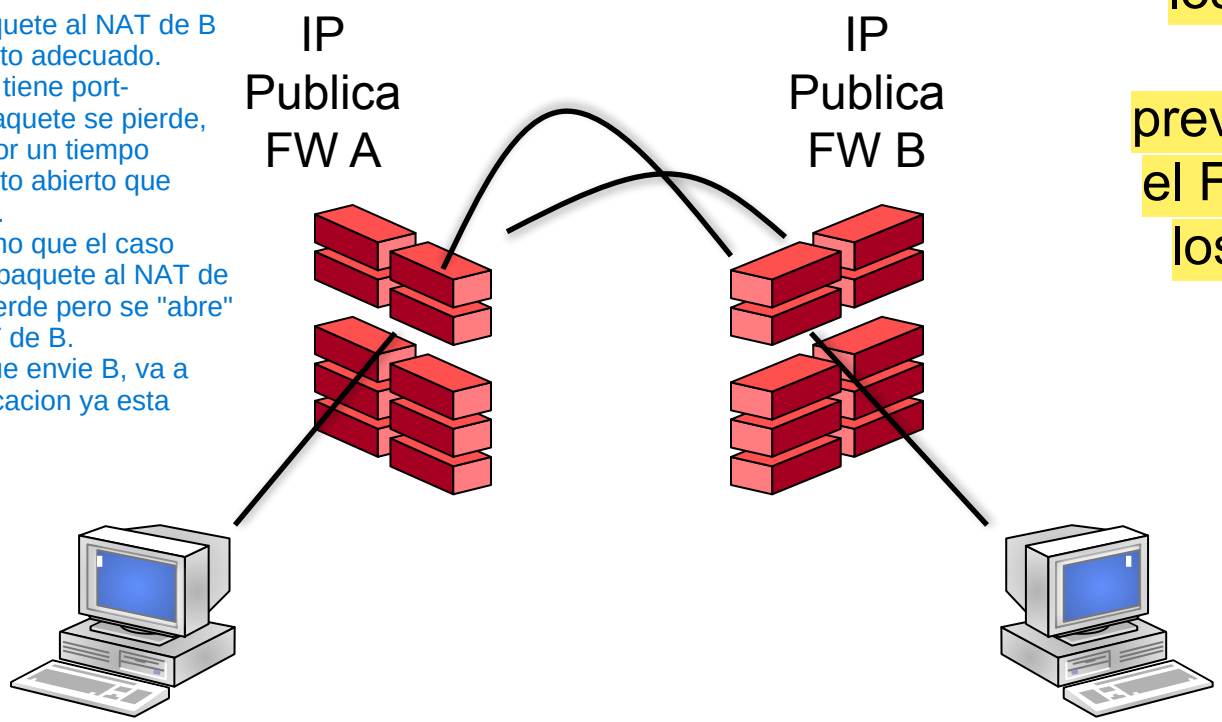
- ¿Cómo conectamos dos PCs que están atrás de un Firewall?
- Se debe buscar una solución para servicios del tipo P2P o VOIP
- No es posible crear reglas de NAT entrantes para todos los clientes posibles

- Soluciones
  - UDP Hole Punching
  - STUN
  - TURN
  - ICE

UDP Hole (sin servidor)

El host A en una red privada se quiere conectar con el host B en una red privada.
El host A sabe la ip publica y puerto de la NAT de B y viceversa.
El host A envia un paquete al NAT de B en la IP publica y puerto adecuado.
Como el NAT de B no tiene port-forwarding, ese ese paquete se pierde, PERO el NAT de A (por un tiempo limitado) tiene un puerto abierto que espera paquetes de B.
El host B hace lo mismo que el caso anterior, le manda un paquete al NAT de B y ese paquete se pierde pero se "abre" ese agujero en el NAT de B.
El proximo paquete que envie B, va a llegar a A y la comunicacion ya esta establecida.

Se conocen los puertos UDP previamente y el Firewall no los cambia



UDP Hole (sin servidor)

Este comando hace UDP Hole Punching. Se puede ver como se intenta varias veces hasta que contesta pampero.

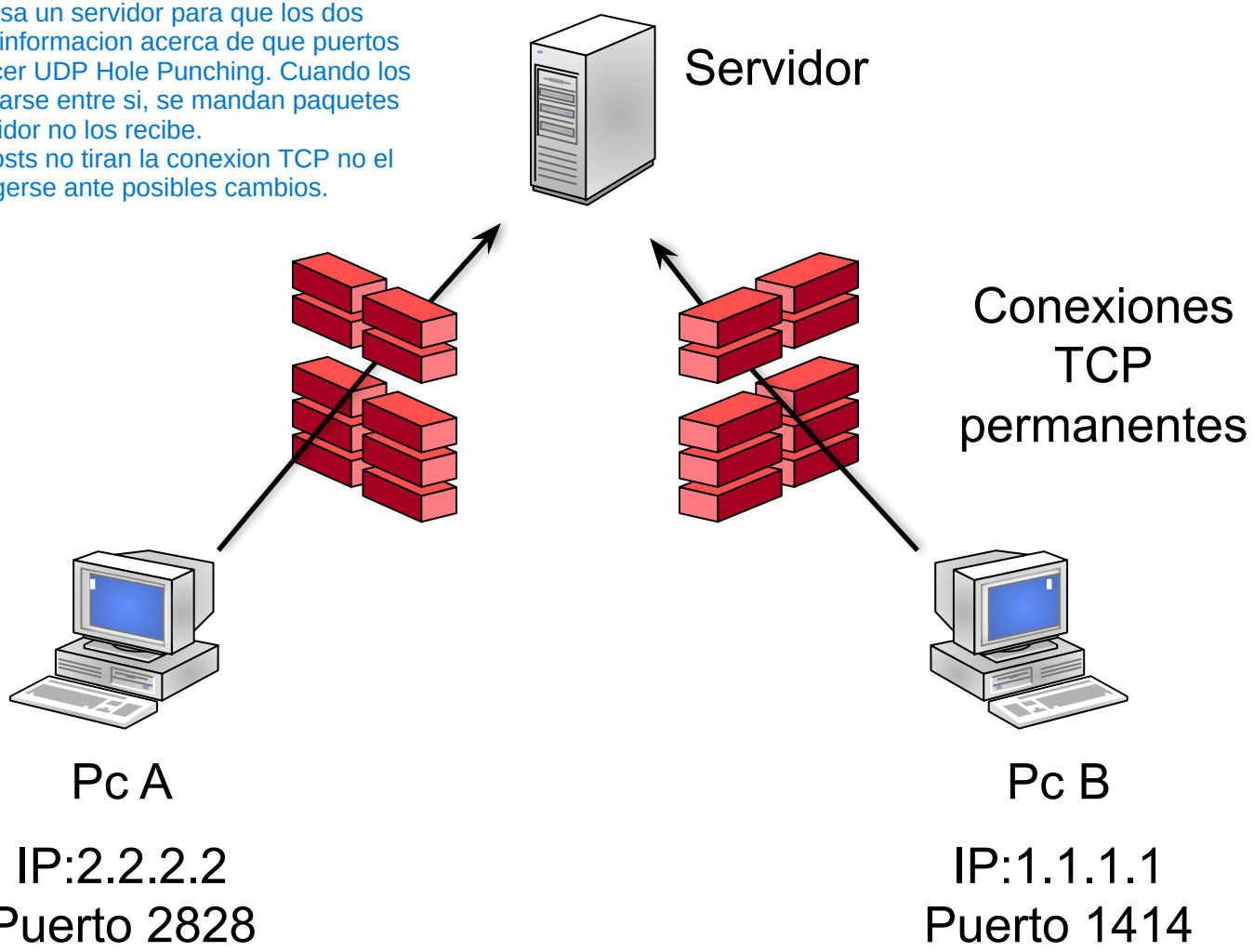
```
srv@srv-nb:~$ sudo tcpdump udp -i eth2 | grep 4000
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
00:57:43.481685 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
00:57:45.482073 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
00:57:47.482508 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
00:57:49.482922 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
00:57:51.483405 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
00:57:53.483781 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
00:59:09.492181 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
00:59:11.064525 IP pampero.it.itba.edu.ar.40000 > srv-nb.local.40001: UDP, length 20
00:59:11.492727 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
00:59:12.062764 IP pampero.it.itba.edu.ar.40000 > srv-nb.local.40001: UDP, length 20
00:59:12.492940 IP srv-nb.local.40001 > pampero.it.itba.edu.ar.40000: UDP, length 20
```

user1@left $ nat-traverse 40001:pampero.itba.edu.ar:40000

user2@pampero $ nat-traverse 40000:ejemplo.dyndns.com:40001

UDP Hole punching (con servidor)

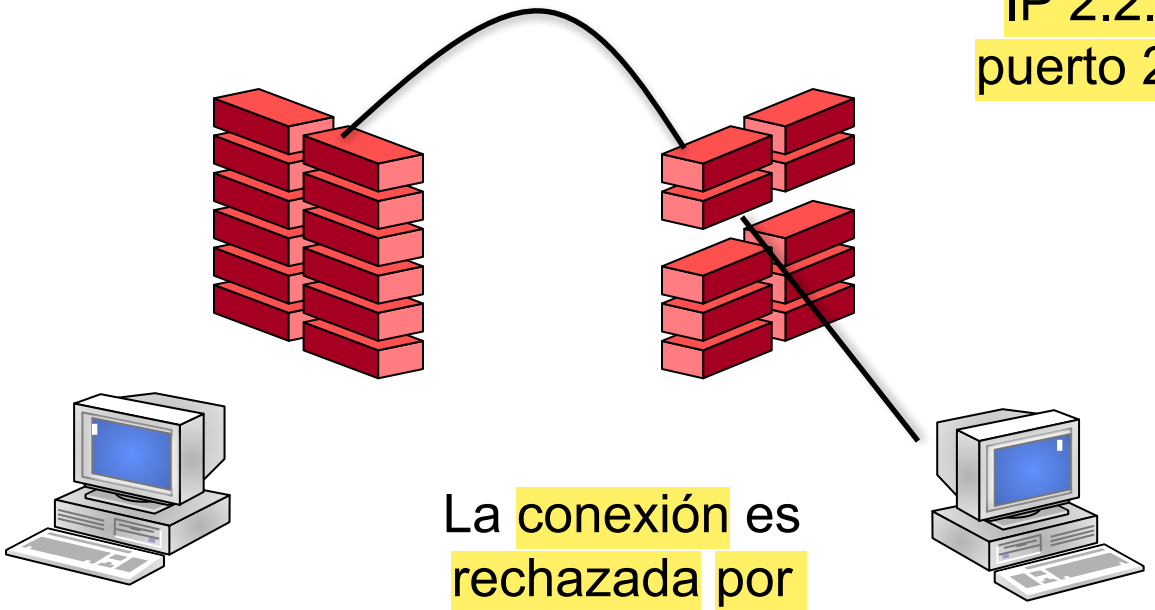
La idea es que se usa un servidor para que los dos hosts intercambien informacion acerca de que puertos van a usar para hacer UDP Hole Punching. Cuando los hosts logran conectarse entre si, se mandan paquetes entre ellos y el servidor no los recibe.
Sin embargo, los hosts no tiran la conexion TCP no el servidor para protegerse ante posibles cambios.



UDP Hole punching (con servidor)

El servidor pasa las coordenadas de la Pc A a la Pc B

La Pc B Intenta una conexión a la IP 2.2.2.2 puerto 2828



La conexión es rechazada por el Firewall de A

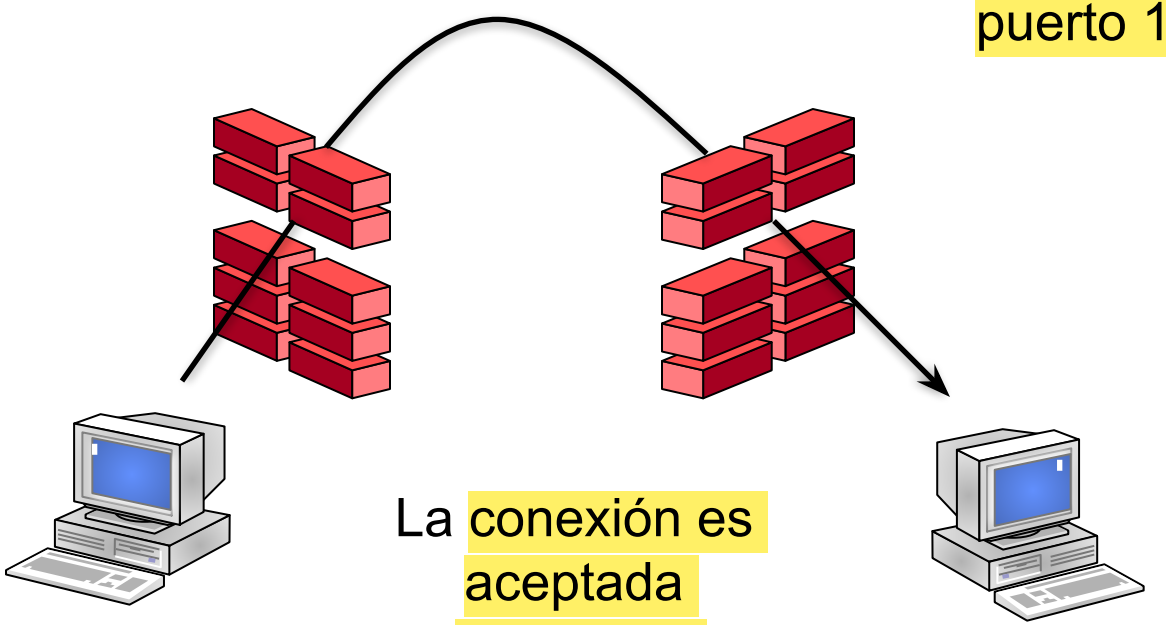
Pc A

IP:2.2.2.2
Puerto 2828

UDP Hole punching

El servidor pasa las coordenadas de la Pc B a la Pc A

La Pc A intenta una conexión a la IP 1.1.1.1 puerto 1414



La conexión es aceptada porque es esperada por el firewall de B

Pc A

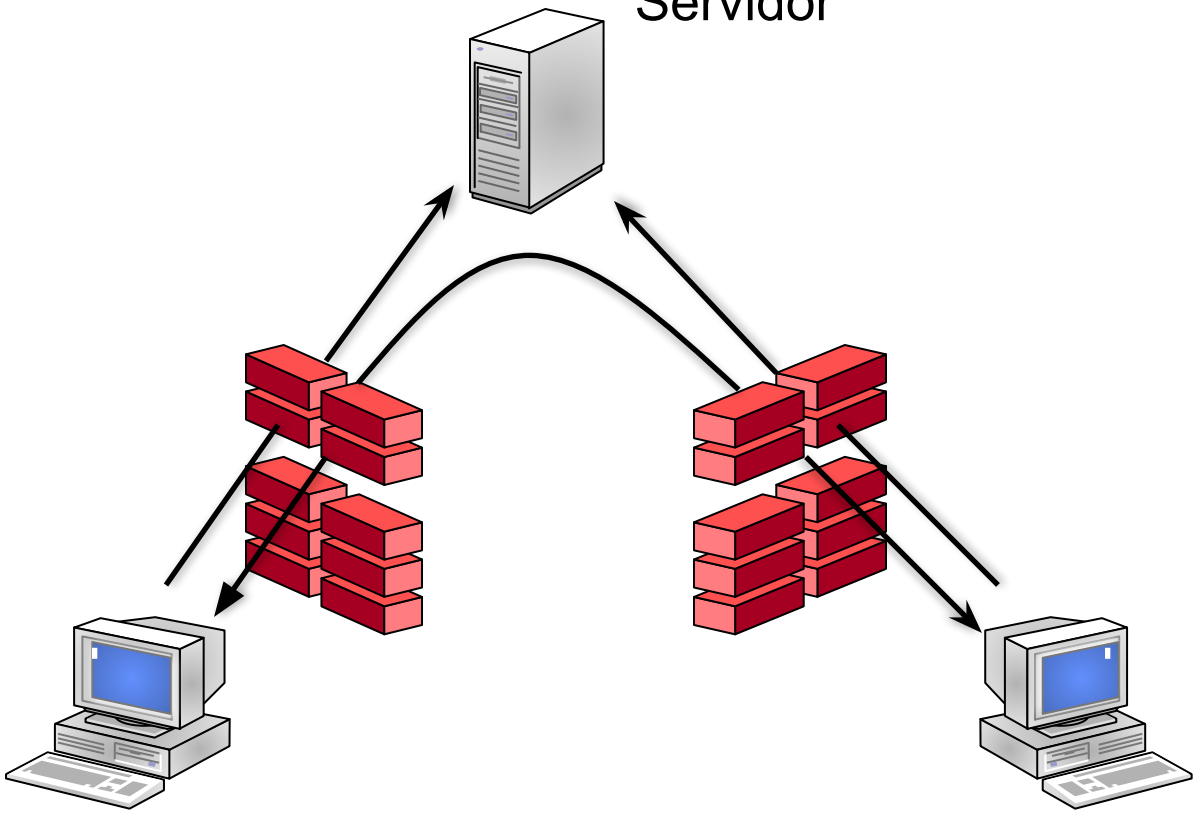
IP:2.2.2.2
Puerto 2828

Pc B

IP:1.1.1.1
Puerto 1414

UDP Hole punching

Se mantienen las conexiones TCP por posibles cambios



Pc A

IP:2.2.2.2
Puerto 2828

Pc B

IP:1.1.1.1
Puerto 1414

UDP Hole Punching

- Esta técnica no esta estandarizada
- Si no funciona esta técnica se debe pasar los datos a través del server (relay)
  - Alto uso de recursos de hardware
  - Limitaciones de ancho de banda y latencia

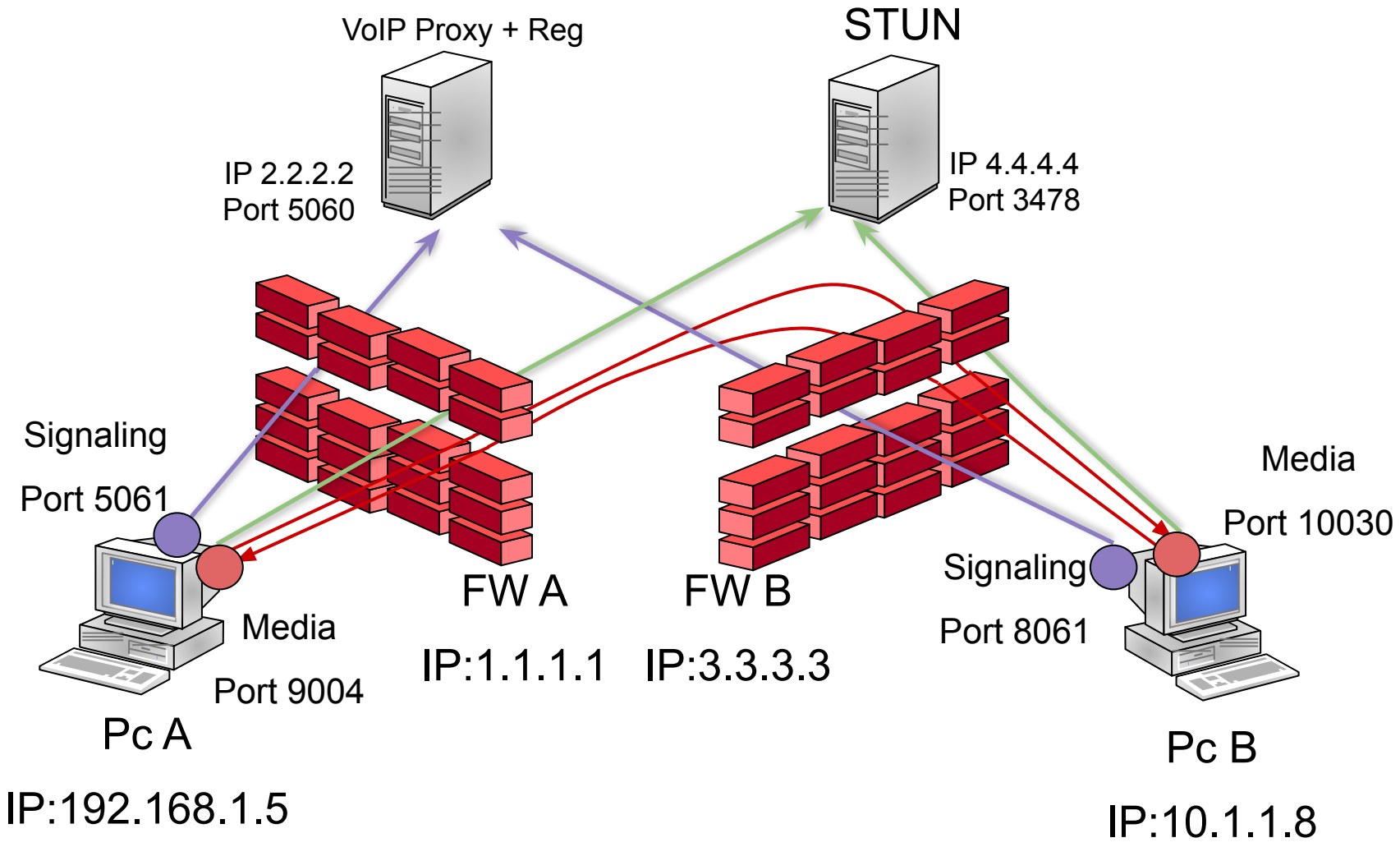
STUN

Cuando un host en una red privada no sabe su IP publica (de su NAT), le pregunta a un STUN server. Ademas, el STUN server me abrio un "hole" en mi NAT, y asi puede adivinar que tipo de NAT tengo.

El STUN me devuelve la IP y puerto a traves del cual yo me estoy conectando.
Algun host puede conectarse a mi a traves de esa IP y puerto. Esto funciona para todos los tipos de NAT menos symmetric.

- Session Traversal Utilities for NAT
- RFC 5389 (antes RFC 3489)
- Protocolo para descubrir distintos tipos de NAT
- Define los pasos a seguir y tipos de mensajes a enviar
- Permite a los clientes conocer la IP pública y puerto con los que salen sus datos.
- Chequea conectividad entre dos endpoints

UDP Hole punching (caso VoIP/P2P)



UDP Hole Punching con STUN (VoIP)

1.- A y B se registran en la PBX
2.- Antes de mandar la llamada A envía desde su puerto de Media la consulta STUN para conocer su IP:Port y poder incluirlos en el SDP.
3.- A recibe la respuesta del STUN Server
4.- A envía el INVITE a la PBX con To: B y SDP con la IP:Port que recibió en 3
5.- La PBX envía el INVITE a B
6.- B recibe el INVITE y realiza la consulta STUN para conocer su IP y puerto de Media. Como en 2 esta consulta se hace desde el puerto de Media de B.
7.- B contesta al INVITE a través del Proxy con la IP:Port que recibió en 6 en el SDP
8.- A envía Media a la IP:Port que recibió de B. Si B tiene NAT full cone los paquetes pasarán por el hole punched en el paso 6. De lo contrario se debe esperar al paso 9 para que pasen.
9.- B Envía Media a la IP:Port que recibió de A análogo al paso anterior.

STUN

- Existen servidores gratuitos en Internet
  - stun.ekiga.net
  - stun.xten.com
  - stun.voipbuster.com
  - …….
- No suelen ofrecer relay

TURN

Para resolver el problema de cuando hay NAT Symmetric, hay que pasar siempre por el servidor. Esto se llama un TURN SERVER. En este caso estas perdiendo la posibilidad de que una conexion realmente sea P2P.

- Traversal Using Relay NAT
- RFC 5766
- Debería ser la segunda opción en caso que no funcione STUN
- Protocolo para hacer relay de UDP y TCP detrás de NAT
- Única solución para Symmetric NAT

ICE

Es una norma (usada por WebRTC), que busca encontrar la mejor forma para conectarse.
El orden de prioridad es: IP Local, STUN, TURN

- Interactive Connectivity Establishment
- Evalua todos los cantidatos de conexión
  - STUN / TURN / IPs Locales
- Los ordena por prioridad
- Chequea conectividad
- Una vez que se establece la conexión deja de operar

REDES

VPN

VPN

- Virtual Private Network
- Se utiliza para interconectar redes o equipos a través de otras redes.
- Si se atraviesa redes de terceros
  - Aplicar encriptación de datos
- Normalmente se lo denomina túnel porque encapsula la comunicación entre las dos puntas

VPN

- Modos de conexión más comunes
  - Cliente a Sitio
  - Sitio a Sitio

- Tipos soluciones
  - Hardware (Firewall)
  - Software (Cliente – servidor)
  - Mixtas

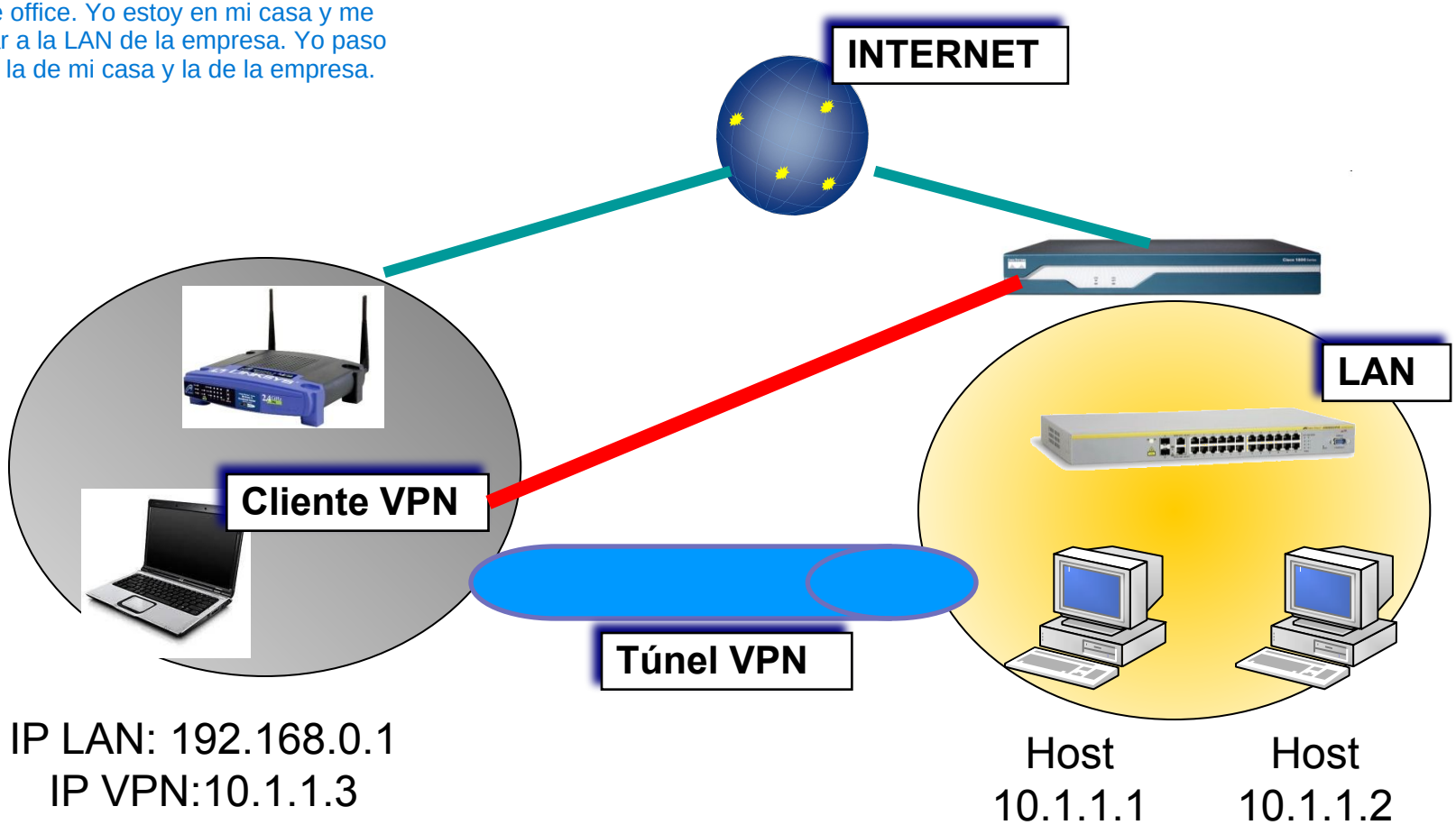
Tipos de VPN

- Cliente a Sitio
  - Uso común:
    - Usuarios móviles
  - Licencias
    - Suele ser por cliente/usuario

- Sitio a Sitio
  - Uso común:
    - Sucursales a través de Internet
    - Interconexión entre empresas
  - Licencias
    - Suele ser por sitio

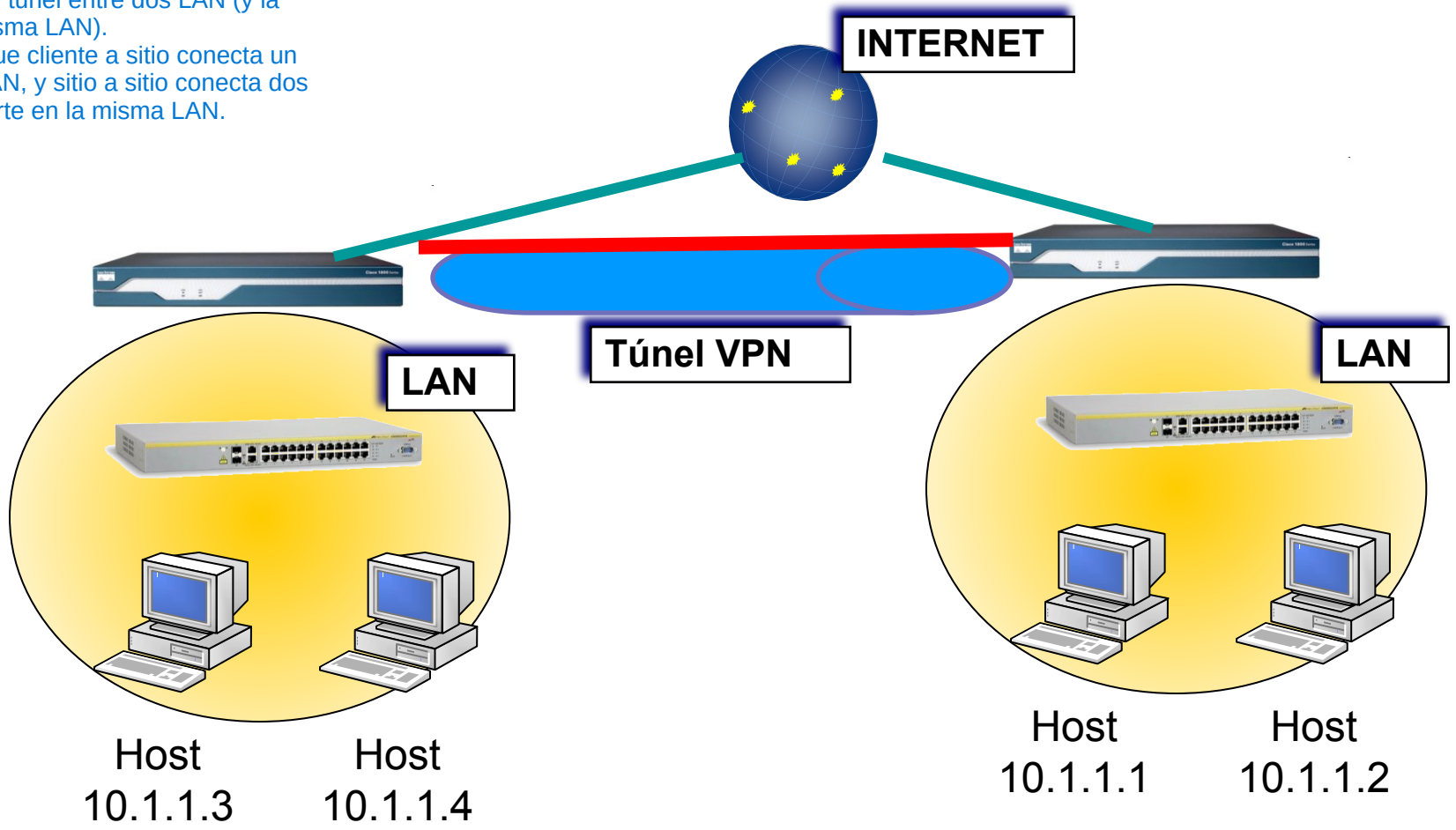
VPN – Cliente a sitio

Ejemplo: home office. Yo estoy en mi casa y me quiero conectar a la LAN de la empresa. Yo paso a tener dos IP: la de mi casa y la de la empresa.



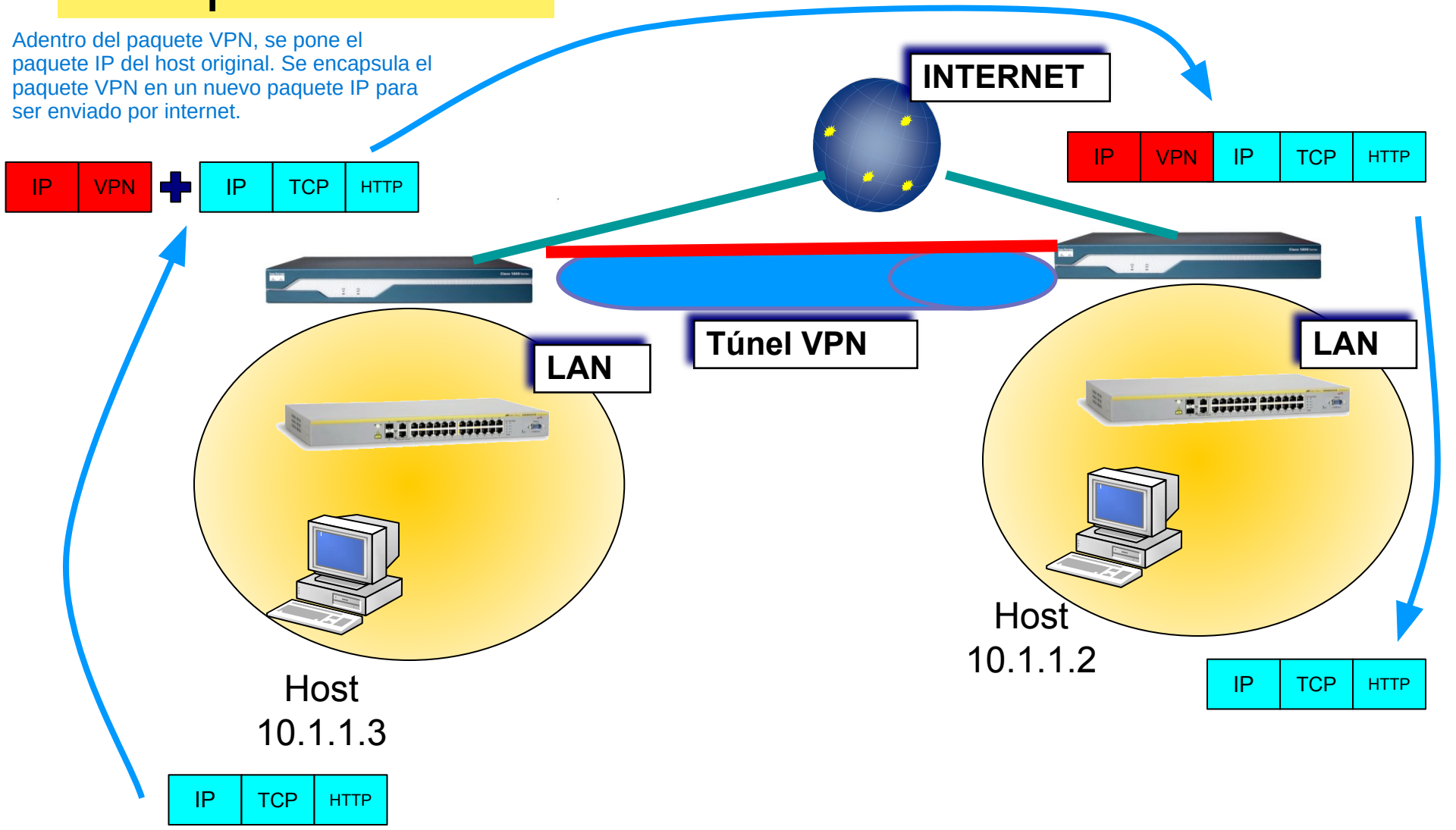
VPN – Sitio a sitio

El firewall arma un tunel entre dos LAN (y la convierte en la misma LAN).
La diferencia es que cliente a sitio conecta un solo host a una LAN, y sitio a sitio conecta dos redes y las convierte en la misma LAN.

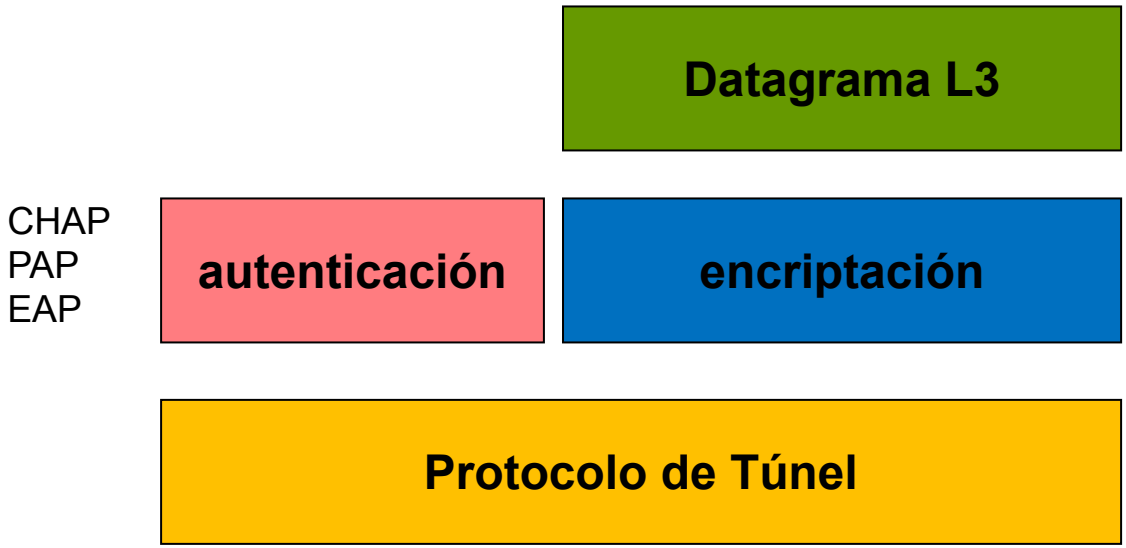


Encapsulamiento

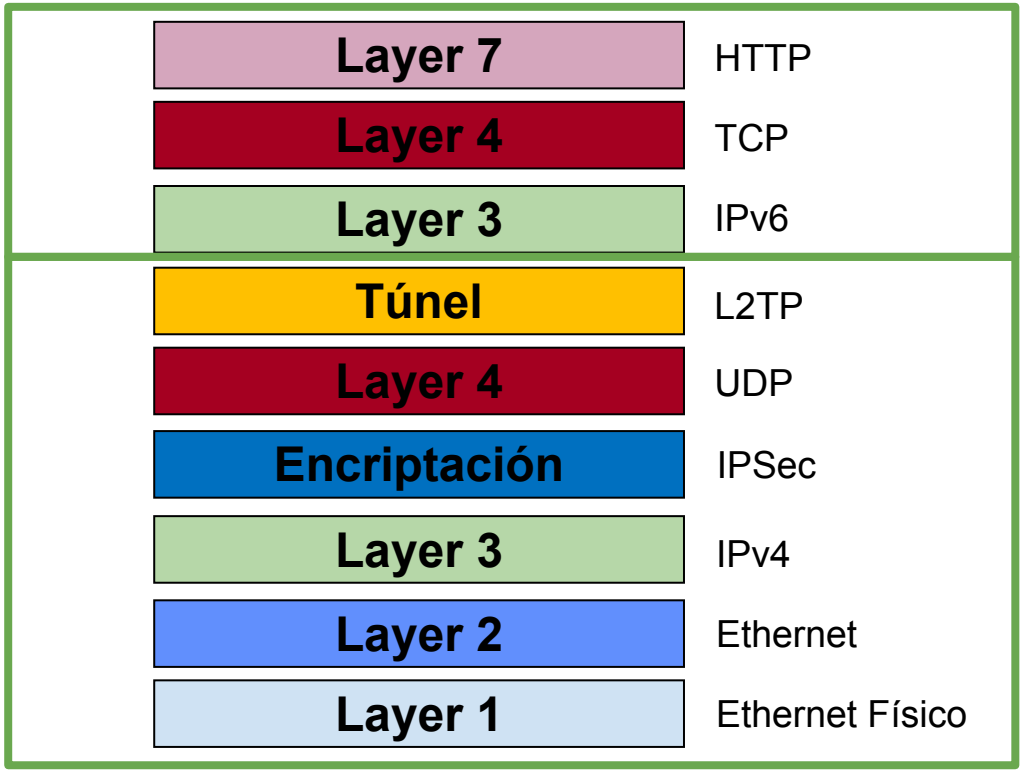
Adentro del paquete VPN, se pone el paquete IP del host original. Se encapsula el paquete VPN en un nuevo paquete IP para ser enviado por internet.



Capas de una VPN

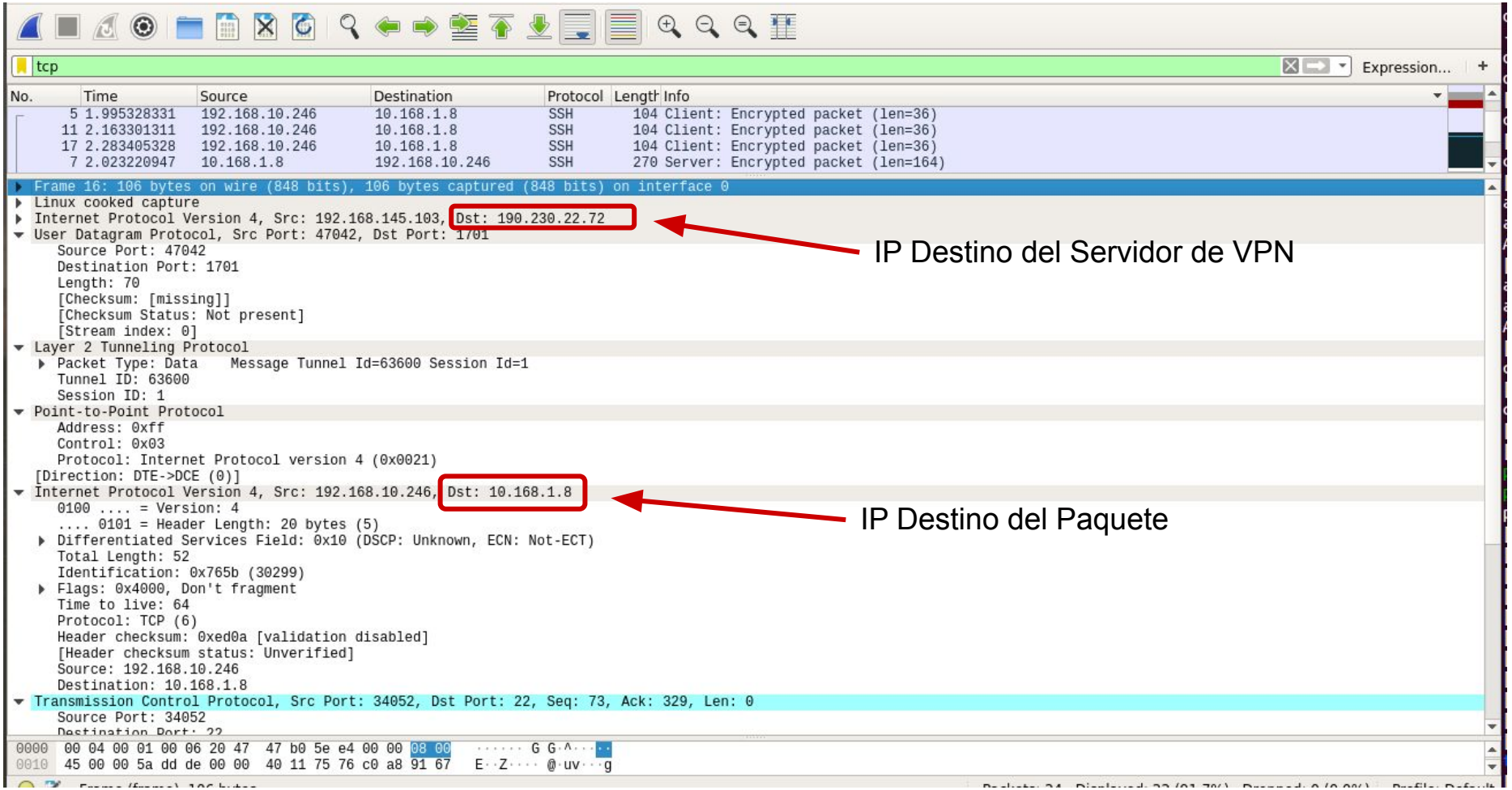


Ejemplo de capas de una VPN



- GRE (Generic Routing Encapsulation)
- L2TP (Layer 2 Tunneling Protocol)
- PPTP (Peer to Peer Tunneling protocol)

Captura de paquetes con VPN



Soluciones de software

- Sistemas operativos

- OpenVPN
  - Licencia GPL
  - Permite crear VPN SSL
  - Usa un solo puerto UDP
  - Pasa a través de Proxy y Firewall
  - No tiene problemas con NAT

Soluciones de Hardware

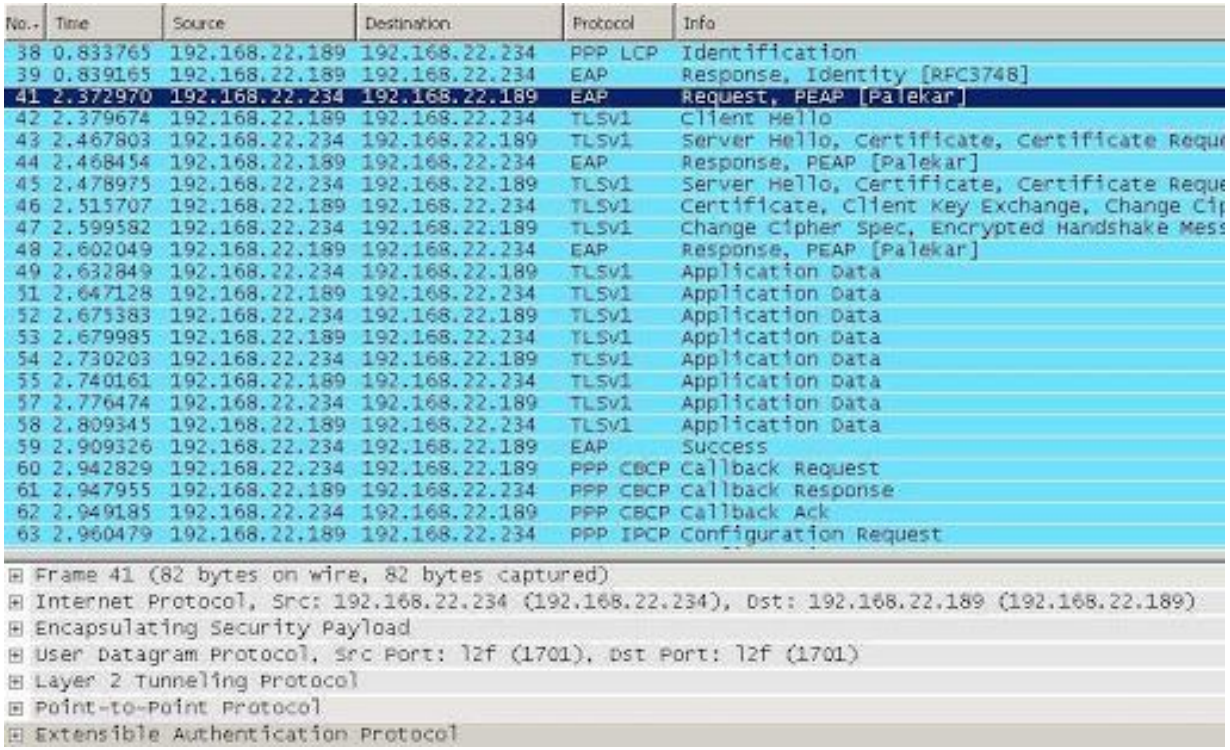
- Firewalls
  - Soportan VPN Cliente-Sitio y Sitio-Sitio
  - Proveen cliente propietario
  - Permiten conexión por VPN SSL (el navegador necesita un plug-in que se instala en la PC)

IPSec

- En 1998 la IETF define los RFC's para las VPN
- IPSec es un conjunto de protocolos para crear VPNs
  - RFC 2401 (IPSec)
  - RFC 2402 (Authentication Header)
  - RFC 2406 (Encapsulating Security Payload)
  - RFC 2408 (ISAKAMP)
  - RFC 2409 (IKE – Internet Key Exchange)

  - Muy complejo de implementar !
  - Corre en espacio Kernel en los SO

Captura de paquetes con VPN



No.	Time	Source	Destination	Protocol	Info
38	0.833765	192.168.22.189	192.168.22.234	PPP LCP	Identification
39	0.839165	192.168.22.189	192.168.22.234	EAP	Response, Identity [RFC3748]
41	2.372970	192.168.22.234	192.168.22.189	EAP	Request, PEAP [Palekar]
42	2.379674	192.168.22.189	192.168.22.234	TLSv1	Client Hello
43	2.467803	192.168.22.234	192.168.22.189	TLSv1	Server Hello, Certificate, Certificate Requ
44	2.468454	192.168.22.189	192.168.22.234	EAP	Response, PEAP [Palekar]
45	2.478975	192.168.22.234	192.168.22.189	TLSv1	Server Hello, Certificate, Certificate Requ
46	2.515707	192.168.22.189	192.168.22.234	TLSv1	Certificate, Client Key Exchange, Change Ci
47	2.599582	192.168.22.234	192.168.22.189	TLSv1	Change Cipher Spec, Encrypted Handshake Mes
48	2.602049	192.168.22.189	192.168.22.234	EAP	Response, PEAP [Palekar]
49	2.632849	192.168.22.234	192.168.22.189	TLSv1	Application Data
51	2.647128	192.168.22.189	192.168.22.234	TLSv1	Application Data
52	2.675383	192.168.22.234	192.168.22.189	TLSv1	Application Data
53	2.679985	192.168.22.189	192.168.22.234	TLSv1	Application Data
54	2.730203	192.168.22.234	192.168.22.189	TLSv1	Application Data
55	2.740161	192.168.22.189	192.168.22.234	TLSv1	Application Data
57	2.776474	192.168.22.234	192.168.22.189	TLSv1	Application Data
58	2.809345	192.168.22.189	192.168.22.234	TLSv1	Application Data
59	2.909326	192.168.22.234	192.168.22.189	EAP	Success
60	2.942829	192.168.22.234	192.168.22.189	PPP CBCP	Callback Request
61	2.947955	192.168.22.189	192.168.22.234	PPP CBCP	Callback Response
62	2.949185	192.168.22.234	192.168.22.189	PPP CBCP	Callback Ack
63	2.960479	192.168.22.189	192.168.22.234	PPP IPCP	Configuration Request

⊞ Frame 41 (82 bytes on wire, 82 bytes captured)
⊞ Internet Protocol, Src: 192.168.22.234 (192.168.22.234), Dst: 192.168.22.189 (192.168.22.189)
⊞ Encapsulating Security Payload
⊞ User Datagram Protocol, Src Port: l2f (1701), Dst Port: l2f (1701)
⊞ Layer 2 Tunneling Protocol
⊞ Point-to-Point Protocol
⊞ Extensible Authentication Protocol

Protocolos de Autenticación de VPN

- EAP (Extensible Authentication Protocol) - Más Seguro
- CHAP (Challenge Handshake Protocol)
- PAP (Password Authentication Protocol) - Menos seguro

- Intercambian credenciales entre el cliente y el servidor VPN
- Handshake e intercambio de claves o shared secret

Ejemplo de conexión (1)

- Interface Eth0 de Notebook
  - IP: 192.168.0.49
  - GW: 192.168.0.1
  - MASK: 255.255.255.0
  - Serv DNS: 8.8.8.8

IPv4 Tabla de enrutamiento Antes de conectarme a la VPN, esta era mi tabla de ruteo

Rutas activas:

Destino de red	Máscara de red	Puerta de enlace	Interfaz	Métrica
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.49	55
192.168.0.0	255.255.255.0	En vínculo	192.168.0.49	311
192.168.0.49	255.255.255.255	En vínculo	192.168.0.49	311
192.168.0.255	255.255.255.255	En vínculo	192.168.0.49	311
255.255.255.255	255.255.255.255	En vínculo	127.0.0.1	331
255.255.255.255	255.255.255.255	En vínculo	192.168.0.49	311

En vinculo = conexion local

Ejemplo de conexión (2)

- Conexión a empresa
  - Host: vpn.miempresa.com.ar (181.22.33.44)
  - Sabemos que dentro de la empresa están las redes:
    - 10.4.0.0/16
    - 10.50.50.0/24
    - 10.40.1.0/24
    - 192.168.0.0/16

Ejemplo de conexión (2)

- Creamos la conexión de VPN

- Interface Eth0 de Notebook (no cambió)
  - IP: 192.168.0.49
  - GW: 192.168.0.1
  - MASK: 255.255.255.0
  - Serv DNS: 8.8.8.8

- Interface Eth1 de Notebook

  Aparece esta nueva interfaz, que tiene una nueva IP

  - IP: **10.50.50.152**
  - GW: (no tiene)
  - MASK: 255.255.255.0

Ejemplo de conexión (3)

- Nueva tabla de ruteo luego de la conexión VPN

IPv4 Tabla de enrutamiento

Rutas activas:

Destino de red	Máscara de red	Puerta de enlace	Interfaz	Métrica
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.49	55
10.4.0.0	255.255.0.0	En vínculo	10.50.50.152	2
10.40.1.0	255.255.255.0	En vínculo	10.50.50.152	2
181.22.33.44	255.255.255.255	192.168.0.1	192.168.0.49	55
192.168.0.0	255.255.0.0	En vínculo	10.50.50.152	2
192.168.0.0	255.255.255.0	En vínculo	192.168.0.49	311
192.168.0.49	255.255.255.255	En vínculo	192.168.0.49	311
192.168.0.255	255.255.255.255	En vínculo	192.168.0.49	311
255.255.255.255	255.255.255.255	En vínculo	127.0.0.1	331
255.255.255.255	255.255.255.255	En vínculo	192.168.0.49	311

En verde las que ya tenia, en rojo las nuevas.
La IP de interfaz apunta a la interfaz virtual nueva de mi cliente VPN (10.50.50.152)
Tambien agrego las nuevas redes disponibles (192.168.0.0, 10.4.0.0, 10.40.1.0), poniendo la interfaz virtual como interfaz.
La regla amarilla es la que se me agrega cuando ya se la direccion real de la VPN destino, esto es porque despues de que yo mande el primer paquete VPN, ese servidor VPN me contesta por esa IP y mi tabla de ruteo ya se la aprende.
Notar que la interfaz para comunicarse con esa IP nueva es la de mi computadora.
NOTA: en realidad, la regla amarilla no es del todo necesaria porque para eso ya tengo el default gateway.
NOTA2: fijarse que tanto la red de la VPN como la de mi casa es 192.168.0.0. Por default, los paquetes se envian por donde la mascara de red es mas especifica (en este caso la de mi casa).

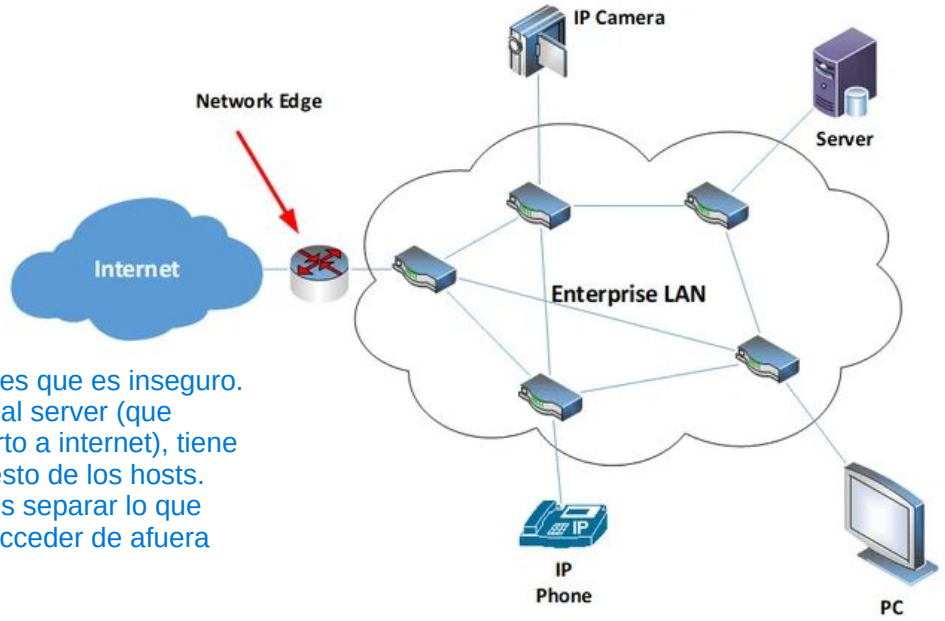
REDES

DMZ

Network Edge

- Borde entre red interna (Hogareña o Empresa) y externa (Internet)
- Podría ser más que el router…

- Cual es el problema con este esquema?



El problema del esquema es que es inseguro. Si alguien malicioso entra al server (que suponemos que esta abierto a internet), tiene acceso directo a todo el resto de los hosts. Lo que buscamos haces es separar lo que queremos que se pueda acceder de afuera con lo que no.

Network Edge

- Exponer servicios a internet
- Evitar ataques a red interna
- Controlar el acceso

DMZ

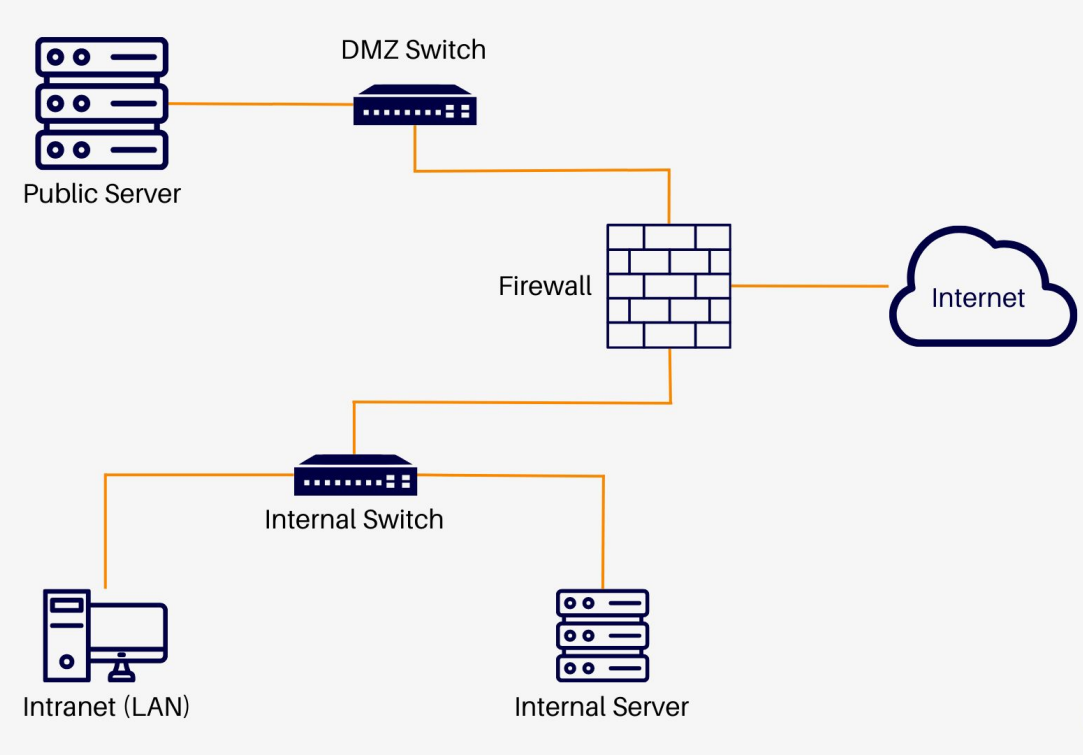
- Demilitarized Zone (Zona Desmilitarizada)
- Tierra de nadie
- Se usa para exponer servicios externos de manera controlada

DMZ

- Topología con 1 Firewall

El firewall tiene 3 interfaces
- Para internet
- Para red interna
- Para DMZ

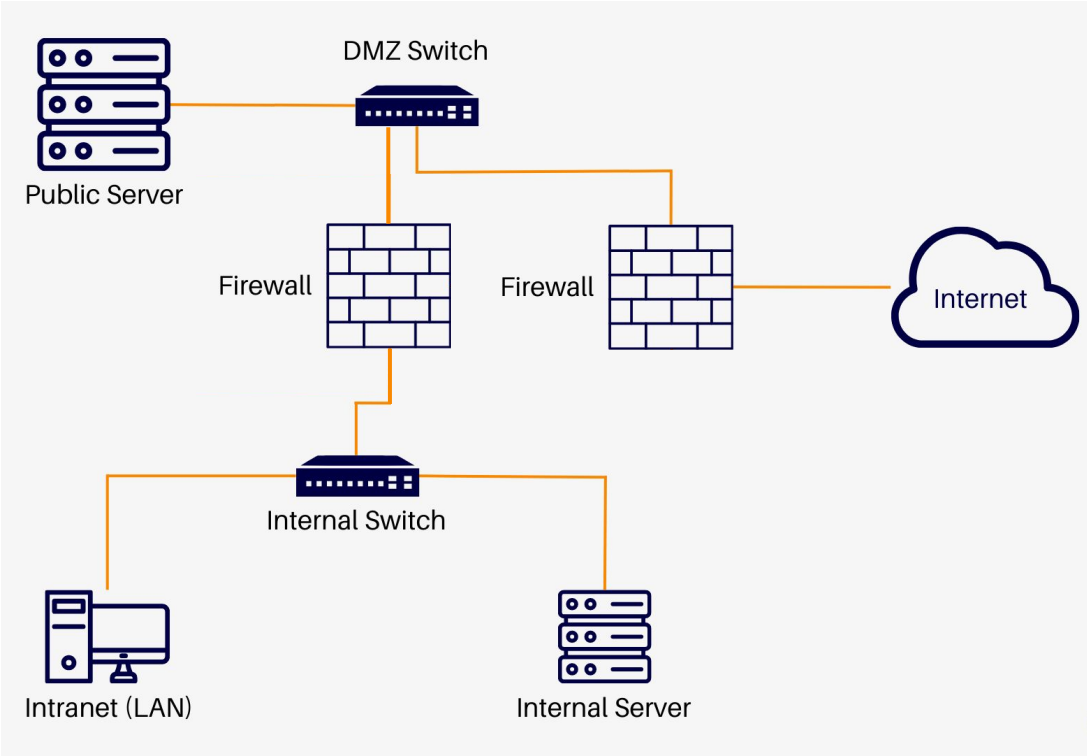
El problema aca es que si alguien puede hackear el firewall, entonces ya tiene acceso a nuestros hosts locales.



DMZ

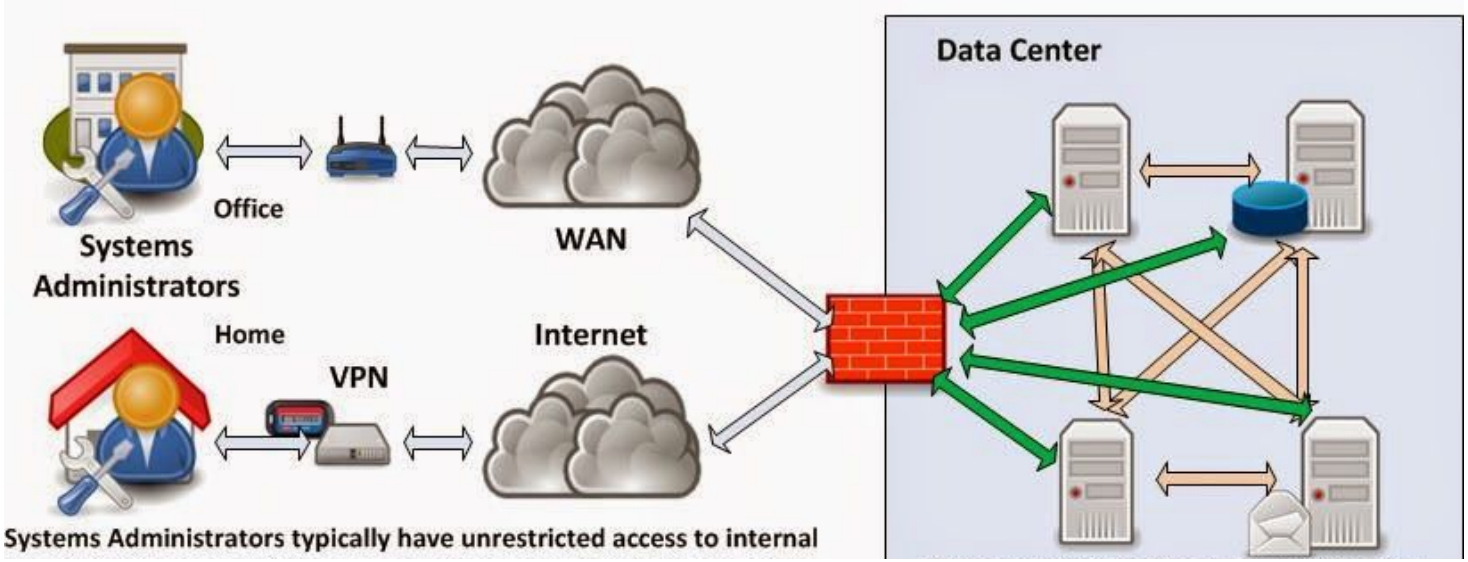
- Topología con 2 Firewalls

Este es mas seguro, porque para llegar a los hosts hay que pasar a traves de dos firewalls.



DMZ - Jump Server

- Jump Server controla el acceso a los distintos servidores de la DMZ
- Controla el Side Channel attack donde desde un servidor se accede al resto



Antes de la DMZ, uno tiene un firewall. Los administradores se conectan a los servidores, que estan en la misma red. El problema en este caso es que si hackean un servidor, tienen acceso a todo el resto de los servidores (esto se llama Side Channel Atack)

DMZ - Jump Server

- Jump Server permite el acceso a usuarios específicos a servidores específicos
- Auditado (queda registrado quien accede y cuando)
- Muy seguro, sólo levanta y abre los puerto y servicios mínimos

Se usa una interfaz aparte, en otra VLAN, que puede autenticar con 2FA y tiene permisos puntuales para cada administrador. La conexion entre servidores se da por otras redes que ni el administrador tiene acceso.