Quiesta è una modifica

Jav326c

Grep lo uso per il contenuto

Find per le caratteristiche del file

Which command deve ritornare la funzionare che effettua un comando esterno (which ls)

Apropos ipsec

Kathara:

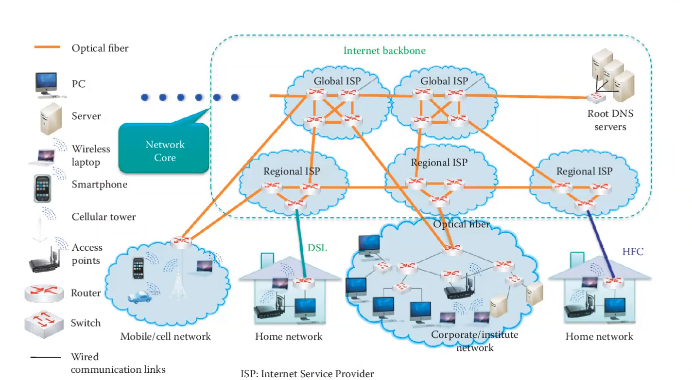
broadcast address -> prendi la network part, inserisci tutti 1 nella parte host ed ottieni l’indirizzo.

L’architettura di internet è composta da diversi Internet Service Provider divisi gerarchicamente.  
Esistono i Global ISP (livello 1) che costituiscono la dorsale di internet ed hanno velocità di trasmissione dati molto elevate. Questi sono collegati fra di loro attraverso cavi in fibra ottica. La connessione tra diversi ISP è effettuata tramite gli IXP (internet Exchange point).   
Dopo di loro esistono ISP di secondo o terzo livello, chiamati regionali o locali. Non sono proprietari della dorsale ma usufruiscono della banda acquistandola da ISP di primo livello.

L’utente accede ad internet attraverso il POP (point of presence).

Questa è una prova da linux

github323232



Esistono diverse aree dell’internet:

* Edge: parte della rete dove vengono prodotti dati verso internet (pc, laptop)
* Core: responsabile di fornire servizi internet e gestore dei pacchetti in internet.
* Access: intermedio tra edge e core

Internet non è composto solo da device connessi fra loro ma anche da protocolli che gestiscono i failure che questi device possono avere. I protocolli gestiscono e si adattano a quello che succede nella rete, notando tutto e cercando di aggiustarlo.

Il funzionamento avviene tramite l’utilizzo di algoritmi distribuiti.

I protocolli descrivono le procedure che devono seguire i vari messaggi quando vengono scambiati, il formato dei messaggi e il timing durante l’evento che scaturisce il protocollo.  
Sono quindi delle regole da seguire per rendere possibile la comunicazione. Gestisce il tipo e le sequenze di messaggi che devono essere scambiati, il formato di questi messaggi e il timing che gestisce gli eventi che avvengono durante una comunicazione.

Per i protocolli vale il concetto di modularizzazione, cioè la presenza di più protocolli per ogni livello della rete. Questo permette di isolare i dettagli di ogni implementazione dei protocolli e permette ai vari livelli di avere delle fasi di manutenzione o aggiornamento senza creare disturbi.

La rete ethernet comporta che ogni dispositivo abbia un suo identificativo che è il MAC address composto da 48 bit. Ogni pacchetto nella rete ethernet ha un formato fisso.   
Gli host che sono connessioni agli stessi cavi basati su una rete ethernet sono raggiungibili tra di loro (anche attraverso l’utilizzo di device come hub, router, bridge e switch).  
La rete ethernet costituisce quello è che il Brodcast domain, cioè un dominio dove ogni host può raggiungere qualsiasi altro host connesso alla rete.

Nonostante internet è composto da tanti dispositivi connessi tra di loro, non può essere visto come una grande rete ethernet perché il broadcast domain fa utilizzo dei pacchetti inviati in broadcast che è inefficiente per le grandi reti.  
In questo caso internet viene diviso logicamente in tante piccole reti.   
Entra in gioco il distribution layer in questo caso, in collegamento con l’access layer utilizzando il protocollo IP. Gli switch moves ethernet (ricevono il traffico locale ma hanno bisogno del router per accedere a internet), router moves IP.

La differenza fra ethernet ed IP address: Ethernet ha un indirizzo fisico (MAC) non modificabile che ci permette di essere raggiungibile solo nella stessa rete. IP ha un indirizzo logico modificabile che permette l’essere raggiungibili nelle diverse reti.

La comunicazione fra gli host nella rete avviene tramite i livelli. Ogni livello offre un servizio al livello superiore e sfrutta quelli del servizio inferiore. Lo scambio dei messaggi è regolato dai protocolli. Ogni livello incapsula i propri dati per trasferirli al livello inferiore fornendo informazioni aggiuntive al pacchetto PDU. Spostandosi dal livello applicativo al livello fisico.

Due tipi di architettura a livelli: ISO/OSI composta da 7 livelli, e TCP/IP composta da 4 livelli (Application layer è composto da application layer, presentation layer, session layer).   
I pacchetti che attraversano la rete per raggiungere un host sono tutti indipendenti fra di loro e possono raggiungere l’host in differenti modi (path).

Ogni livello possiede un tipo di indirizzo: indirizzo internet ([www.xxx.it](http://www.xxx.it/)), numero di porta, ip address, mac address.

Le porte vanno da un range di 0-65535.  
La porta di destinazione determina quello che è il servizio richiesto. Generalmente le porte già assegnate sono 0-1023 e sono chiamate “well-known ports” e sono usate di server per le applicazioni internet come: SMPT, HTTP, IMAP. Le porte che vanno da 1024-49151 possono essere registrate tramite lo IANA, mentre le porte 49152-65535 sono porte effimere.

# IPv6

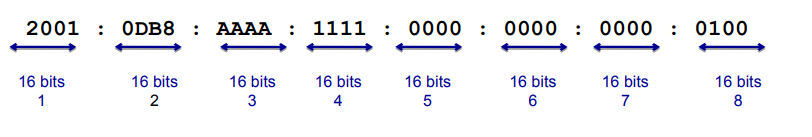
## Ipv6 addressing

IPv6 è il protocollo nato in sostituzione di IPv4 a causa dello scarso spazio di indirizzamento restante. È composto da 128 bit in esadecimale.

I benefici di IPv6 sono un maggior spazio di indirizzamento allunga l’indirizzo IP ad una lunghezza di 128 bit, la stateless autoconfiguration che fornisce la possibilità di autoassegnarsi un indirizzo routable, la possibilità di connettersi in modo peer to peer rimuovendo l’uso di indirizzi privati e NAT.

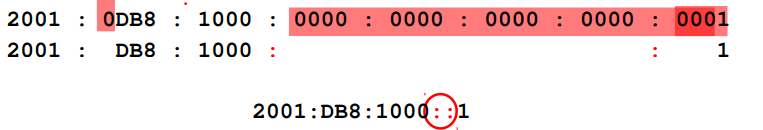
La notazione di IPv6 prevede 8 gruppi da 16 bit. Ogni 4 bit abbiamo un carattere esadecimale, quindi vi sono 4 caratteri esadecimali per ogni gruppo.

I valori esadecimali vanno da 0000 a FFFF

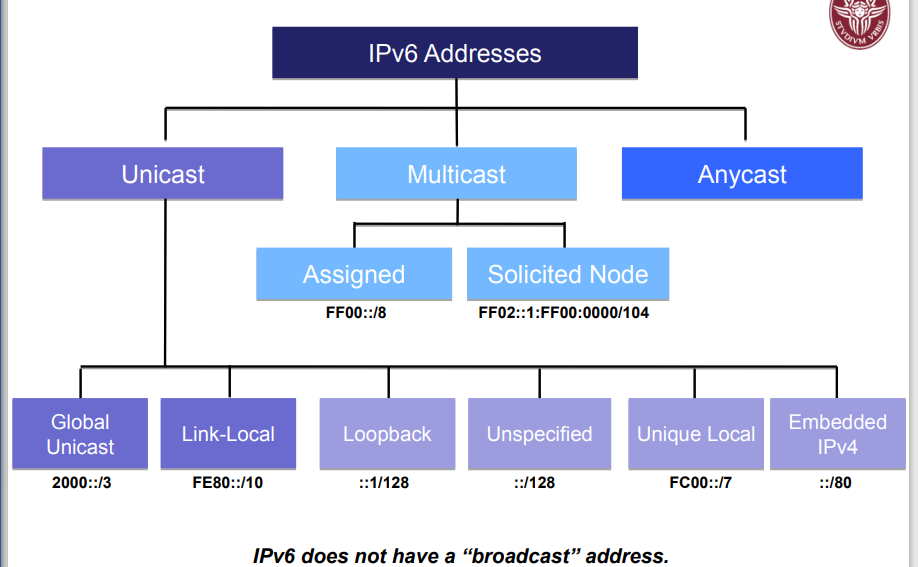


È possibile ridurre in lunghezza gli indirizzi tramite due regole.

* Se esistono 16 o più bit consecutivi di tutti 0 è possibile non scriverli e sostituirli con “::”
* Se estistono degli 0 iniziali di un gruppo è possibile rimuoverli



Esistono diversi tipi di indirizzi IPv6:



Non esiste più un indirizzo broadcast che viene sostituito dall’anycast che è un indirizzo che viene assegnato a diversi tipi di dispositivi pubblici che devono essere raggiungibili in internet. Questi dispositivi avranno gli stessi indirizzi. Il router cerca di raggiungere il dispositivo più vicino con quell’indirizzo.

L’indirizzo multicast permette l’invio di un pacchetto a più destinazioni allo stesso momento.

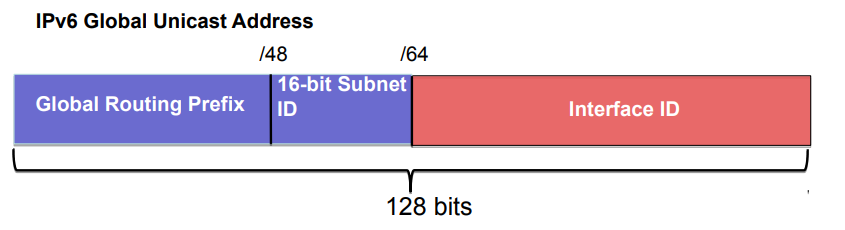
**Global Unicast Address**

Sono gli indirizzi pubblici routabile e unici.

Hanno un indirizzo del tipo 2000::/3 (fino a 3FFF::/3) dove la /3 è riferita ai bit binari.



Ogni indirizzo GUA è composto da:

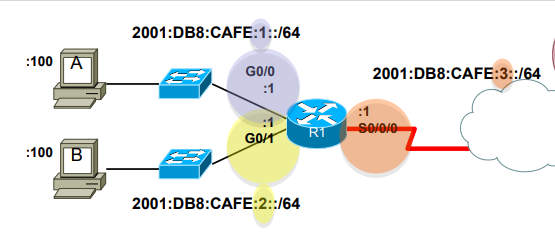


* /64 è la parte di rete dell’indirizzo composto a sua volta da:
  + /48 con il global routing prefix che sta a definire il network ID
  + /16 bit che rappresentano la subnet part
* I restanti 64 bit sono composti dall’interface ID, valore univoco per ogni dispositivo

Un Indirizzo IPv6 può essere facilmente descritto da un formato tipo: 3-1-4, dove i primi 3 gruppi di 16 bit appartengono al prefixID, un gruppo di 16 bit alla subnet ID e i restanti 4 gruppi di 16 bit all’interface ID



Quando un router deve fornire gli indirizzi a due lan, la parte di rete rimane la stessa mentre cambia la parte di subnet. Questa tecnica è chiamata prefix delegation

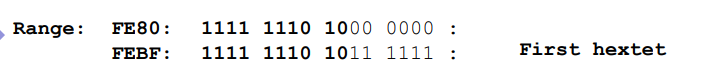


**Link-Local Unicast**

Un’indirizzo link local è l’equivalente dell’indirizzo privato IPv4.

Questo tipo di indirizzo è usato principalmente per comunicare con gli altri device all’interno dello stesso link, non essendo routable dall’esterno. È unico all’interno del link ed è inoltre necessario in quanto viene usato per ottenere un indirizzo di tipo pubblico.

Gli indirizzi vanno da: FE80 a FEBF prendendo in considerazione i primi 10 bit binari.



Il formato di un indirizzo local è composto da:

se automatico:

* FE80 (usato questo di solito), primi 10 bit
* 54 bit set to 0
* 64 bit interface ID:
  + EUI-64: usando i 48 bit del mac address, dividendoli a metà, inserendo FFFE tra loro, e flippando il 7 bit
  + random

static configuration, usata generalmente per i router.

Gli indirizzi link local sono utili alla generazione di un indirizzo global.  
All’interno del link invia degli ICMPv6 router solicitation verso un indirizzo multicast (cercando il router) per richiedere un indirizzo globale, al quale seguirà un ICMPv6 router advertisement come risposta. L’indirizzo link local del router è inoltre usato come default gateway per i vari device.

Questo indirizzo inoltre è usato dai router per parlare fra di loro scambiandosi routing protocol messaging

**Multicast address**

È un modo per inviare un singolo pacchetto a più destinazioni allo stesso momento, equivalente all’indirizzo 224.0.0.0/24 in IPv4. Presenta come prefisso FF00::/8

Esistono due tipi di multicast address:

* **assigned**: gruppi di indirizzi già assegnati, FF00::/8
* **solicited** **node**: usato nella neighbour discovery. È composto utilizzando gli ultimi 24 bit del link local address aggiunti a FF02:0:0:0:0:1:FF00::/104

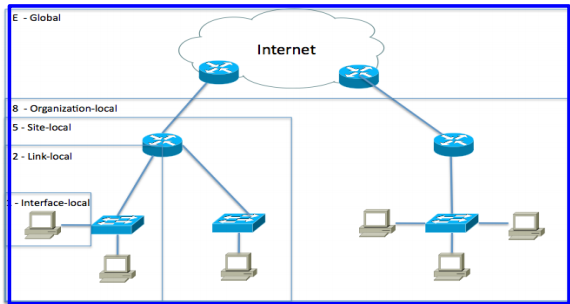
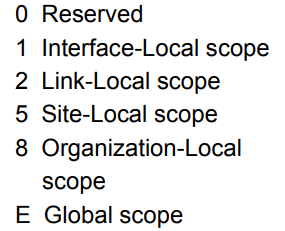


FF sono fissi, I due 0 successivi sono i 4 bit del flag e i 4 bit dello scope che variano.

Il **flag** definisce se un indirizzo è permanente o no. Può valore 0 se è permanente, assegnato da IANA o 1 non permanente assegnato dinamicamente.

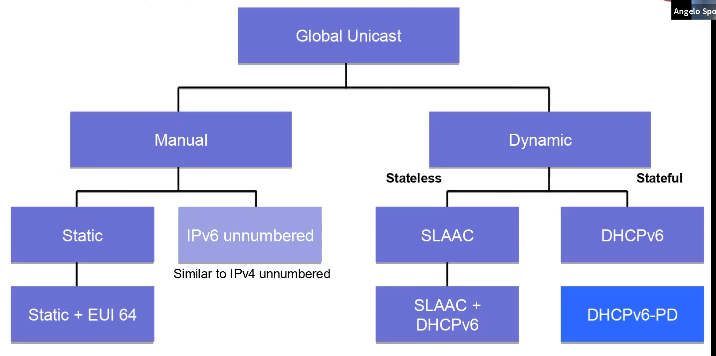
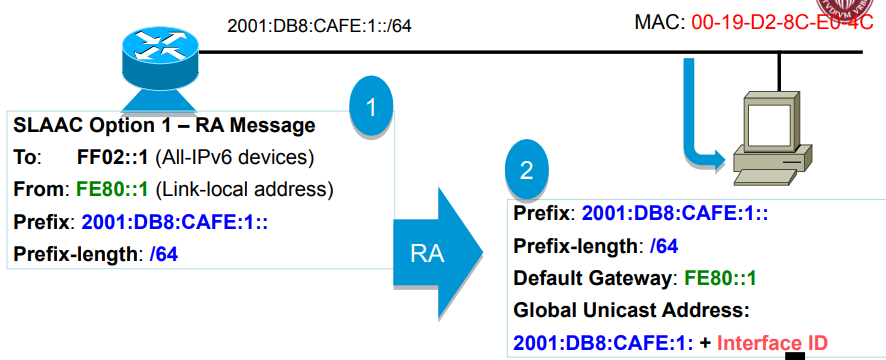
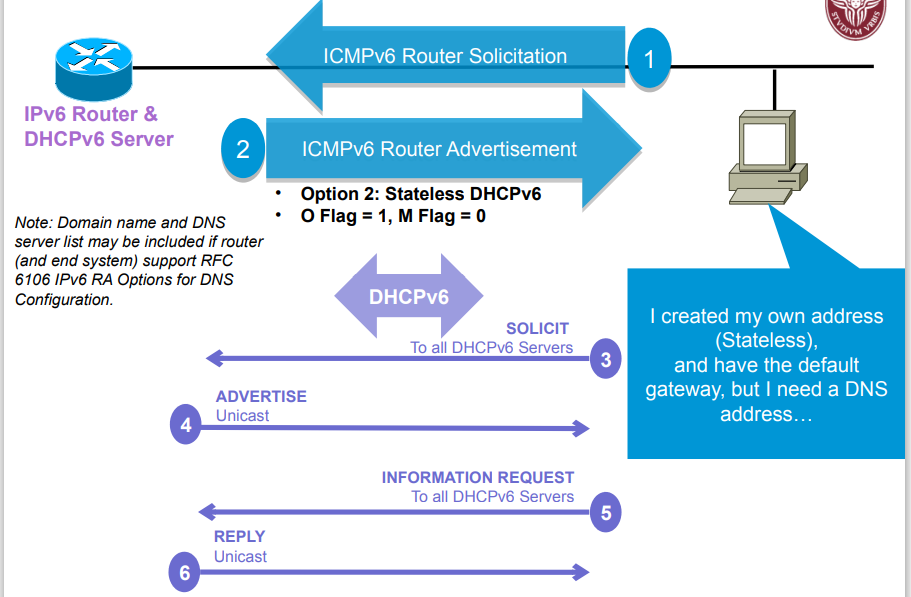
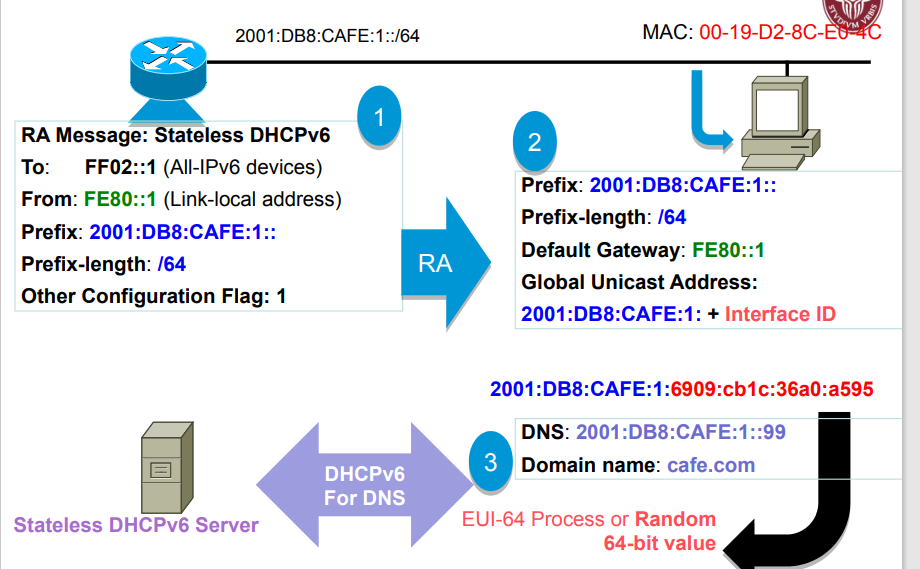
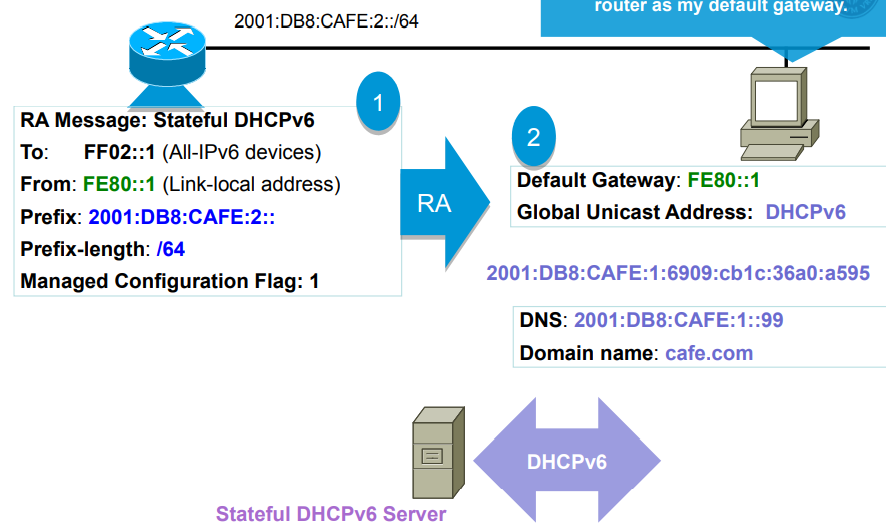
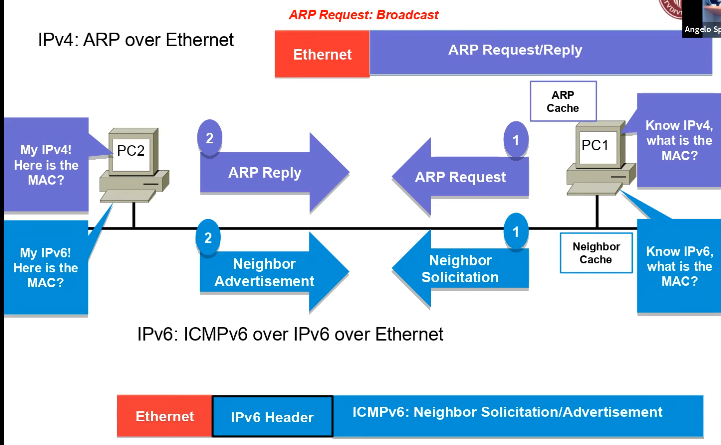
Lo **scope** definisce che tipo di impatto sulla rete ha quell’indirizzo, quando lontano può andare.

I valori sono:



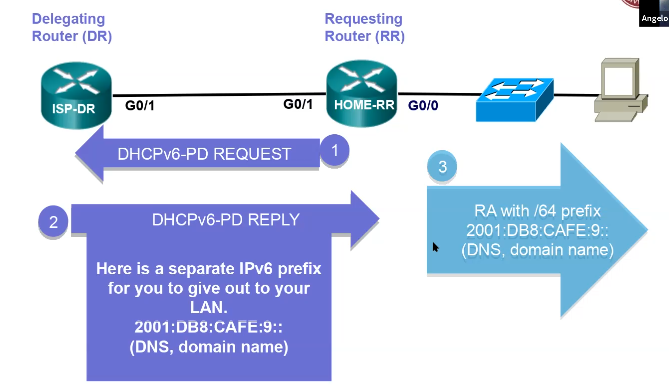
## Neighbour discover protocol

Il ICMPv6 neighbour discovery definisce 5 tipi diversi di pacchetti:

* **Router Solicitation / Advertisement**: viene usato per l’allocazione dinamica degli indirizzi. È un tipo di pacchetto scambiato tra router e device. Il router conosce gli indirizzi che sono allocabili nella rete e, all’inserimento di un device nella rete, fornisce le informazioni necessarie per l’autoconfigurazione dell’indirizzo. Informazioni come: chi è il default gateway, come autoassegnarsi un indirizzo..   
  Si parla quindi di stateless dato che, non usando DHCP, non si ha la piena conoscenza dello stato della rete e ci si autoassegna un indirizzo. In caso contrario si configura in maniera statica manualmente o tramite DHCP in maniera dinamica.  
    
    
  Viene inviato un messaggio di router solicitation verso un indirizzo multicast per tutti i router nella rete tramite un indirizzo link local. Il router risponderà con un advertisment e il tipo di configurazione possibile:
  + **SLAAC**: fornisce il prefix, il prefix-lenght e il default gateway  
      
      
    La parte di Interface ID viene generata con EUI-64 o in modo randomico.  
    Nel caso in cui viene usato il metodo randomico, può essere generato un valore simile ad un altro IP già presente nella LAN. In questo caso entra in gioco il Duplicate Address Detection (DAD) che previene questi problemi.
  + **SLAAC + Stateless DHCP6 for DNS address**: informazioni come prima ma inserisce anche le informazioni riguardante gli indirizzi DNS da un server DHCP6 Server.  
      
      
      
    La risposta dal router avrà dei flag in più che definiscono la presenza del DHCP server per ottenere informazioni lato DNS. Quindi il device invierà una DHCP solicitation verso il server e le conseguenti risposte.  
      
    
  + **Statefull DHCP6**: L’indirizzo IPv6 verrà fornito direttamente dal DHCP6 server. Il router fornirà semplicemente le informazioni riguardanti il default gateway dicendo tramite la “managed configuration flag=1” che si deve usare il server DHCP.  
      
    
* **Neighbour Solicitation / Advertisement**: viene usata per la risoluzione degli indirizzi MAC-IP tra device nella rete. In IPV4 questo compito era svolto tramite messaggi ARP inviando in broadcast l’arp request via ethernet. In IPv6 è stato sostituito con il Neighbour message utilizzando l’ICMP packet sempre via ethernet. Il messaggio viene inviato tramite un solicited node multicast address.  
    
  
* **Redirect**: come router-to-device message

## DHCP Prefix delefation

Nell’ottica IPv4, il router di casa richiede ad un ISP un indirizzo pubblico che poi tramite NAT verrà “riservato” ai vari host all’interno della rete tramite indirizzi privati.  
In IPv6 non esiste più il concetto di NAT, ogni indirizzo deve essere routabile. Quindi il router non chiederà all’ISP un solo indirizzo pubblico ma un prefix che può essere associato a tutti gli host nella rete per creare l’indirizzo pubblico ipv6.



# Link-local attacks

Il network eavesdropping è la tecnica di catturare i pacchetti che vengono trasmessi ad altri nodi in modo da leggerne il contenuto alla ricerca di informazioni sensibili. Viene fatto usando dei tools chiamati sniffer.

La modalità in cui viene fatta è totalmente passiva, cioè i pacchetti vengono catturati come vengono e passati ad altri livelli per poter essere analizzati.

Lo sniffing viene fatto sulle interfacce di rete in promiscuous mode, cioè viene accettato il pacchetto e inviato su.

Gli switch registrano l’associazione fra porta e MAC address nella tabella CAM (content Addressable memory) che possiede una grandezza limitata. La tabella viene riempita con un record ogni volta che un frame ethernet si muove attraverso lo switch inserendo il MAC address e la porta da cui l’ha ricevuto. Quando arriva un pacchetto che non ha un MAC conosciuto all’interno della CAM, questo viene inviato in broadcast su tutte le porte. Questa tecnica viene chiamata flooding.

La tabella CAM è soggetta ad attacchi CAM overflow. Consiste nell’inviare tanti frame con valori random come MAC address nella source. In questa maniera lo switch è costretto a registrare tutti i MAC address che riceve. Lo spazio però è limitato e quando viene raggiunto il limite massimo, lo switch viene costretto a mandare in flood continuo ogni singolo frame comportandosi come un hub. In questa maniera i frame possono essere intercettati da un aggressore che sta sniffando il traffico di rete. Nel caso in cui lo switch crasha l’aggressore ha creato un “denial of service”.

Un ARP request è la request associata all’algoritmo di ARP per chiedere chi, nella rete, possiede un determinato ARP. Viene inviato un frame con questa request verso tutti gli host della rete e viene fatto un confronto tra l’IP del pacchetto IP ed il proprio. Se combacia, viene inviata una response ARP.

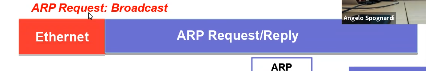
La ARP table contiene le associazioni fra IP-MAC e viene acceduta prima di inviare in broadcast il frame ethernet. Viene riempita col tempo e i MAC address più vecchi vengono rimossi dopo la fine di un timeout.

All’interno di una rete accade anche il così detto “Gratuitos arp response” dove un host annuncia alla lan il proprio IP address in modo da evitare duplicati. Dato che non vi è protezione applicata all’ARP, l’host che annuncia il suo IP lo potrebbe fare in modo ripetitivo. Quando un host possiede già nell’arp table quell’IP lo ignora. Allo scadere del tempo, questo Ip può essere aggiunto insieme al MAC nella tabella ARP.

L’attacco Main in The middle fa si che un hacker posto tra client e server finge di essere il server, intercetta il traffico lo ridireziona verso il server che invia la risposta al client. In mezzo a tutti questi passaggi sniffa tutto il traffico. La ARP table del client viene riempita con un valore sbagliato dato che avrà l’IP del server con il MAC dell’attacker.

La versione IPv6 di ARP è la neighbour discovery. La differenza sta nell’uso di ICMPv6 su ethernet mentre ipv4 usa ARP message su ethernet. L’arp request viene inviata in broadcast mentre su ipv6 viene inviata ad un solicited node multicast

Il pacchetto su ipv4 è composta da:



Mentre per ipv6



Il duplicate address detection viene usato per garantire l’unicità di un indirizzo ipv6 all’interno di un link. Viene fatto atrtaverso la neighour solicitation con il suo indirizzo unicast e se non riceve risposta significa che l’indirizzo è unico nel link.

# Firewall

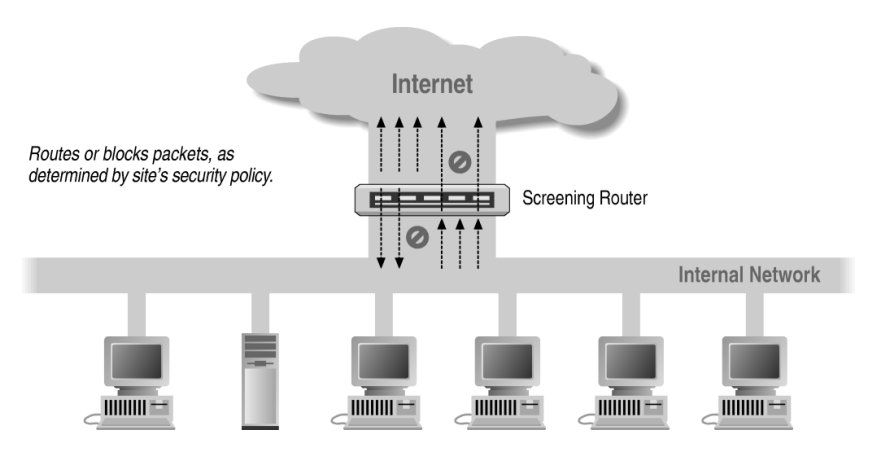
Un firewall è un dispositivo che oltre a fungere da router per i pacchetti decide quali pacchetti possono o meno passare tra le varie network, restringendo gli accessi dall’esterno e attuando una prevenzione degli attacchi.

L’architettura di un firewall è pensata per

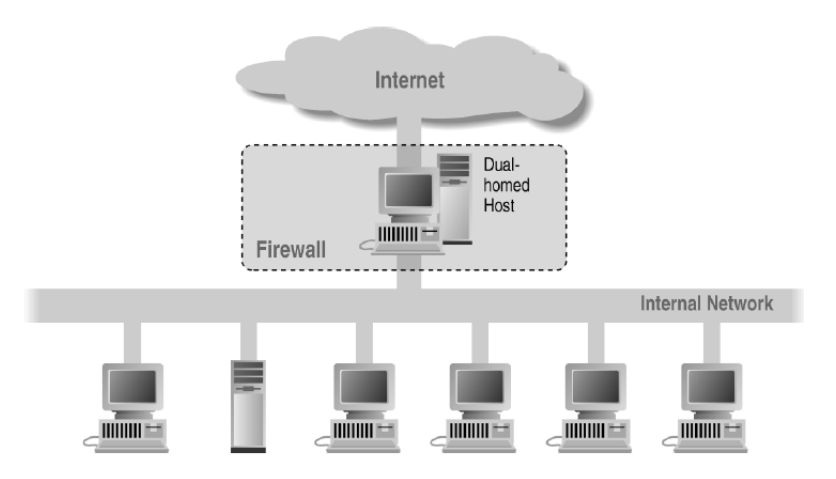
* fornire il numero minimo di privilegi.
* Fornire un ampia gamma di strumenti di difesa su più livelli
* Avere un punto in cui tutto il traffico viene analizzato

Un semplice tipo di firewall è quello **host based packet**, visto anche come un personal firewall sviluppato da qualche società per un sistema operativo. Questo firewall conosce il tipo di traffico che i vari software scambiano e creano delle regole ad hoc.

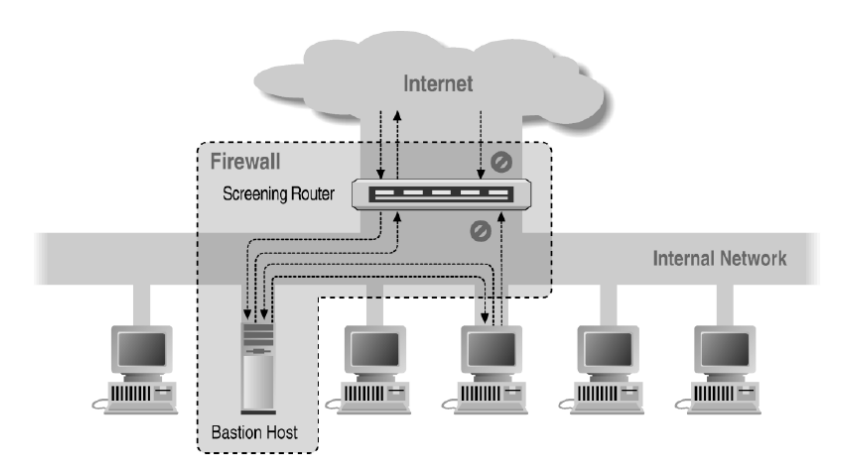
**Screening router (ACL-based)**: è un tipo di implementazione del firewall che lavora con un approccio veramente low level. Possiede una list di ACL right per i diversi pacchetti. Le decisioni che questo router prende vengono fatte considerando gli ip che ci sono nei pacchetti. Infatti, si parla di network access control list. Ogni pacchetto viene gestito in maniera indipendente dall’altro, in maniera stateless.

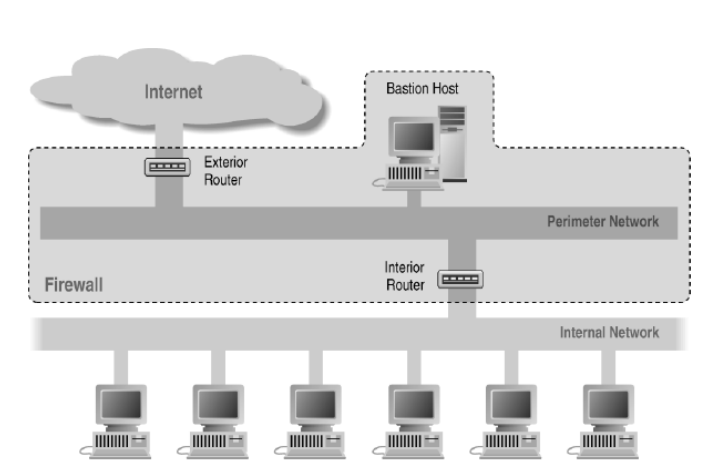


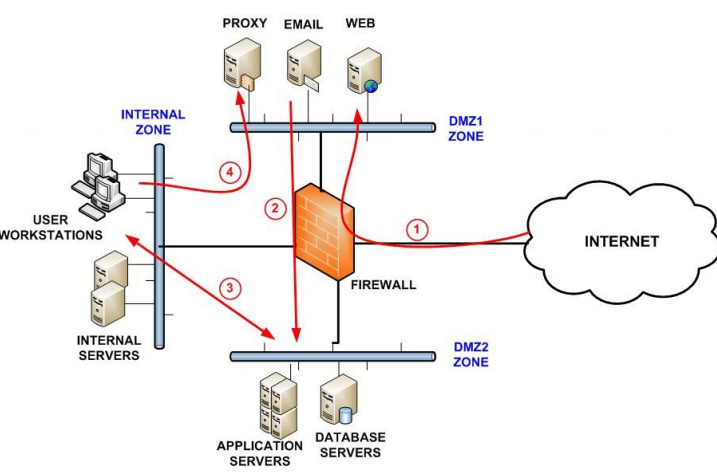
**Dual-homed host:** il firewall controlla sia gli IP sia il contenuto dei pacchetti. Vengono chiamati Bastion host, visti come metafora di un castello che protegge l’interno. Attua il processo di “hardening” cioè di ridurre o rimuovere quelle che sono le vulnerabilità in un sistema permettendo l’accesso ad un numero limitato di utenti, spegnendo servizi non più utilizzati o dannosi o impostando delle configurazioni specifiche. È spesso usato come application proxy gateway.

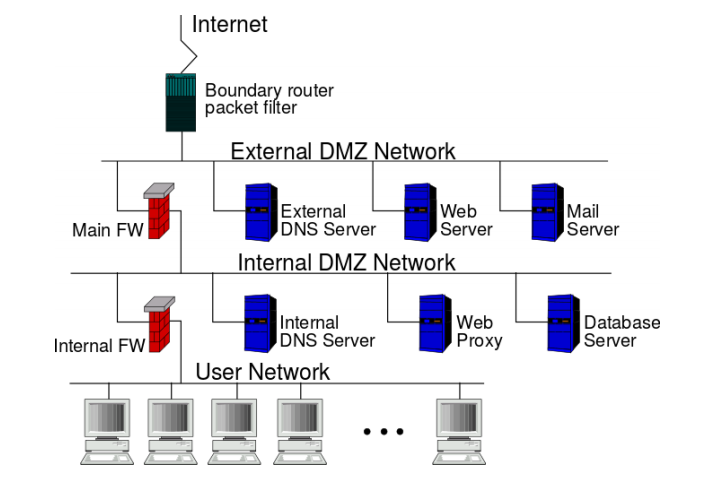


All’interno della rete può esserci una zona chiamata DMZ che possiede degli host che espongo servizi che vogliamo rendere disponibili all’esterno. In questa maniera chi compromette la DMZ non compromette automaticamente la nostra rete interna. Un modo per assicurare la DMZ non è quella di utilizzare diversi firewall ma è quella di fornire la “multiple overlapping” cioè più diversità possibile di sistemi di sicurezza come autenticazione, firewall e configurazioni specifiche.



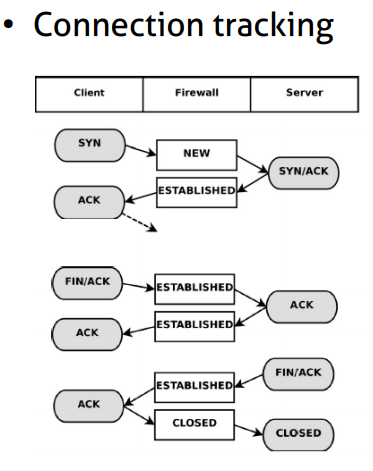






I firewall possono essere di due tipi:

* **Packet filter (stateless)**: droppano i pacchetti in base alla source o alla destination, al numero di porta o ai flag. Non guardano il contesto ma solo il contenut, quindi non tengono traccia di uno storico. Operano su entrambe le interfacce.  
  Lavora a livello di rete (indirizzi, payload), livello di trasporto (porte, flag), livello applicativo.  
  Il problema di questo tipo di soluzione sta nel fatto che i parametri di decisione sono pochi o troppo generali e il payload TCP non viene per niente controllato.
* **Stateful packet inspection**: i firewall tengono traccia delle connessioni stabilite dai pacchetti, quindi del loro stato e contesto. Oltre ad attuare il drop sui singoli pacchetti, controllano se esiste una connessione tra pacchetti e se i pacchetti vi appartengono.

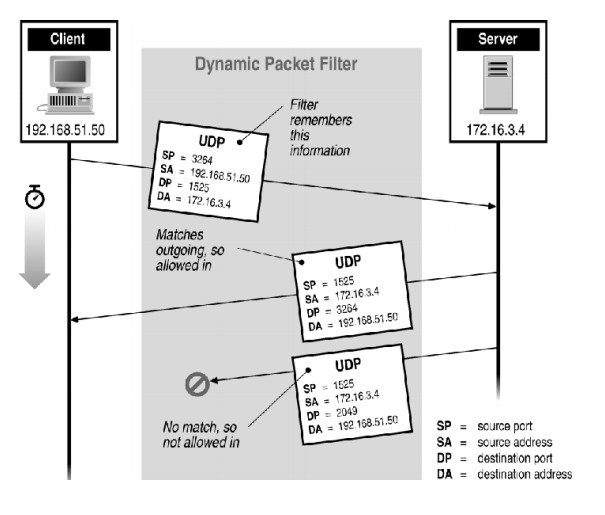


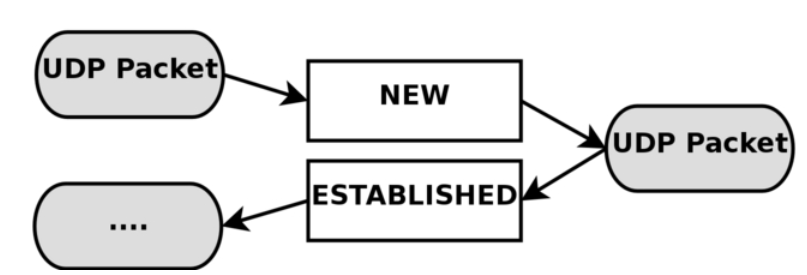
All’interno di iptables i pacchetti TCP possono essere quindi relazionati ad una connessione in quattro differenti stati:

* **NEW**: indica che è il primo pacchetto che viene tracciato e che da inizio ad una connessione. Se il pacchetto presenta il flag SYN significa che è il primo pacchetto dello scambio TCP. Quando il Server risponderà con un pacchetto ACK di risposta al SYN verrà stabilità la connessione e si passerà in ESTABLISHED con un pacchetto di ACK da parte del client per confermare l’avvenuta connessione.
* **ESTABLISHED**: è lo stato che si stabilisce quando il firewall vede uno scambio di pacchetti in entrambe le direzioni a seguito di una connessione generata (SYN – SYN/ACK)
* **CLOSED**: lo scambio TCP va verso la chiusura della connessione se dal client viene inviato un pacchetto FIN/ACK verso il server. Ci troviamo ancora nello stato established in quanto questi pacchetti fanno sempre riferimento alla connessione stabilita. Quando il server risponderà anche lui con un FIN/ACK, il client avrà la certezza che il server ha compreso la chiusura della connessione, invia un ACK e si passa in CLOSED

Questo discorso non vale solo per TCP ma anche per UDP nonostante sia un protocollo stateless. La ricezione di due datagrammi UDP in sequenza non ci dice nulla riguardo l’ordine, tuttavia è possibile settare uno stato della connessione livello kernel.

Il client invia un datagramma UDP contenente indirizzo e porta mittente e indirizzo e porta destinataria creando una nuova connessione. Il firewall memorizza queste informazioni per un certo periodo di tempo e, se entro quell’arco temporale, il client riceve una risposta dal server con un datagramma UDP con le informazioni al contrario, viene considerata la connessione come ESTABLISHED e vale per tutti i pacchetti che rispettano quelle informazioni all’interno del timer.





Oltre a questi stati per TCP abbiamo altri stati per quanto riguarda iptables che sono **RELATED**. **INVALID** e **UNTRACKED**. La RELATED esiste solo se è presente uno stato ESTABLISHED. Da questa connessione stabilità viene generata una nuova connessione che viene considerata RELATED.

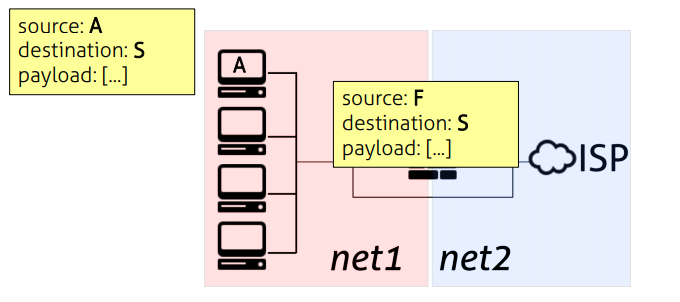
# NAT

Nat sta per Network address translation ed è un servizio che permette di far connettere a internet una LAN usando indirizzi privati non routabili, di far si che i servizi all’interno di una LAN siano raggiungibili dall’esterno e di non esporre gli host all’interno della lan allo scan port ottenendo maggiore sicurezza.

In questa maniera è come se il servizio che viene fornito o richiesto dalla LAN venga in realtà richiesto o fornito dal router/firewall che maschera col proprio indirizzo pubblico. E che i device presenti nella LAN non siano indirizzabili in modo esplicito dall’esterno.

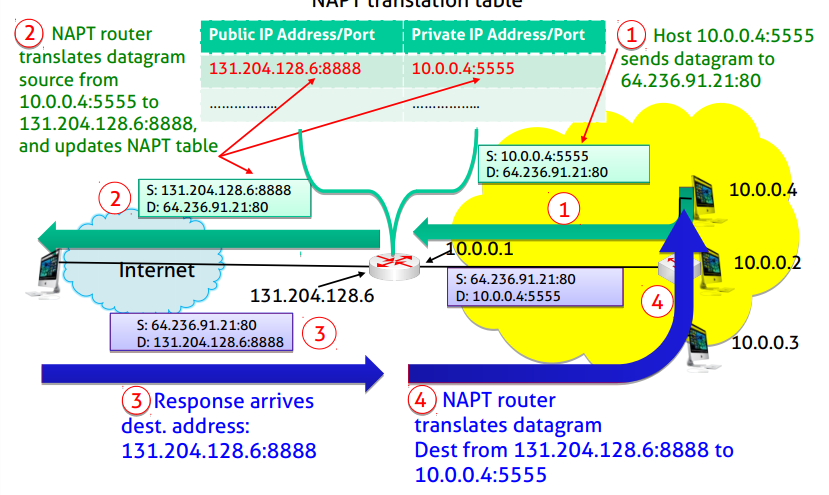
Gli host all’interno della LAN nel fare questo utilizzo l’indirizzo pubblico del NAT mascherando il proprio indirizzo privato. In questa maniera tutti i cambiamenti che avvengano all’interno della LAN non sono visibili all’esterno ed inoltre anche nell’eventualità di un cambio di ISP non è necessario cambiare gli indirizzi degli host nelle reti private.

In base al tipo di indirizzo che viene mascherato durante questo processo si parla di:

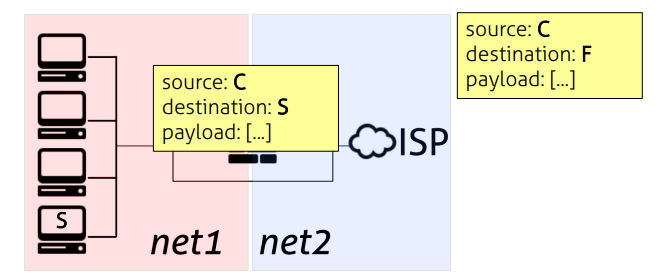
**Source Nat** se viene mascherato l’indirizzo IP della source nell’header di un pacchetto diretto dalla LAN verso la WAN. In questo caso il pacchetto viene tradotto con l’indirizzo pubblico del NAT.  


Il Basic Nat è quindi la traduzione a livello di IP di un indirizzo privato verso un indirizzo pubblico. Ma dato che all’interno di una LAN possono esserci tanti host che vengono tradotti in un unico indirizzo pubblico, vi è la necessità di definire un modo per discernere i diversi host. Questa cosa viene fatta tramite il NAPT cioè un NAT in cui i vari host vengono definiti dalle diverse source porte da cui provengono i pacchetti. Non viene solo tradotto l’indirizzo IP ma anche la porta privata in pubblica fornendo quindi una diversa mappatura tra interno ed esterno.

All’interno della NAT table ci sarà una coppia <indirizzo privato:porta> -> <indirizzo pubblico:porta> che servirà per i successivi pacchetti in fase di traduzione.



**Destination Nat** se viene mascherato l’indirizzo IP di destinazione di un pacchetto proveniente dalla WAN verso la LAN. Il pacchetto all’inizio conterrà l’indirizzo pubblico del firewall per poi essere tradotto in un indirizzo privato tramite la NAT table.

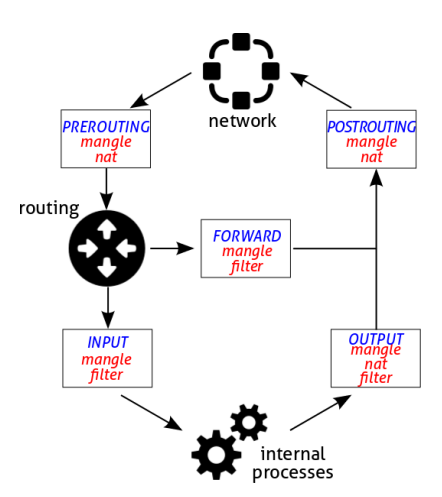


Il destination NAT è anche conosciuto come Port Forwarding poiché in base al tipo di porta su cui viene richiesto il servizio dalla WAN, la nat table indirizza il pacchetto all’host stabilito nella lan.

In Iptables la tabella NAT viene quindi usata nel firewall per tradurre gli indirizzi.

Presenta 4 tipi di azioni:

* **DNAT:** traduce il destination IP dei pacchetti in entrata, usando la catena di PREROUTING
* **SNAT:** traduce il source IP dei pacchetti in uscita, usando la catena di POSTROUTING
* **MASQUERADE** (usato nella stessa maniera in cui viene usato il SNAT ma con la differenza che l’indirizzo che viene scelto in fase di traduzione è scelto dinamicamente in base all’indirizzo dell’interfaccia da cui uscirà il pacchetto. SNAT è statico)
* **REDIRECT** (redirect verso il firewall stesso).



Solo il primo pacchetto verrà sottoposto alla tabella nat, quelli successivi seguiranno la decisione imposta al primo.

Le catene della tabella NAT sono: prerouting, postrouting, output.

# VPN

Sito to site network: due o più site che vogliono essere collegati da una rete privata. Dal punto di vista dell’utente i due site è come se fossero collegati direttamente. Ma invece a livello tecnico è un livello addizionato sopra la rete, tutti i meccanismi di routing sono effettuati dai router dei due site. Quando un pacchetto deve arrivare nell’altro site, segue semplicemente i classici path di routing conosciuti dai router passando per internet ma vi è la garanzia che i pacchetti sono coperti da privacy fornita dalla vpn.

Host to site network: un host che vuole essere collegato ad una rete privata (site) usando internet.

Host to host network: point to point VPN, connessione tra un host dentro un site con un host al di fuori della rete privata.

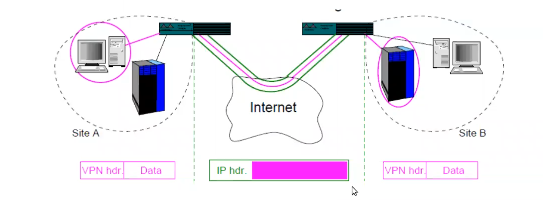
L’encryption può essere applicato ai diversi layer della rete:

* Physical: protezione dei dati su alcuni cavi speciali che connettono le reti. Tutti dati encripted
* Data link: protezione del singolo link. Tutti i dati tranne il livello fisico vengono encripted
* IP level: Protezione dei dati application e transport ma non di ip e data link. È transparent dal punto di vista utente e software
* Transport: Protegge i dati a livello applicativo ma lascia l’header del transport layer liberi
* Application: garantisce sicurezza solo fra singole application

## VPN Tunneling

Il concetto di tunnel prevede l’encapsulazione di un payload di un pacchetto di un altro protocollo all’interno di un altro pacchetto. Il tunneling offre quindi un metodo base per fornire il servizio di VPN. Bisogna solo decidere il tipo di architettura che gestirà i dati e su che livello applicare il tunnel-

I due endpoint prendono i dati, creano un nuovo pacchetto che va su internet. Quando arriva nella nuova destinazione, viene estratto e usato. Con questo tipo di soluzione possiamo creare una site to site. Il tunneling viene trasformato in un secure tunneling perché oltre a encapsulare la pdu, questa viene anche protetta con l’encryption. Ciò che viene protetto sono tutti i dati uscenti dalla macchina, mentre vengono lasciati liberi gli header creati dal router. Dato che l’encription e decription viene svolta solo dai router ai confini della rete, sono rispettate le proprietà di sicurezza tipiche della VPN. La vera sfida per la VPN nel criptare i dati sta nello scambio sicuro e protetto delle chiavi.

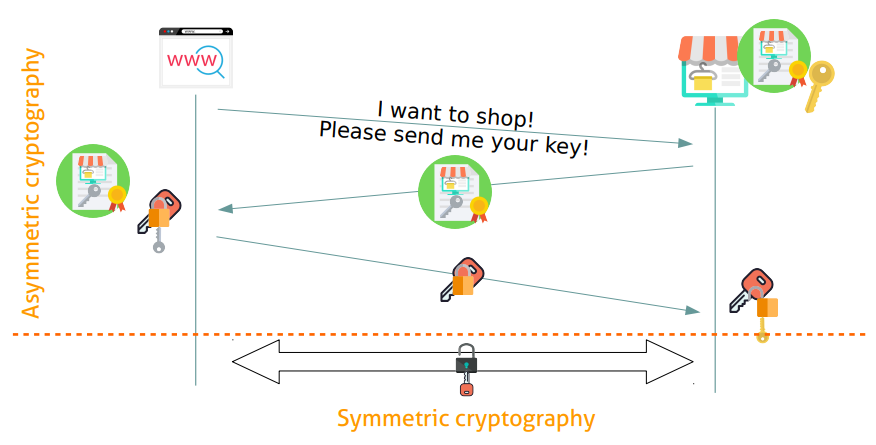


## SSL

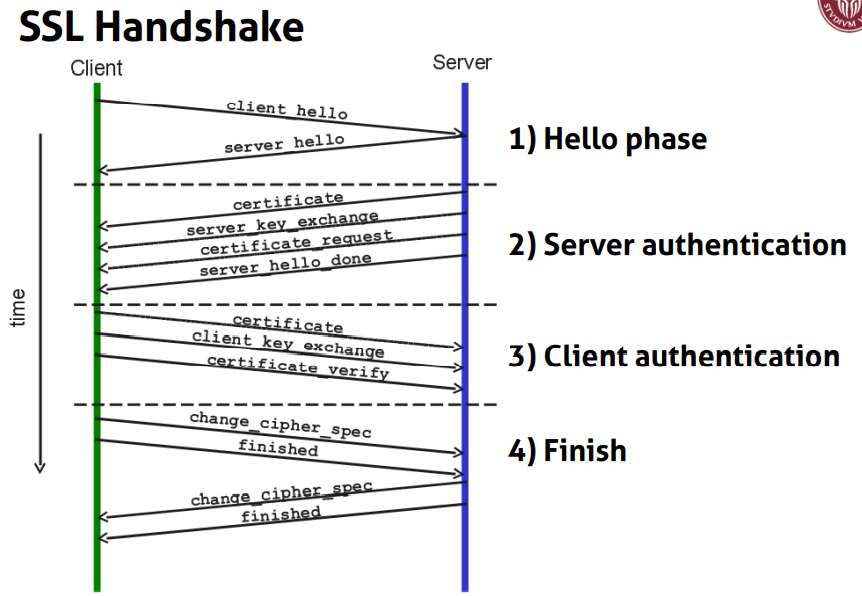
SSL (TLS) è un protocollo crittografico che applica sicurezza al livello di trasporto e fornisce protezione tra le sessioni client-server. I due endpoint creano una sessione sicura sul livello di trasporto in modo che venga gestita come un tunnel per encapsulare i dati se applicato sui router o proxy.  
SSL funziona così con HTTPS: Stabilita la connessione tramite TCP, il client richiede una key al server. Il server risponde con una public key e una signature di un certificato che associa la key al server. Questo perché l’autenticazione è unilaterale, il client rimane anonimo ed è il solo server ad autenticarsi verso il client tramite il controllo della firma digitale dei certificati tramite la public key.

Ricevuto il certificato viene generata un’altra key dal client che verrà criptata con la public key del certificato per essere usata come key della sessione. E viene inviata al server che sarà capace di decriptarla grazie alla chiave pubblica del certificato. Effettuati questi passaggi, la connessione è crittografata dalla chiave di sessione. TLS è una crittografia simmetrica.

Tutto lo scambio iniziale per negoziare una key è una crittografia di tipo asimmetrica. Dopo il tutto avviene tramite una crittografia simmetrica.



SSL è il protocollo che aggiungere informazioni tra livello TCP e livello Application inserendosi come livello di mezzo fra questi. I parametri dell’SSL sono l’handshake (usato per autenticare il server e concordare sulla chiave e sull’algoritmo), change cipher spec (usato per selezionare la chiave concordata e l’algoritmo), alert che fanno parte del record protocol il quale offre servizi di integrity e encryption per le applicazioni



L’architettura VPN SSL è composta da due modelli:

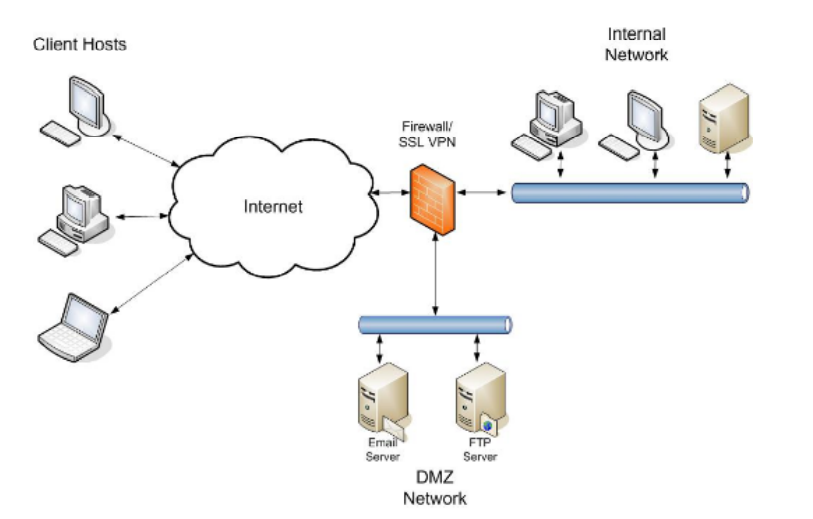
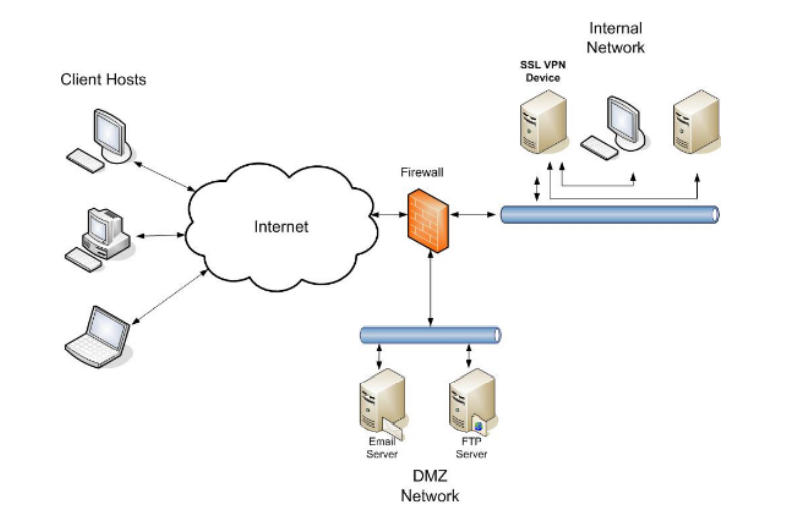
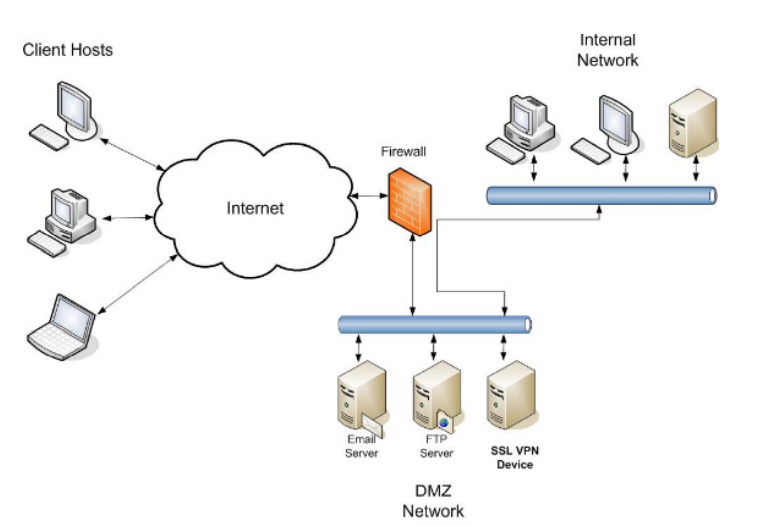
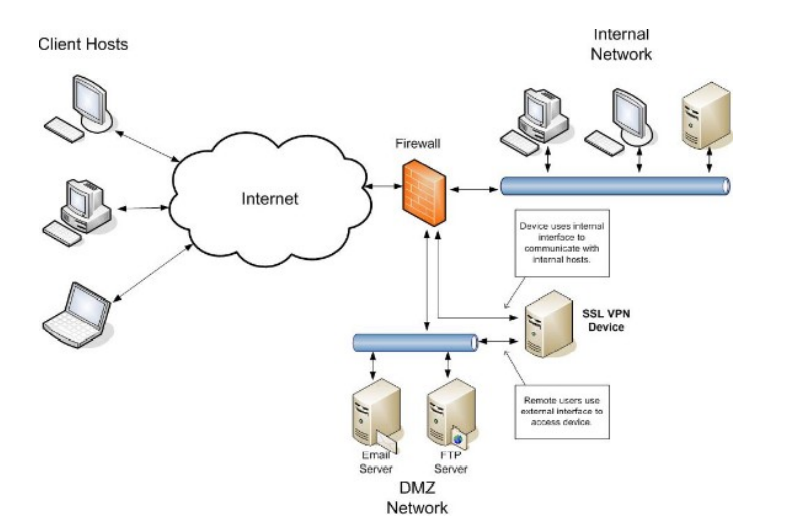
* SSL portal VPN: connessione ad una VPN gateway da un Web Browser e la vpn viene usata come gateway per accedere a dei dati direttamente dal web browser
* SSL tunnel VPN: accesso ad una rete tramite una VPN gateway tramite un web browser.

Una delle funzionalità principali di questa ssl vpn è il proxy che fa la route dei pacchetti al posto della original source; Network extension che fornisce accesso alla rete per alcuni utenti tramite tunnel.

I servizi inclusi sono l’authentication tramite certificato, security token; encryption; access control, endpoint security controls, intrusion prevention.

## VPN Device Placement

Il posizionamento della VPN in diverse parti della rete porta a più o meno vantaggi:

* **Firewall with an SSL VPN**: gestione facile dato che abbiamo un host che svolge funzione da firewall e vpn, quindi il traffico che passa tra interno ed esterno viene gestito dal firewall stesso. Gli svantaggi sono nelle modalità in cui è sviluppato il firewall ed inoltre si sovraccarica il firewall di compiti che non gli competono del tutto come quello della VPN. Inoltre il firewall deve fornire la porta 443 TCP per fornire il servizio di VPN  
  
* **SSL VPN in internal network:** il firewall deve permettere la connessione al device VPN che poi dovrà reindirizzare i pacchetti all’interno della rete. Il firewall ha solo una regola che permette l’accesso alla vpn, in questo modo è concesso solo il traffico verso la VPN. Inoltre, essendo la VPN dietro il firewall, viene protetta da qualsiasi attacco dalla DMZ.  
  Gli svantaggi sono nel fatto che una volta che il traffico ha passato il firewall, non vi è più nessuna protezione, in questa maniera se la VPN è compromessa lo sarà anche l’internal network
* **SSL VPN in DMZ:** Il traffico passa il firewall arriva nella DMZ nella VPN che farà il reindirizzamento del traffico verso l’internal network passando di nuovo tramite firewall. In questa maniera l’internal network sarà protetta da una compromissione della VPN dato che tutto ripassa per il firewall. È possibile avere un intrusion detection system nella DMZ per analizzare il traffico verso l’internal.  
  la complessità sta nel fatto che bisogna gestire numerose porte nel firewall. Inoltre il traffico dalla DMZ viene decriptato e reinviato verso l’internal network.
* **Dual Interface VPN device in DMZ:** è il più sicuro. Il traffico diretto alla VPN è staccato dalla DMZ utilizzando due interfacce diverse, una diretta per l’internal network ed una per la DMZ. Il traffico che viene decriptato non è così accessibile alla DMZ. Il firewall deve gestire una interfaccia specifica della VPN in modo semplice.  
  Anche qui però devono essere gestite diverse porte nel firewall oltre al fatto che vi è una complessità di routing maggiore 

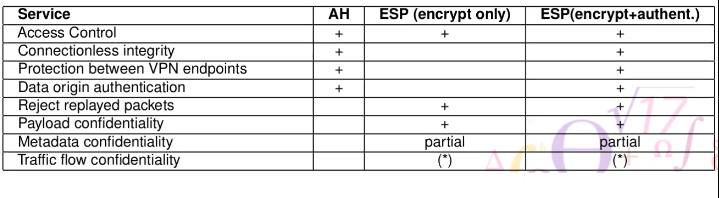
## IPSec

Protocollo di sicurezza applicato al livello IP. Ogni pacchetto viene gestito in maniera indipendente l’uno dall’altro.

Fornisce l’autenticazione sui pacchetti non permettendo la falsificazione degli indirizzi senza il ricevitore se ne accorga. Fornisce integrità dei dati quindi ogni modifica fatta al datagramma ip viene notata. Fornisce confidenzialità non permettendo l’intercettazione del contenuto del datagramma e protegge i pacchetti con delle security policy.

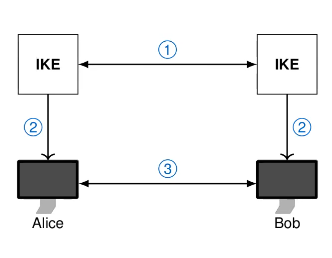
Le funzionalità che si ottengono con ipsec sono:

* Authentication header (**AH**): data integrity e authentication
* Encapsulated security payload (**ESP**): encryption e (opzionale) authentication
* Internet key Exchange (**IKE**): key management



È composto da due fasi:

1. Setup: autenticazione, scambio di chiavi e negoziazione degli algoritmi di criptazione. Può essere fatto in modo manuale o in modo automatizzato (IKE, ISAKMP)
2. Use: Utilizzo del canale sicuro creato usando AH o ESP



L’architettura di IPSec è composta da: SA, SAD, SP e SPD.

La security association (SA) è il canale che vi è tra due host che descrive il modo in cui i pacchetti devono essere processati. Ogni SA è monodirezionale e tutti i parametri che la compongono devono essere negoziati tramite IKE prima che la comunicazione avvenga. I parametri che compongono la SA sono il protocollo da utilizzare (AH, ESP), l’indirizzo di destinazione (unicast IP) e l’indice SPI di 32 bit, le tecniche di cifratura e le relative chiavi. Sono quindi contenute tutte le specifiche da attuare a seguito della policy.

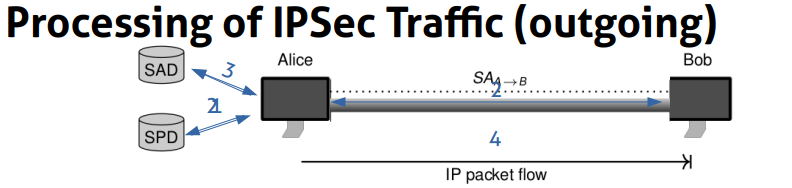
Le security Association sono contenute nel database SAD tramite il security parameter index (SPI).

Oltre alla associazione tra due host vi è il security policy, cioè l’insieme di regole che stabiliscono le modalità di scambio e protezione dei dati. Anche esse sono contenute in un database SPD.

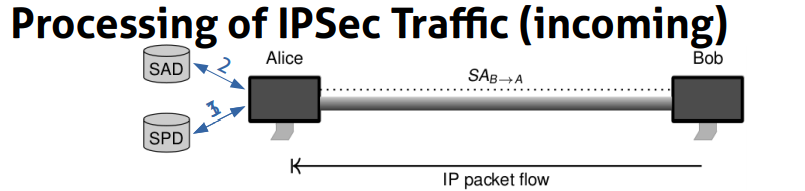
Quando si vuole proteggere uno scambio si cerca l’IP specifico all’interno del database tramite SPI per controllare il tipo di SA: il protocollo (AH, ESP), la modalità (transport, tunnel) e altri parametri insieme alle azioni da intraprendere:

* Discard: rifiuta invio e ricezione di quel pacchetto
* Bypass: non gestire con IPSec
* Secure: gestire con IPSec. Viene processato con “protocol/mode/src-dst/level”

La security association (SA) insieme alla Security Policy (SP) rendono possibile la totale implementazione dello scambio sicuro dei dati.



In caso di pacchetto in uscita viene controllato l’SPD per vedere riferito a quell’IP che tipo di regola vi sia (discard, bypass, secure). In caso debba essere messo in sicurezza con IPSec si cerca l’SA da applicare. Se non è già stabilita viene creata tramite IKE. Viene usata cercandola nell’SAD e vengono quindi applicate le trasformazioni di sicurezza descritte nell’SA per poi inviare il pacchetto.

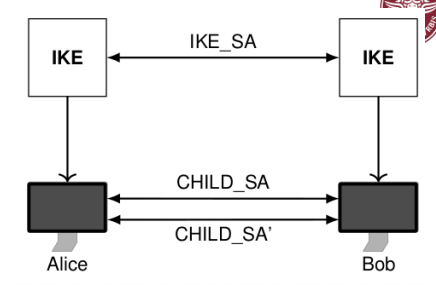


In caso di pacchetti in entrata, se il pacchetto contiene un header IPSec viene cercato nel SPD le regole di processamento del pacchetto. Se il pacchetto deve essere gestito viene estratto l’SPI dall’header, viene cercato l’SA nella SAD e viene svolto il processo appropriato. In caso l’SA non esista relazionato al SPI, il pacchetto viene scartatato. Se è tutto ok viene mandato ai livelli superiori.

### IKE

Protocollo per lo scambio delle chiavi e per l’autenticazione atto a stabilire gli SA tra due host. Si compone di fasi:

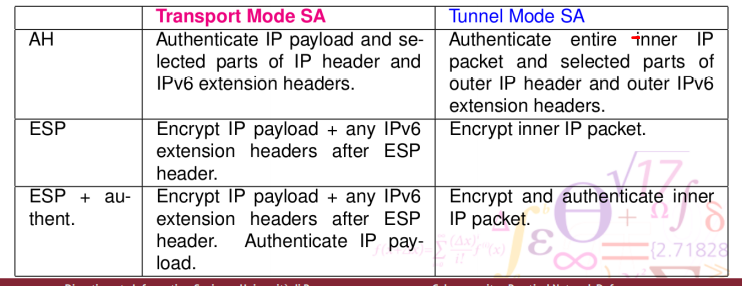
1. IKE\_SA\_INIT per inizializzare la negoziazione dei parametri di sicurezza per l’SA, inviando i Diffie\_hellman value per accordarsi sulla chiave. Vengono scelti gli algoritmi di criptazione e di integrity.
2. IKE\_AUTH per autenticare i messaggi precedenti, scambiare identità tra gli host e creare la CHILD\_SA (un set di SA usate per ah o ESP, è un sinonimo di SA)

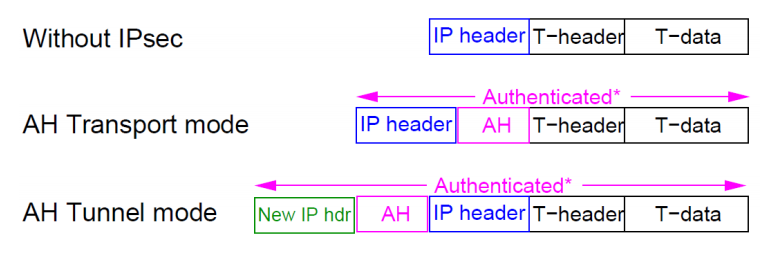


IKE\_SA è una SA che si crea dopo l’INIT. Viene usata per criptare e proteggere gli scambi di messaggi successivi in modo da creare la CHILD\_SA vera e propria che conterrà i dettagli di protezione con AH e ESP.

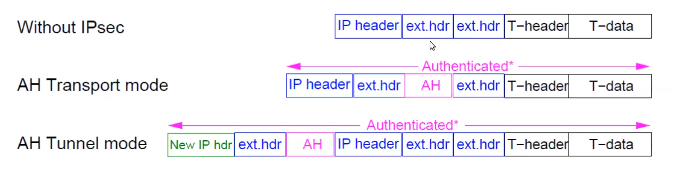
IPSec offre due modalità:

* **Transport Mode:** fornisce protezione per un pacchetto del livello di trasporto che si trova all’interno di un pacchetto IP come payload
* **Tunnel Mode:** fornisce protezione ad un pacchetto IP contenuto all’interno di un altro pacchetto IP come payload. In tunnel mode uno dei due peer che scambiano i dati non è il destinatario reale del pacchetto ma svolge la sola funzione di decriptazione per qualcun altro.



**Authentication IPv4:** l’authentication header che permette l’autenticazione del pacchetto interno viene inserito dopo IP header come ulteriore payload dell’IP header in trasport mode.   
Nel tunnel Mode viene generato un nuovo IP header e l’ah verrà inserito successivo a questo ma prima dell’IP header del pacchetto da autenticare.

**Authentication Ipv6:**



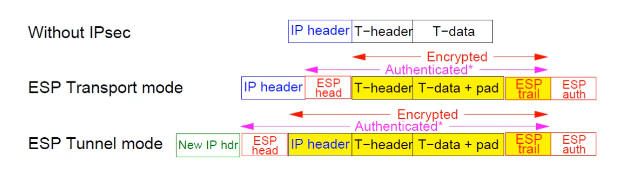
**ESP IPv4:** L’ESP header viene inserito dopo l’IP header e prima del Transport Header in Transport mode.  
Viene inserito subito dopo il nuovo IP header e prima del vero pacchetto IP da criptare.

Il campo padding viene inserito alla fine del payload del livello di trasporto per fornire protezione del traffico.

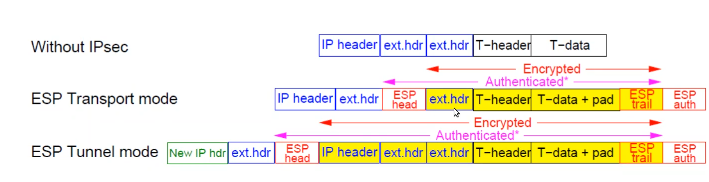
L’ESP trailer viene aggiunto alla fine del payload del transport layer insieme a un authentication ESP opzionale.

Viene criptato solo il payload mentre autenticato il payload e l’header esp in transport mode.

Nel tunnel mode viene criptato anche l’ip header del pacchetto IP e autenticato anche l’esp header



**ESP IPv6:**



In caso di combinazione di ESP e AH si può avere:

1. ESP con authentication, esp ai dati e poi viene aggiunti un campo ah
2. Una catena di encription: una coppia di SA, prima con esp e poi con AH

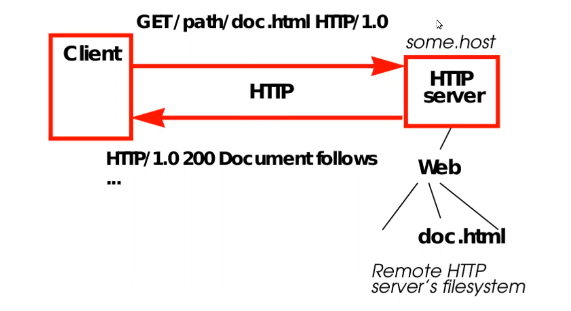
# Proxy

Il proxy è nato come server che forniva l’accesso ad internet ad utenti vicino a delle subnet che potevano accedere ad internet attraverso un firewall.

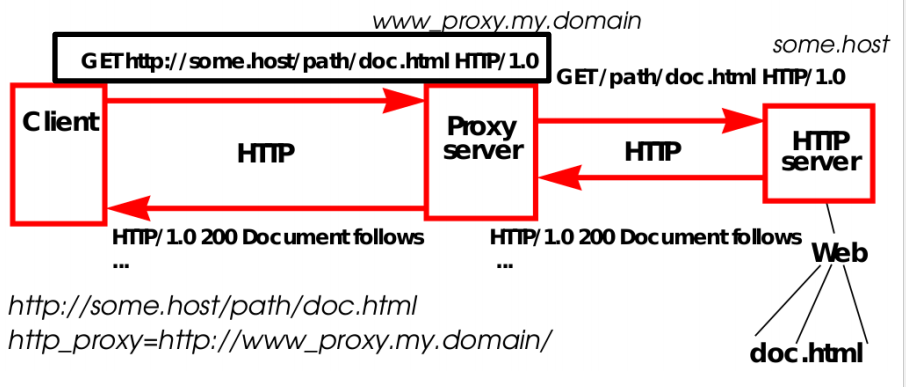
Oggi il proxy è un server interposto fra client e server utile ad effettuare le richieste al posto del client fornendo alcune funzionalità come autenticazione (in base a chi richiede la risorsa fornisce il permesso o no), autorizzazione, caching, logging.

Soprattutto la funzione di caching è stata molto importante in quanto, in caso di connessioni molto lente, il proxy velocizzava la velocità di risposta con cui si otteneva una risorsa. Il documento di risposta viene mantenuto dal proxy in modo da non richiederlo di nuovo al server in caso di request. Il problema è che questo documento può diventare obsoleto in quanto l’originale si potrebbe essere aggiornato. Per far fronte a questo si utilizza il campo “if modified since” nel request header.

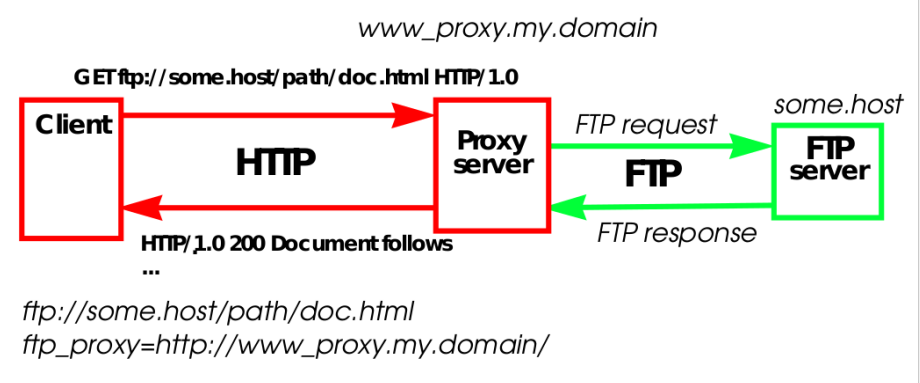
In una richiesta classica fra client e server, il client richiede una particolare risorsa ad un server identificato nell’header HTTP da un campo Host. Questo cercherà la risorsa nel suo file system e la tornerà come risposta.



In una situazione in cui vi è il proxy di mezzo, cambia il tipo di request che viene fatta dal client. Nella richiesta GET viene specificato un indirizzo assoluto come “some.host” ma il campo Host nell’header HTTP ha come valore quello del proxy server, quindi la richiesta dal client avrà come destinazione finale effettiva il proxy, il quale poi creerà una nuova request verso il server che detiene la risorsa che è some.host

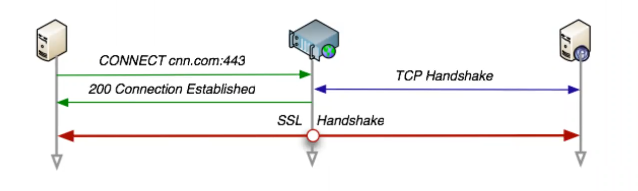


Il proxy può essere utilizzato anche per richiedere delle request diverse da HTTP, ma di tipo FTP per esempio che sono sempre delle richieste testuali. Basta settare il tipo di protocollo che deve utilizzare il proxy.



## Forward Proxy

Il concetto quindi di richiedere una risorsa ad un server con il proxy di mezzo è chiamato “forward Proxy”, cioè il proxy semplicemente inoltra quelle che sono le request del client verso il server. Le richieste standard che vengono fatte sono HTTP request inviate con path assoluto dal client verso il proxy.  
In aggiunta alle request HTTP, il forward proxy può essere utilizzato per altri tipi di request:

* **HTTP tunneling**: viene effettuata una richiesta non HTTP (FTP, per esempio) verso il server usando una richiesta HTTP verso il proxy.
* **HTTP connect**: Il proxy stabilisce una connessione TCP col server al posto del client e permette l’utilizzo e il passaggio di dati di qualsiasi protocollo che si basa su TCP a seguito della connessione.  
  Il proxy riceve dal client le informazioni sul server con cui vuole connettersi. Lui fa un TCP handshake col server. Da questo momento, ogni singolo byte sopra a TCP che il client manda al proxy viene inoltrato verso il server.  
  

Data una situazione in cui il traffico viene criptato con SSL, il proxy è impossibilitato a capire il contenuto dei pacchetti e a fungere content scan per preservare la sicurezza della rete. Quindi il proxy deve decriptare il traffico prima di inoltrarlo.  
In questo senso la TLS handshake non viene più svolta tra client e server ma tra client e proxy e tra proxy e server, in modo che il proxy abbia i certificati utili per la decriptazione e criptazione. Quando il server invia il traffico al client, viene decriptato e controllato in chiaro, criptato di nuovo e reinviato al client. Vedi SSL Bump.

Il proxy viene utilizzato per filtrare il contenuto di ciò che viene scambiato fungendo anche da blacklist, malware scan. Può bloccare l’accesso a siti (es: pornhub in università) o bloccare il passaggio di file eseguibili.

Si parla anche di anonymizer proxy quando il server proxy posto a metà tra client e server nasconde totalmente le informazioni del client al server, risultando come il mittente delle request.

## Reverse Proxy

Il Reverse Proxy è un proxy utilizzato dal server e non dal client, riceve le request dal client effettua alcune operazioni e poi inoltra la richiesta al vero server. È molto simile al forward proxy.

Le funzioni che svolge questo proxy sono:

* l**oad balancing**, permette il collegamento di un url a diversi server all’interno della rete privata in modo da distribuire le richieste in entrata su più server evitando il sovraccarico di un solo host.
* **cache static content**, mantenendo salvata nella cache risposte dei server ai client in modo evitare una ricerca successiva.
* **compression**, comprime i dati in entrata e in uscita
* **TLS acceleration**, a causa del peso computazionale per il processore nello scambio della chiave pubblica durante l’handshake di TLS potrebbe crearsi un collo di bottiglia lato server. Attraverso la SSL offloading il proxy si occupa dello scambio SSL mentre il server riceve semplicemente il traffico con il contenuto HTTP. Quindi il proxy decripta tutti dati SSL per poi inviare solo la vera request in chiaro riducendo l’overhead al server. Allo stesso modo il proxy può decidere di criptare il contenuto di nuovo e inviarlo al server dopo averlo controllato.
* **internal server protection**, praticamente è la funzione di anonimizzazione ma lato server. Nasconde i server e la logica con le quali le pagine vengono servite e si comporta come il mittente della response fosse il proxy stesso.
* **Cifratura HTTPS,** effettuandola viene scaricato il server dal compito di attuare la cifratura dei contenuti oppure viene offerta ai quei server che invece supportano solo HTTP.

L’internal server protection funziona che riceve le request dal client ed effettua alcune operazioni per creare una nuova request verso il server effettivo in modo da staccare quella che è la connessione diretta tra l’esterno e il server per difendersi da attacchi DoS. Fornisce inoltre supporto HTTPS per quei server che hanno solo HTTP, fornendo quindi un servizio che il server non possiede.

Un reverse proxy viene utilizzato come **Application firewall**, cioè un firewall a livello applicazione che applica delle regole per poter bloccare eventuali contenuti malevoli evitando attacchi SQL, XSS ed altri tramite l’ispezione del traffico HTTP. Tramite l’SSL offloading, il firewall Proxy riesce anche a ispezionare il traffico criptato e criptarlo prima di inviarlo al server.

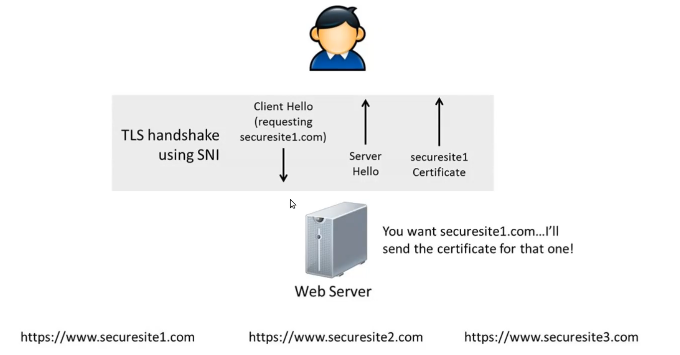
## SSL Bump

Dato che il traffico HTTPS è criptato, è impossibile leggere il contenuto di ciò che viene scambiato tra client e server. Il forward proxy quindi si pone in mezzo a questi due fingendo di essere il vero server. L’SSL Bump consiste nell’utilizzare il requested host name per generare dinamicamente un certificato server ed impersonare quel server richiesto al posto del server stesso.

Ma accade il HTTPS certificate dilemma quando un virtual host (stesso webserver che può ospitare più website) presenta diversi web site ognuno di questi con diverso certificato.

L’hostname da cui il proxy deve prendere il nome per fingere di essere il server e creare un certificato è contenuto all’interno dell’header http che però, come richiesta, viene inviata dopo TCP e SSL. Mentre TCP ed SSL lavorano con l’indirizzo IP del Virtual Host, il certificato invece lavora con il singolo dominio di interesse dietro a quello stesso IP. Il proxy, in questo modo, non riesce a capire quale specifico certificato deve creare perché il pacchetto http da cui prendere le informazioni arriva criptato, poiché a livello logico è già avvenuto lo scambio SSL in precedenza.

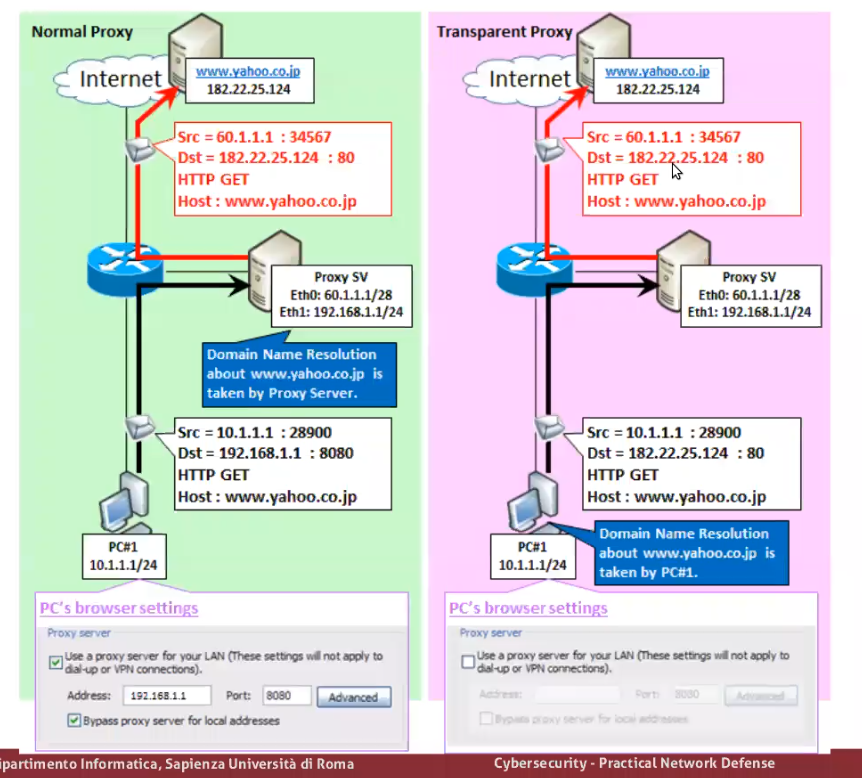
Viene fatto uso del **Server Name Indication (SNI)**, che è un’estensione di TLS dove il client indica l’hostname della richiesta HTTP all’inizio dell’handshake TLS in clear text. Questo permette ai virtual host di presentare più domini sullo stesso IP senza fornire lo stesso certificato per tutti.



## Transparent proxy

Il transparent proxy è una funzionalità che permette l’utilizzo di un proxy senza che venga settato manualmente dagli utenti, quindi senza che questi possano saperlo (appunto, trasparente).

Durante uno scambio di pacchetti, all’inizio della sessione, un client invia un pacchetto TCP verso il Server. Questo pacchetto viene intercettato dal Proxy che vedendo la destinazione diversa dal proprio indirizzo, lo accetta fingendo di essere il Server, ed effettua la richiesta al posto del server.



Come si può vedere nel primo caso il client è configurato per inviare i pacchetti al proxy indicando l’host finale. Nel secondo il client non ha nessun proxy configurato ed invia la richiesta direttamente al server finale ma il proxy di mezzo intercetta il pacchetto, lo elabora e lo inoltra. È come un NAT, dove il proxy rimuove i dati iniziali generandone di nuovi. Quindi esegue una funzionalità NAT a livello applicativo.

Questi pacchetti vengono intercettati tramite la Policy based routing (PBR) dove viene decisa una policy per intercettare dei pacchetti specifici. Nel caso di HTTP, viene intercettato qualsiasi connessione TCP con porta di destinazione 80 sennò tutto va diretto verso il server. Questo viene fatto attraverso iptables con la “packet marking”.



L’ICAP (internet content adaptation protocol) è un protocollo che fornisce uno standard per processare i pacchetti HTTP. Quando vengono ricevuti questi pacchetti, vengono applicate delle trasformazioni prima di reinviarle alla destinazione finale.

Il proxy server viene usato come ICAP client. Riceve dai client delle request, esegue delle trasformazioni, passa il contenuto ad un server ICAP che processerà il pacchetto per reinviarlo ai client.

# Intrusion Detection System

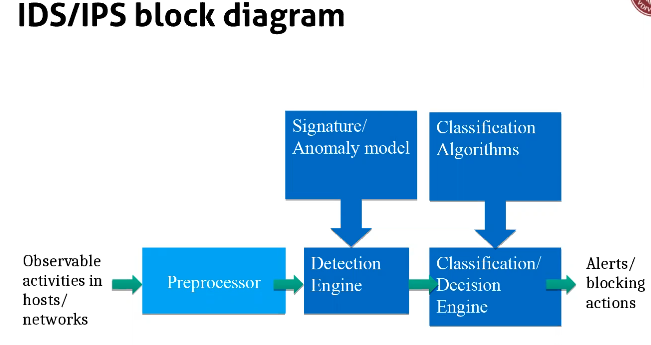
IDS è un Sistema atto a trovare la presenza di alcuni intruders all’interno del sistema per evitare che causano danni. L’IDS si è evoluto in Intrusion Prevention System (IPS) il quale a seguito di un riscontro effettua delle azioni, come modifica del firewall per bloccare questo tipo di attacchi.

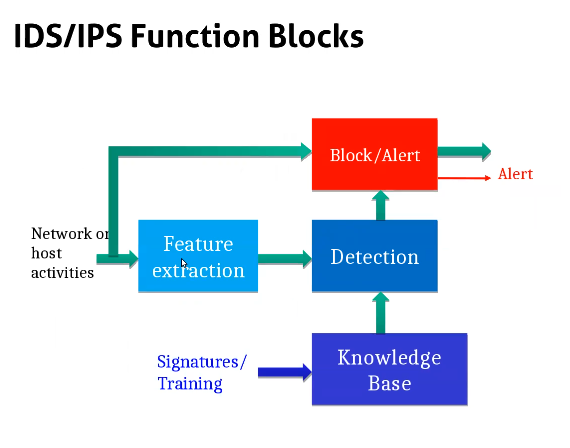
Può essere visto come un device o come un servizio all’interno del device posto dietro il firewall perché generalmente tutti gli intruder vengono bloccati dal firewall stesso. Lo fa attraverso il controllo del payload del pacchetto.

L’IDS trova le intrusioni attraverso l’out of band: osserva ciò che succede nella rete e lo trova senza controllare il contenuto dei pacchetti. È un metodo passivo.

L’IPS attraverso l’in line: deve essere visto come un intermediario (tipo firewall, router) della rete che controlla il contenuto dei pacchetti bloccandoli o meno. È un metodo attivo, reagendo dopo aver trovato l’intrusore.

Gli allarmi che vengono creati possono essere positivi o negativi. Un allarme può essere un falso positivo quando un pacchetto “legale” viene visto dal sistema come “intrusion” e fa scattare un allarme in modo da bloccarlo, sbagliando. Un falso negativo è invece quando un allarme non viene generato ma dovrebbe esserlo. Un vero negativo è quando un pacchetto legale viene visto come legale.

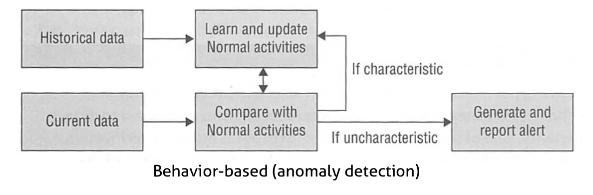
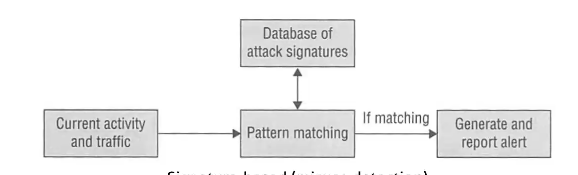




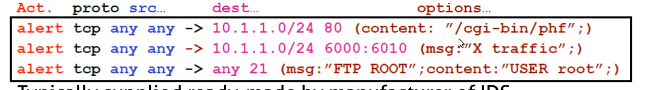
I tipi di IDS sono:

* host based (hids): si focalizzano su tutti gli elementi che riguardano l’host, non solo pacchetti che entrano ed escono dall’host. Anche chiamate di sistema, login, cambi di configurazioni e altre cose.
* network based (nids): Opera in promiscuous mode (sniffer). É generalmente connesso agli switch con port mirrored (span mode with switch port analizer). Tutto il traffico generato sulle porte dello switch viene replicato sulle mirrored port dove si trova il NIDS. Può presentare una serie di sensori posti in vari punti della rete (distributed detection system)
* wireless (wids)

L’approccio dell’IDS si basa su due modi:

* behavior based (anomaly detection): in questo caso viene creato un modello di attività “normale” dei dati e quindi dall’analisi dei dati ricevuti vengono cercati comportamenti che sono lontani e anomali da questo modello. (esempio: solitamente si hanno 300 chiamate al server, un giorno se ne hanno 1000). La difficoltà sta nel definire tutti i possibili comportamenti normali. Questo comporta che le nuove attività ricevute, seppure “normali”, vengano viste come intrusion creando un falso positivo.  
  
* Signature based (misuse detection): invece di creare un modello di comportamento normale, si ha un database di “attack signature” e vengono quindi confrontati i dati con quelli che sono gli attacchi già conosciuti. La difficoltà sta nel creare un catalogo immenso di tutti gli attacchi per attività anormali. Se qualche attacco manca in questo database, si viene a creare un false negative cioè un’attività negativa che non viene vista come intrusione. Viene controllato il payload ed estratto il pattern (signature) che serve per confrontare con gli attacchi che abbiamo nel database.  
  

Nella signature ci sono delle regole che definiscono le condizioni del confronto del payload in entrata:

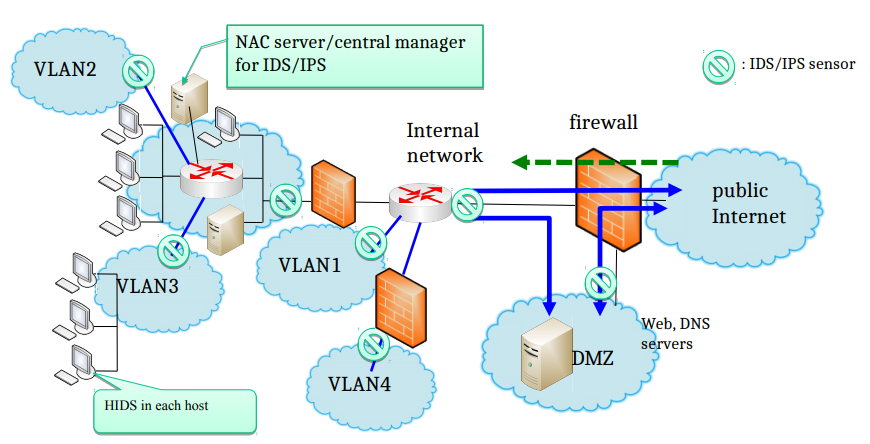


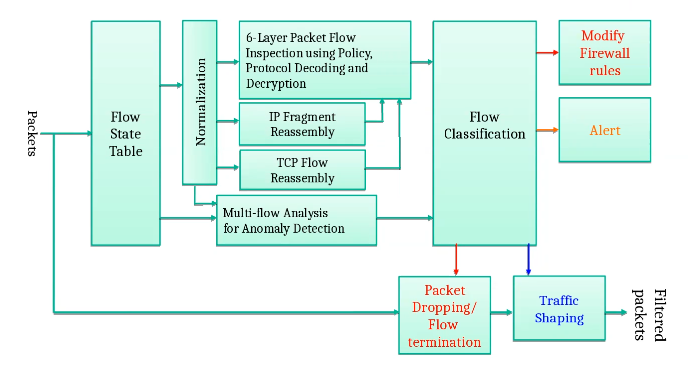
Le problematiche sono che non tutti i tipi di traffico possono essere controllati, ad esempio quelli criptati (vpn, ssl) e soprattutto va in difficoltà quando c’è una grande quantità di traffico.

Honeyspot viene usata per la signature extraction. L’idea dell’honeyspot è quella di attrarre gli attacchi in modo da collezionare esempi di attacchi per essere protetti in futuro.

Nel behavior based detection può essere aggiunta una tecnica basata sull’anomalia dei protocolli o anche sulla statistica.

Nel Distributed IDS si parla di Intrusion Detection Exchange protocol, cioè un protocollo a livello applicazioni di scambio di data tra vari IDS. Tramite l’analisi di singoli messaggi è possibile fare un resoconto di quello che potrebbe essere un attacco all’interno della rete

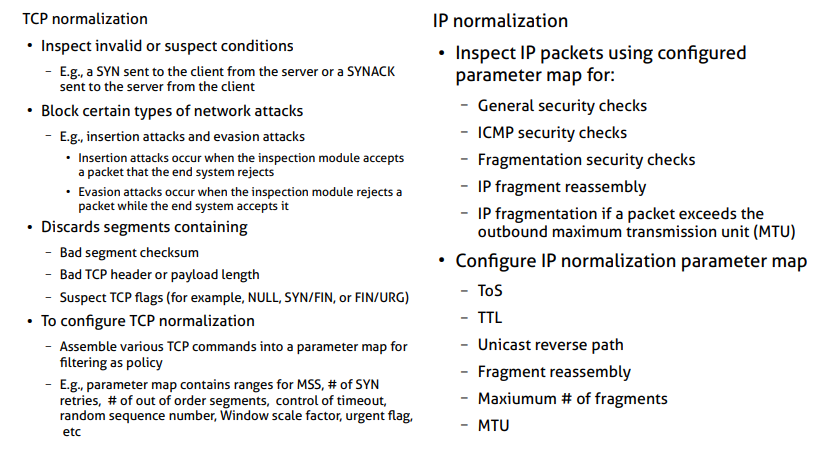


Il network IPS presenti dei diagrammi a blocchi: 

Viene creata una rappresentazione di ciò che si ha nel sistema tramite la flow state table, poi attuano la normalizzazione e l’inspection, ip fragment riassemby, TCP flow riassembly. Per poi passare alla flow classification che permette la modifica del firewall e la generazione di alert.

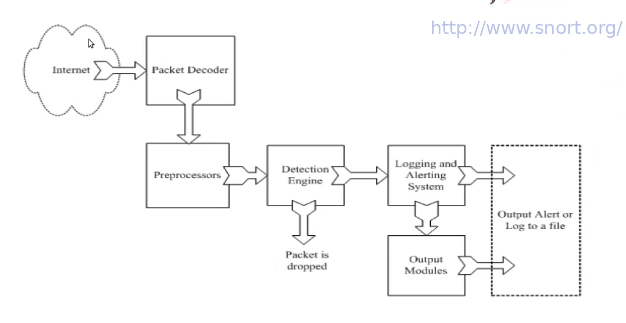
La comunicazione tra due endpoint è una sessione e le informazioni che la riguardano vanno a creare quello che è un flusso.

Il concetto di normalizzazione avviene se si vuole controllare se vi sono strane condizioni all’interno dei pacchetti come flag, scartare pacchetti strani, controllare dei parametri in IP.



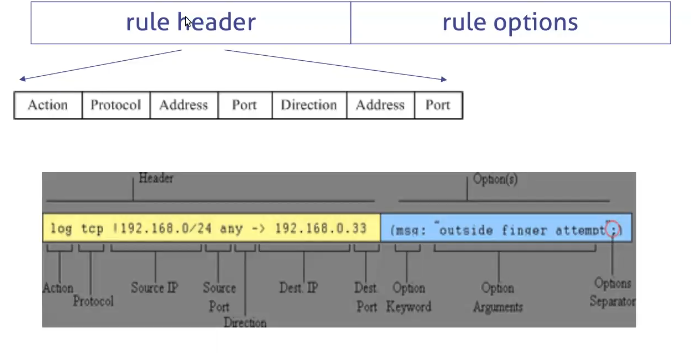
## SNORT

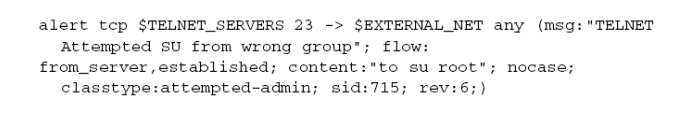
È il piu famoso IDS/IPS signature-based.



I pacchetti vengono presi dal packet decoder e passati alla parte di preprocessor che è quella di normalizzazione (TCP fragmentation, ip, riassembly). Il detection engine applica le regole sui pacchetti.

Le regole sono composte da:

* **Header,** sono i campi che vengono controllati da snort in un pacchetto  
    
  
* **Options,** viene specificato il pattern da cercare



## Suricata

Network IDS/IPS. Sviluppato per superare i limiti di SNORT. Cerca di distribuire il carico di lavoro tramite una struttura multi thread.

## Fail2Ban

È un IPS che ha come scopo quello di proteggere gli host da attacchi bruteforce facendo lo scan dei log e bannando quegli IP che in un certo arco temporale hanno avuto un numero elevato di failure nel tentativo di attaccare una macchina. È molto utile pensarlo come se fosse un firewall dinamico dato che adatta le regole di ban in base alla quantità di failure che vi sono nel sistema.

I problemi di fail2ban sono che è semplicemente un log parser quindi tutte le valutazioni del caso vengono effettuate successivamente alla scrittura di un file di log e generalmente attende un 1 secondo prima di ricontrollare il file per eventuali cambiamenti, e in quel lasso di tempo potrebbero esserci più failure di quelli concessi

# SIEM

Security information Management (SIM) colleziona i dati di log per analizzarli allertando i responsabili della sicurezza.

Security event management (SEM) conduce un monitoraggio in real time notificando agli admin di rete eventuali problemi ed eventi correlati.

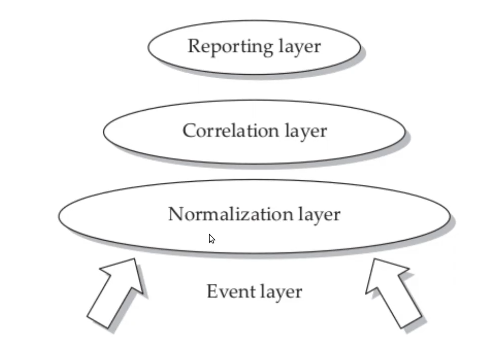
SIEM è composto da diversi software che uniscono SIM e SEM. Non esiste uno protocollo standard per SIEM ma il lavoro che fa è più o meno composto da: collezionare informazioni di processi di processi automaticamente, salvarli in un database centralizzato, correla eventi diversi tra loro, crea alert e report basati su queste informazioni e può funzionare in modo agentless o in agent mode.

Inoltre fornisce le seguenti funzionalità:

* **Log management**: alcuni nodi critici all’interno della rete invia log ad un database centralizzato che è gestito da SIEM, quindi i dati vengono normalizzati e poi fornisce quello è un log storage organizzato per soddisfare i requisiti. Più nodi appartengono al sistema SIEM più accurata sarà l’analisi tramite i dati che vengono inviati
* **IT regulatory compliance:** Una volta catturati i dati vengono creati filtri o regole per fare da audit e validare le compliance oppure identificare le violazioni di queste. (SLA) (monitorare la frequenza di cambiamento delle password, frequenza degli antivirus..)
* **Event correlation:** SIEM riesce a capire quali eventi tra di loro diversi possono essere correlati prima di triggerare un allarme. In questa maniera viene mostrata una immagine più completa dello stato di salute del sistema.
* **Active response:** SIEM si attiva in seguito all’identificazione di alcuni eventi e gestisce alcune procedure di sicurezza
* **EndPoint Security:** alcuni SIEM possono monitorare la sicurezza di alcuni endpoint per validare in modo centralizzato la sicurezza del sistema. Alcuni possono anche gestire le policy di sicurezza di questi endpoint potendo anche installare aggiornamenti o gestire le ACL sui firewall

Differenza fra data:

* Event data: è il log vero e proprio, lista di eventi che accadono nel sistema
* State data: dati che rappresentato lo stato attuale del sistema. è un tipo di dato piuttosto statico, istantaneo e comprende configurazioni, applicazioni in uso, utenti attivi, processi, impostazioni di registro



Lo stack del SIEM è composto da diversi livelli. Il livello di normalizzazione applica alcune trasformazioni, poi vengono fatte delle correlazioni fra eventi per poi poter creare dei report, dashboard.

La funzionalità di correlare log consiste nel monitorare i log in entrata cercando dei pattern logici fra i log, delle relazioni e dei valori all’interno di questi log. Questo per analizzare ed identificare eventi invisibili al sistema.

I supporting data di SIEM sono dei dati collezionati da altre fonti che possono essere importate all’interno di SIEM per effettuare delle comparazioni.

La correlazione degli eventi è ciò che generalmente distingue i diversi SIEM. Prima che vengano correlati devono passare la normalizzazione in modo che i dati abbiano un sintassi comune per essere utilizzati nel SIEM. Vengono applicate poi delle regole di correlazione che triggerano alert o azioni

24/06/20

1. **List and detail which are the characteristics a host needs to have in a LAN for being the next hop for a given destination network.**
2. **Describe the 3-1-4 rule for /64 GUA in IPv6 and explain why you think it is useful.**

7/7/19

**Consider the connection tracking mechanism (implemented for example in iptables): in which type of firewalls can you find it? Describe how it works and explain why it is important.**

Il connection tracking può essere trovato nei firewall statefull cioè in quei firewall che non filtrano i pacchetti in modo indipendente l’uno dall’altro ma tengono traccia delle connessioni e del loro stato.

Viene usato una tabella di stato interna al firewall nella quale ogni connessione TCP e UDP viene rappresentata da due coppie formate da indirizzo ip e porta. Per tenere traccia dello stato di una connessione vengono controllati i flag del pacchetto e gli indirizzi per comprendere se un pacchetto appartiene ad una connessione stabilita in precedenza o se sta aprendo una nuova connessione. Il firewall permetterà il passaggio o meno in base al tipo di connessione che si vuole venga stabilita.

Questo tipo di soluzione permette un overhead inferiore per il firewall in quanto non deve controllare i parametri di tutti i singoli pacchetti ma può fare riferimento al flag.

**What function does each of the following iptables rules serve?**

a) iptables -A INPUT -s 192.168.23.12 -p TCP --sport 22 -j ACCEPT

b) iptables -t nat -A POSTROUTING -o $ext\_if -j MASQUERADE

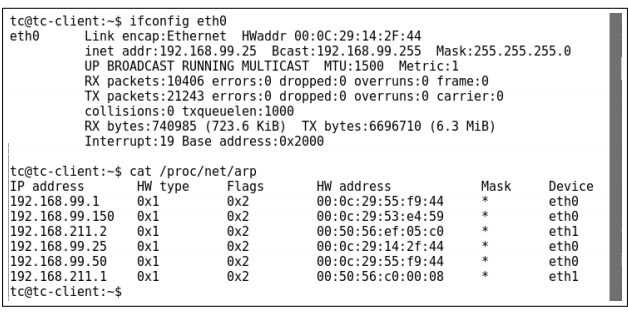
c) iptables -t nat -P DROP

d) iptables -A FORWARD -p udp -m state --state ESTABLISHED -j ACCEPT

e) iptables -A INPUT -j DROP

1. Viene aggiunta una regola per la tabella filter sulla catena di input (cioè i pacchetti destinati all’host stesso) accettando tutti quei pacchetti la cui source è 192.168.23.12, i quali utilizzano un protocollo TCP e la source port è la 22 (SSH)
2. Viene aggiunta una regola per la tabella NAT sulla catena di postrouting ( la modifica viene effettuata dopo che viene decisa la route in uscita per il pacchetto) sull’interfaccia esterna applicando la masquerade (dynamic nat simile source nat con ip address assegnato dinamicamente. Viene scelta in modo intelligente usando l’indirizzo dell’interfaccia da cui il pacchetto esce) [questa regola permette ad host con indirizzi privati in una nat di parlare con l’esterno mascherando il loro indirizzo privato con quello firewall utilizzando quello dell’interfaccia in -o]
3. Viene imposta come policy di default sulla tabella NAT il drop di tutti i pacchetti, ciò significa che ogni pacchetto che viene sottoposto alle regola della tabella NAT, se non ne soddisfa nessuna verrà bloccato.
4. Viene aggiunta una regola sulla tabella filter accettando pacchetti udp appartenenti ad una connessione stabilità in precedenza che devono essere trasmessi ad altri host
5. Viene aggiunta una regola sulla tabella filter di drop di qualsiasi pacchetto in entrata su quell’host

**Figure 1 reports the outcome of some bash commands. Describe what you can observe and explain what you think is relevant. Finally, suggest what you would explore further and why**



Nella figura notiamo che l’host preso in considerazione sull’interfaccia eth0 possiede un indirizzo ip 192.168.99.25/24 con netmask 255.255.255.0. Vi sono mostrati anche il numero di pacchetti trasmessi e ricevuti tramite questa interfaccia e che presumibilmente sono stati scambiati con gli host appartenenti alla sua LAN.

Nella sua arp table troviamo 6 record. 4 di questi record appartengono alla stessa subnet mentre due record appartengono ad un’altra subnet uscenti su due interfacce diverse.

….. ????

**It’s not always true that its better to have false positives than false negatives. Explain what they are in their wider meaning and provide two examples: one in which you think it’s better to have less false negatives and one in wich it’s possible the opposite**

**Describe the main roles and characteristics of Link-local Unicast addresses in IPv6.**

Gli indirizzi Link local sono i corrispettivi degli indirizzi privati dell’IPV6. Non sono routabili globalmente ma vengono usati per poter comunicare tra diversi host all’interno dello stesso link. Non essendo routabili non possono essere inseriti nelle tabelle di routing.  
Sono caratterizzati da un indirizzo tipo: fe80: fino a febf per i primi 10 bit,54 bit e poi 64 bit dell’interface ID che possono essere random o static oppure creati dall’indirizzo mac.

Gli indirizzi local vengono usati per richiede ed ottenere l’indirizzo global al router tramite SLAAC o DHCP6

**Host A in LAN1 has to send a UDP packet to host B in LAN2. Succinctly describe source and destination MAC and IP addresses host A will use and if it needs to use ARP. Also report how the same process changes when using IPv6.**

L’host A che vuole inviare un pacchetta verso Host B di cui conosce l’indirizzo IP invia una ARP request in broadcast contenenti il proprio indirizzo MAC e l’indirizzo IP del destinatario di cui si vuole conoscere il MAC. Tutti gli host della Lan2 ricevono l’arp request e viene fatto il confronto fra l’indirizzo IP del destinatario del pacchetto e il proprio. Se combacia, risponde con un ARP Reply contenenti il proprio MAC address.

In questa maniera l’Host A riceve il pacchetto di risposta, aggiorna la cache dell’arp table ed invia il pacchetto con i campi aggiornati.

Per quanto riguarda IPV6, l’arp request e reply viene sostituita dalla neighbour solicitation e advertisement. Non è presente la tabella ARP ma la neighbour Cache.

All’interno dell’ICMPv6 viene indicato come target l’indirizzo ipv6 dell’host B per cui si richiede il MAC . Come destinazione del pacchetto ipv6 viene settando il solicited node multicast address dell’host b.

Il multicast address del MAC viene ottenuto dall’ultima parte del solicited node multicast address.

La risposta invece ha come destinatario l’indirizzo unicast.

10/9/2019

**Provide the sequence of iptables commands you would use in a firewall with two interfaces ($INT\_IF and $EXT\_IF) that acts as the gateway for $INSIDE\_NET (that is reachable from interface $INT\_IF), considering that:**

**• internal host1 (with IP $HOST1\_IP) has to be allowed only to access remote hosts with HTTPS and DNS**

**• internal host1 can operate via ssh with the firewall**

**• no other traffic has to be allowed**

Iptables -A FORWARD -s $HOST1\_IP -i $INT\_IF -dport 443 -o $EXT\_IF -m state –state NEW, ESTABLISHED -j ACCEPT

Iptables -A FORWARD -d $HOST1\_IP -o $INT\_IF -sport 443 -i $EXT\_IF -m state –state ESTABLISHED -j ACCEPT

Iptables -A FORWARD -s $HOST1\_IP -i $INT\_IF -dport 53 -o $EXT\_IF -m state –state NEW, ESTABLISHED -j ACCEPT

Iptables -A FORWARD -d $HOST1\_IP -o $INT\_IF -sport 53 -i $EXT\_IF -m state –state ESTABLISHED -j ACCEPT

Iptables -A INPUT -s $HOST1\_IP -i $INT\_IF -dport 22 -p TCP -m state –state NEW, ESTABLISHED -J ACCEPT

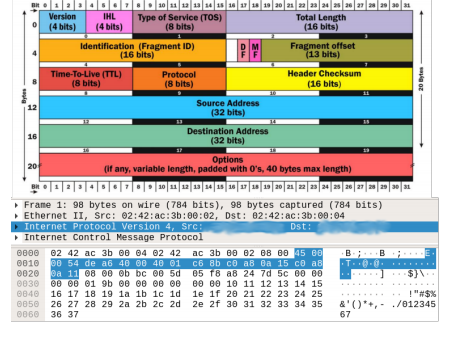
Iptables -A OUTPUT -d $HOST1\_IP -o $INT\_IF -sport 22 -p TCP -m state –state ESTABLISHED -J ACCEPT

Iptables -P FORWARD DROP

Iptables -P INPUT DROP

Iptables -P OUTPUT DROP

**3. Considering the structure of the IP packet header provided below, report the following info related to the shown packet:**



• Header length: 128/160 bit

• IP data size:

• Time-to-live:

• Source IP address in dotted-decimal