

Teoria

13/05/2023

prof. Michele Pagano

Email:

- m.pagano@iet.unipi.it
- michele.pagano@unipi.it

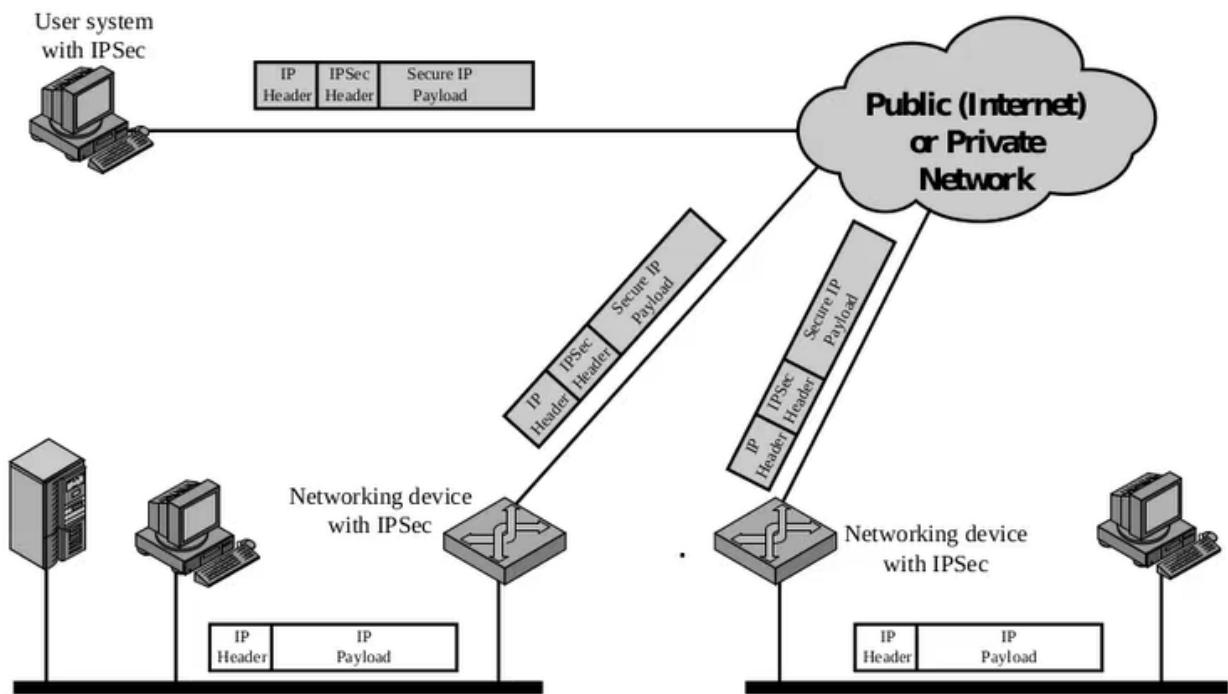
prof. Christian Callegari

Esame: prevede due parti, una parte teorica e una parte pratica. La parte pratica riguarderà rifare qualcosa visto durante le esercitazioni. Per la teoria si tratterà di un quiz di 20 domande a risposta multipla.

Libro di riferimento: *W. Stallings, Cryptography and Network Security: Principles and Practice*

IPSec

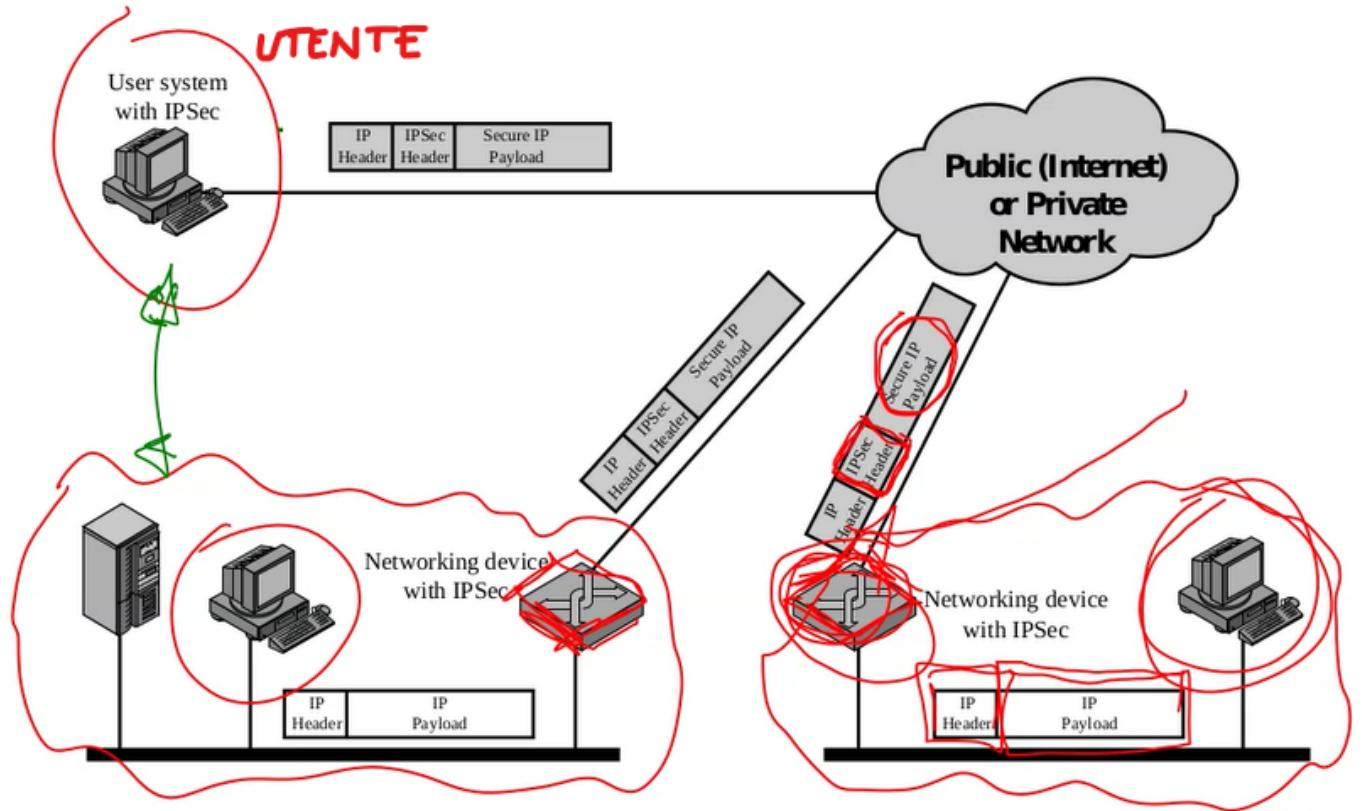
IPSec: Typical Scenario



La soluzione per realizzare delle reti private facendo uso della rete Internet o di qualunque infrastruttura di rete sulla quale non siano garantiti servizi di sicurezza è usare **IPSec**. Fin tanto che comunico all'interno della rete locale non ho bisogno di alcuna protezione, ma supponiamo che i pacchetti debbano raggiungere un sito remoto tramite un canale non sicuro. Se pensiamo alla comunicazione tra un utente in una filiale e un utente in una altra filiale, tutto il traffico può essere ascoltato da un man-in-the-middle ma usando IPSec sarà in grado di conoscere solo la destinazione e la sorgente dei gateway esterni, senza però conoscere l'indirizzo IP usato dai client che si trovano dietro i gateway, in quanto viene nascosta anche l'applicazione che viene usata. L'altro caso possibile è

proteggere l'utente che si trova in remoto:

IPSec: Typical Scenario



Intuitivamente se uso degli algoritmi crittografici avrò necessità di scambiarmi delle chiavi tramite un altro protocollo di scambio delle chiavi detto **IKE (Internet Key Exchange)**.

IPSec Intro

- Many solutions are application-specific
 - TLS for Web, S/MIME for email, SSH for remote login
- IPsec aims to provide a framework of open standards for secure communications over IP
 - Protect every protocol running on top of IPv4 and IPv6
- IETF standard for real-time communication security
- Implemented at IP layer, all traffic can be secured no matter what application
- Transparent to applications, no changes on upper-layer software
- Transparent to end users, no need to train users on security mechanisms, issuing keying material on a per-user basis, or revoking keying material when users leave

IPsec = AH + ESP + IPcomp + IKE

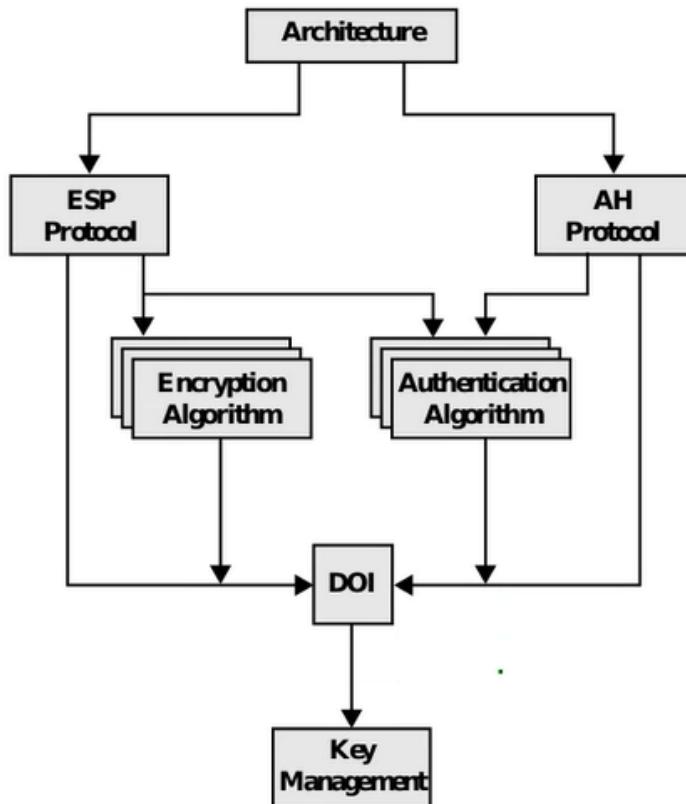
Molto spesso vengono creati dei protocolli di sicurezza o a livello applicativo o a livello di trasporto. IPSec invece anzichè essere un protocollo specifico ad un livello alto è un protocollo che rende sicura qualsiasi applicazione che faccia uso di IP al livello network. L'anno di nascita dell'idea di IPSec è il 1994 anni in cui inizia la standardizzazione di IPv6. Si tratta di uno standard IETF a differenza di tanti algoritmi di cifratura che sono standardizzati dal NIST dunque le specifiche di IPSec si trovano nelle RFC della IETF (infatti una delle critiche ad IPSec è che si tratta di una struttura mastodontica fatta da tante RFC!). IPSec è trasparante alle applicazioni dunque non devo modificare il software delle applicazioni che usano IPSec. IPSec nelle reti aziendali viene gestito solo dal gateway di frontiera con il vantaggio di non avere overhead dovuto alla sicurezza. Gli utenti non conosceranno dunque le chiavi usate da IPSec dunque non c'è alcun problema nel continuare a usare le chiavi usate da IPSec in quanto queste sono trasparenti all'utente finale.

IPSec si compone di 4 elementi. **IKE** come già accennato si occupa dello scambio delle chiavi, **AH** e **ESP** sono 2 protocolli, l'Authentication Header e l'Encapsulating Payload, l'**IPcomp** è il compression protocol, il quale non viene più usato al giorno d'oggi, aveva un senso in collegamenti a basso rate. Un buon algoritmo di compressione rende tutti i simboli equiprobabili e quindi il testo cifrato appare come simboli randomici. Se i simboli sono randomici è difficile comprimerli (con ad esempio zip, gzip) in quanto questi algoritmi di compressione funzionano bene con testi ridondanti e dunque sarà necessario se voglio ricorrere alla compressione, prima comprimere e poi cifrarli. **Questa parte di compressione è stata di fatto eliminata nel tempo.**

IPSec History

- In 1994, the Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture" (RFC 1636)
- Two versions of IPsec can currently be found in implementations
 - The "new" IPsec (IPsec-v3)
[RFC 4301, Security Architecture for the Internet Protocol \(Dec. 2005\)](#)
 - The "old" IPsec (IPsec-v2), still commonly found in operational use
[RFC 2401, Security Architecture for the Internet Protocol \(Nov. 1998\)](#)
 - Earlier version of IPsec (defined in RFCs 1825-1829) was obsoleted by IPsec-v2
- Two versions of IKE
 - The "new" IKE (IKEv2)
[RFC 5996, Internet Key Exchange Protocol Version 2 \(IKEv2\) \(Sept. 2010\)](#)
 - The "old" IKE (IKEv1)
[RFC 2409, The Internet Key Exchange \(IKE\) \(Nov. 1998\)](#)
deprecated by [RFC 9395 \(April 2023\)](#)

IPSec v2: Documents



RFC 2411 (November 1998) - IP Security Document Roadmap

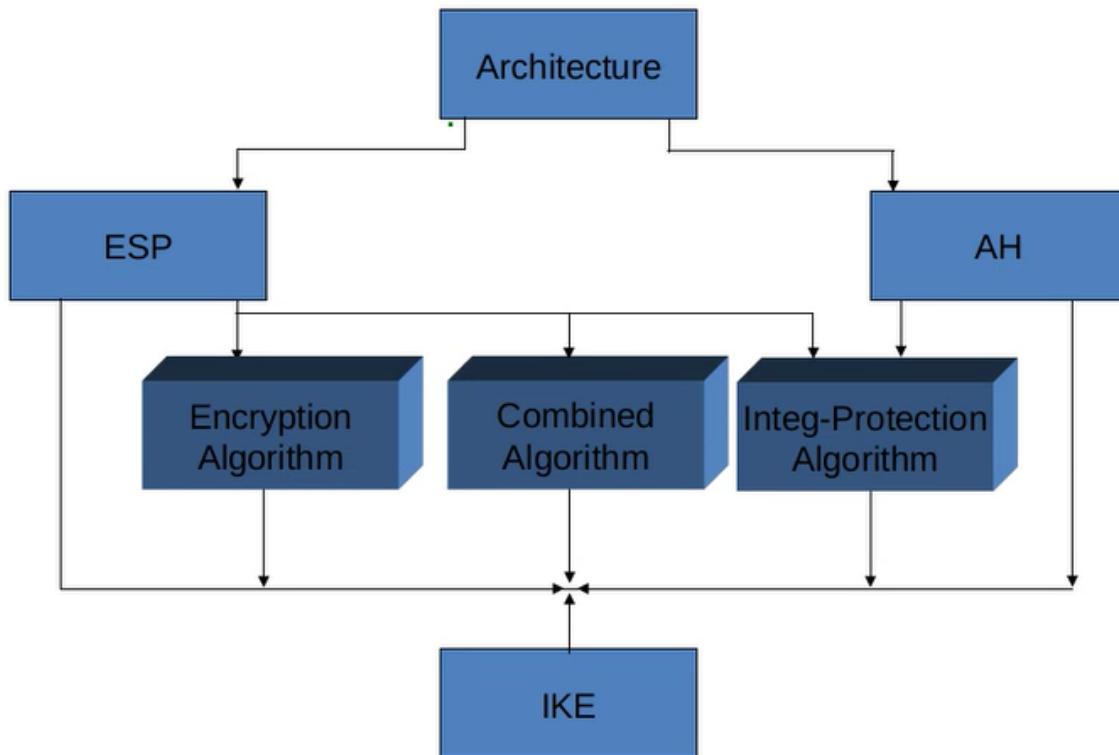
IPSec è un framework molto complicato ma ci sono dei documenti particolari che definiscono l'organizzazione degli aspetti del protocollo. Ci sono delle RFC generali che descrivono quali sono gli elementi funzionali, poi sono stati definiti due protocolli specifici, uno è **ESP** che fornisce **CIFRATURA + AUTENTICAZIONE** (in senso lato, ovvero vorrà dire sempre autenticazione + integrità come ad esempio HMAC), mentre **AH** fornisce solo **AUTENTICAZIONE**.

❓ Che senso ha avere AH quando esiste ESP che fornisce entrambe le funzionalità?

In realtà ci sono alcune piccole differenze relativamente alla protezione di alcuni campi del pacchetto IP.

Nella versione 2 presentata in alto viene rappresentato anche questo **Domain of Interpretation** che definisce gruppi di algoritmi da usare come unica suite, in modo da facilitare la negoziazione iniziale, in quanto la grande difficoltà dell'uso di IPSec come anche per TLS è che tipicamente non so che versione di protocollo abbia implementato l'altro endpoint, quindi è necessario accordarsi sugli algoritmi da usare.

IPSec v3: Documents



RFC 6071 (February 2011) - IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap

La parte superiore è rimasta inalterata, ci sono sempre ESP e AH, la cosa che cambia in questa versione 3 è la parte intermedia per capire le funzioni che svolgono ESP e AH. Per AH non è cambiato nulla mentre per ESP è cambiato in quanto abbiamo aggiunto il blocco in mezzo detto di "**Combined Algorithm**". Algoritmi combinati sono algoritmi che permettono di realizzare con un singolo algoritmo entrambe le funzioni di **cifratura e autenticazione**.

L'AES in modalità Galois Counter Mode è un esempio di Combined Algorithm che permette di ottenere insieme cifratura e autenticazione sui dati trasmessi ed eventualmente altri campi. Questa osservazione sarà compresa meglio quando vedremo il funzionamento di ESP e AH.

Perchè questi algoritmi sono separati dal protocollo di base?

Il protocollo in genere ha una struttura fissa che non cambia nel tempo, gli algoritmi invece hanno un'evoluzione, algoritmi che venivano usati negli anni 90 ora non sono più raccomandati per problemi di sicurezza: il DES ed RC4.

IPSec - Servizi offerti

IPSec - Services

- Authentication and integrity for packet sources
 - Ensures connectionless integrity (for a single packet) and partial sequence integrity (prevent packet replay)
- Confidentiality (encapsulation) for packet contents
 - Also partial protection against traffic analysis (flow level confidentiality)
- Authentication and encapsulation can be used separately or together
- Either provided in one of two *modes*
 - Tunnel
 - Transport
- These services are transparent to applications above transport (TCP/UDP) layer

Authentication and integrity: Una protezione a livello dei singoli datagrammi inviati, per cui i vari pacchetti sia quelli in chiaro che autenticati con IPSec possono andare anche persi, dunque i singoli datagrammi che arrivano saranno integri e non modificati. Inoltre può anche garantire una parziale integrità della sequenza ovvero una protezione per eventuali attacchi di tipo replay.

Confidentiality (encapsulation) for packet contents: può fornire confidenzialità a livello di flusso e si associa all'analisi statistica del traffico. Se volessi analizzare quale protocollo è stato usato su un traffico cifrato, con l'header IPSec non vedo quale protocollo di trasporto ne applicativo viene usato. Si possono fare delle analisi statistiche analizzando i tempi di risposta, le dimensioni dei pacchetti, cercando di ricostruire le informazioni trasportate.

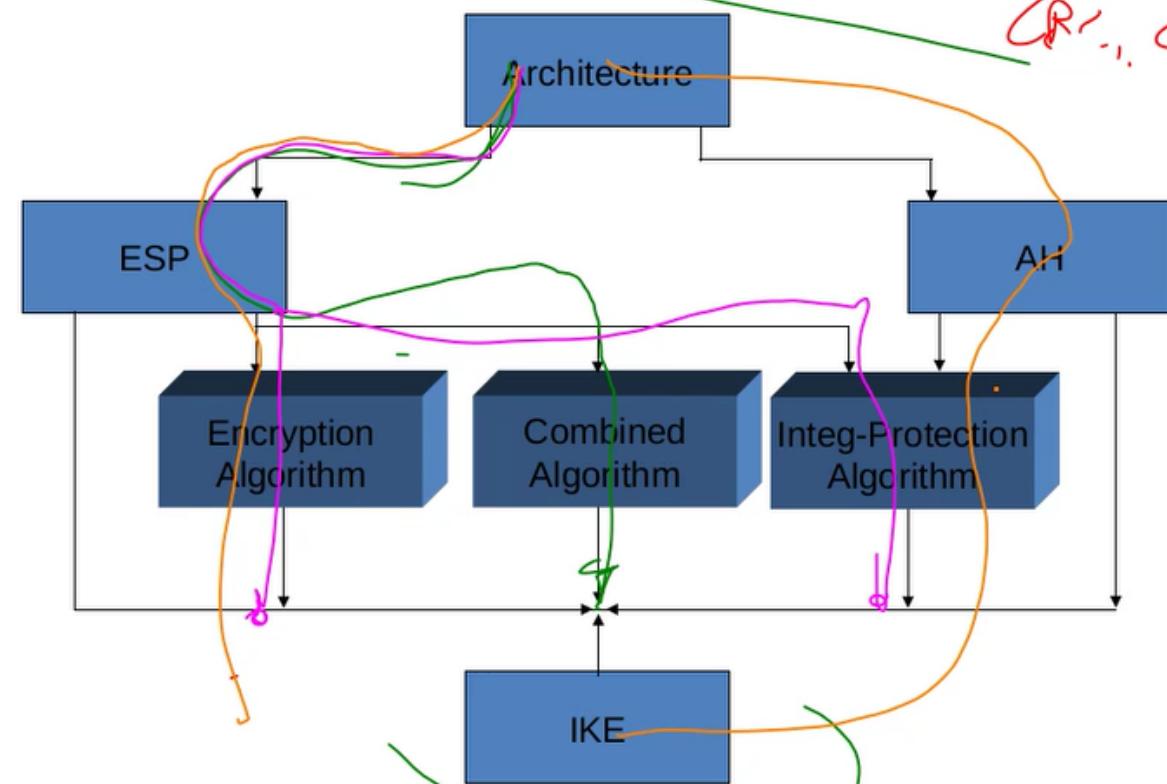
Per poter ingannare queste analisi statistiche è possibile modificare la lunghezza del pacchetto aggiungendo dei byte di riempimento.

Si potrebbe richiedere solo un servizio di autenticazione o solo un servizio di privacy oppure posso chiederli entrambi.

Usando dunque ESP con un algoritmo combinato ho sia auth che cifratura. Oppure potrei usare solo ESP con algoritmi di cifratura e algoritmi di integrità. Oppure potrei pensare di usare ESP con un

algoritmo di cifratura ed AH con un algoritmo di autenticazione:

IPSec v3: Documents



RFC 6071 (February 2011) - IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap

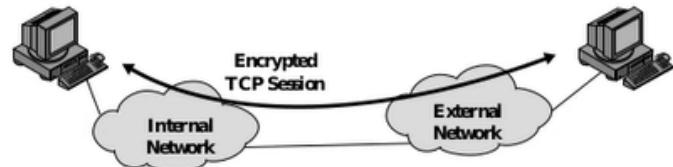
IPSec modes

IPSec può funzionare in modalità **tunnel** o in modalità **trasporto**:

IPSec modes

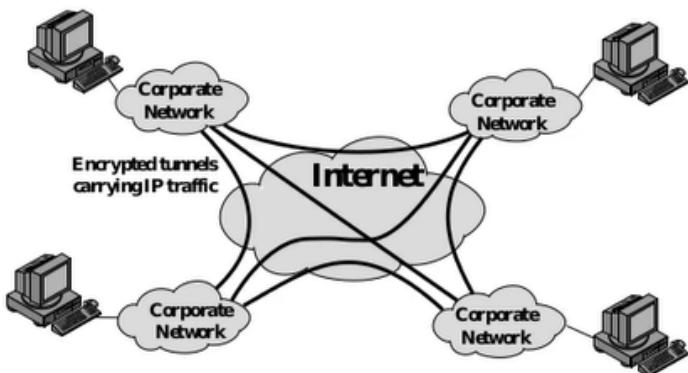
- **Transport mode**

- Used to deliver services from host to host or from host to gateway
- Usually within the same network, but can also be end-to-end across networks



- **Tunnel mode**

- Used to deliver services from gateway to gateway or from host to gateway
- Usually gateways owned by the same organization
 - With an insecure network in the middle



la modalità di trasporto si utilizza per creare connessioni IPsec tra host e host mentre la modalità tunnel si usa tipicamente tra gateway di reti diversi ma anche quando un gateway comunica con un singolo host all'esterno della rete.

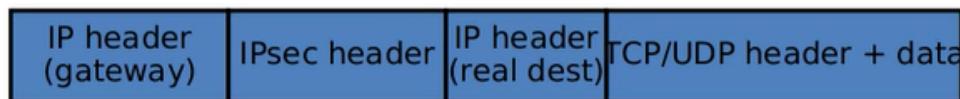
Provando a leggere la struttura semplificata dei pacchetti IPSec descriviamo le 2 modalità presentate:

IPSec modes

- **Transport mode** secures packet payload and leaves IP header unchanged
 - Used to deliver services from host to host or from host to gateway
 - Usually within the same network, but can also be end-to-end across networks



- **Tunnel mode** encapsulates both IP header and payload into IPsec packets
 - Used to deliver services from gateway to gateway or from host to gateway
 - Usually gateways owned by the same organization
 - With an insecure network in the middle

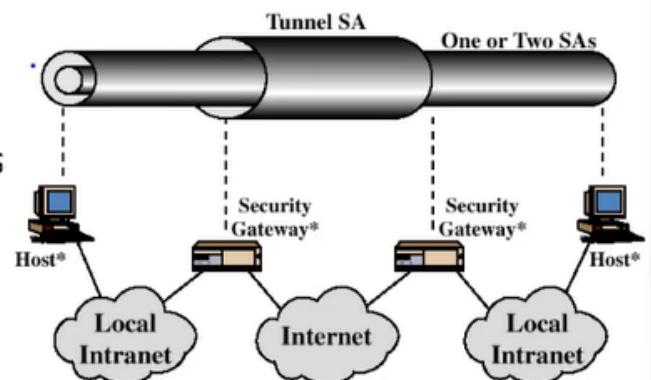


- **transport mode:** i servizi di sicurezza (encryption e/o authentication) vengono garantiti al SOLO payload del pacchetto IP mentre lascia l'header IP inalterato. In modalità trasporto il nuovo pacchetto forgiato tramite IPSec avrà lo stesso header IP originario, poi ci sarà l'header IPSec che potrà essere AH o ESP e poi ci sarà il payload protetto, ovvero se uso ESP sarà cifrato mentre se uso AH avrà l'autenticazione dei dati che però saranno mandati in chiaro. In modalità trasporto non posso cifrare l'header in quanto i router hanno bisogno di vedere gli indirizzi IP per poter instradare il pacchetto.
- **tunnel mode:** viene aggiunto un header esterno al pacchetto originale e il payload include l'intero pacchetto originario mentre il nuovo header ha indirizzo IP sorgente diverso dal reale sorgente ovvero tipicamente l'IP del gateway che implementa IPSec. L'header più esterno deve essere trasmesso in chiaro per permettere ai router di instradare il traffico, mentre tutto ciò che viene dopo è payload dunque tutto il resto può essere cifrato, l'intero pacchetto incluso l'header originale.

Security Association SA

Security Association (SA)

- One-way sender-recipient relationship
- SA determines how packets are processed
- SA is defined by the triple <SPI, destination address, flag for whether it's AH or ESP>
- Each IPsec connection is viewed as one-way so two SAs required for a two-way conversation
 - Hence need for Security Parameter Index
- Security association (SA) defines
 - Protocol used (AH, ESP)
 - Mode (transport, tunnel)
 - Encryption or hashing algorithm
 - Negotiated keys and key lifetimes
 - Lifetime of this SA
 - ... plus other info



Definisce una relazione **UNIDIREZIONALE** tra sender e recipient e determina come vengono processati i pacchetti associati a quella comunicazione definendo se vengono usati AH o ESP e quali algoritmi utilizzare.

Una security association è definita da 3 parametri. Quelli importanti sono 2: SPI, il **security parameter index** ovvero l'identificativo di quella SA specifica, l'indirizzo di destinazione e infine un flag che mi specifica se userò AH o ESP.

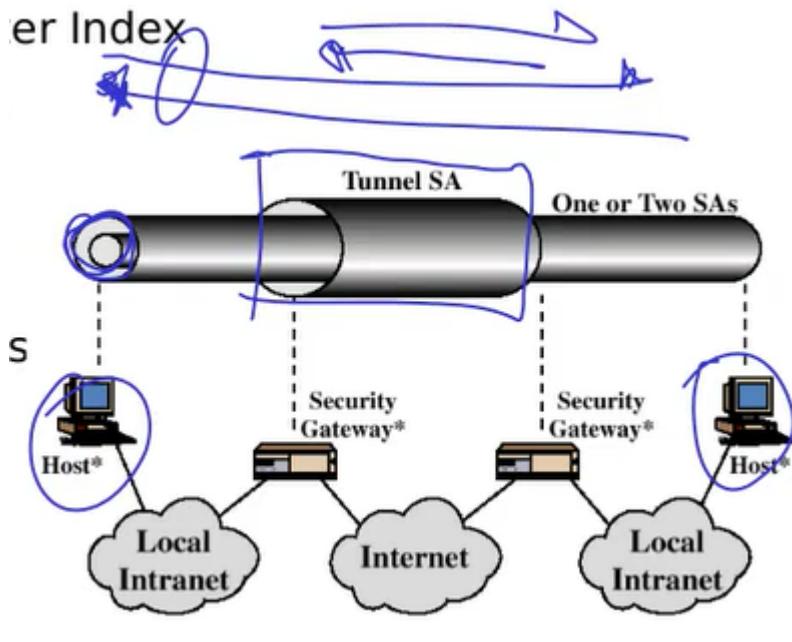
Se ho necessità di instaurare uno scambio di dati bidirezionale avrò necessità di instaurare due SA tra gli stessi host, una per ciascuna direzione di comunicazione.

Dunque ogni SA definisce:

- protocollo usato (AH o ESP)
- la modalità (transporto o tunnel)
- gli algoritmi di encryption o hashing
- le chiavi negoziate e il loro lifetime
- il lifetime della specifica SA

Si potrebbe inoltre creare una SA tra due end-systems che si trovano però già in reti protette da IPsec in modalità tunnel, dunque avremo 4 SA relative allo scambio di dati fra i due host remoti (2 SA fra gli

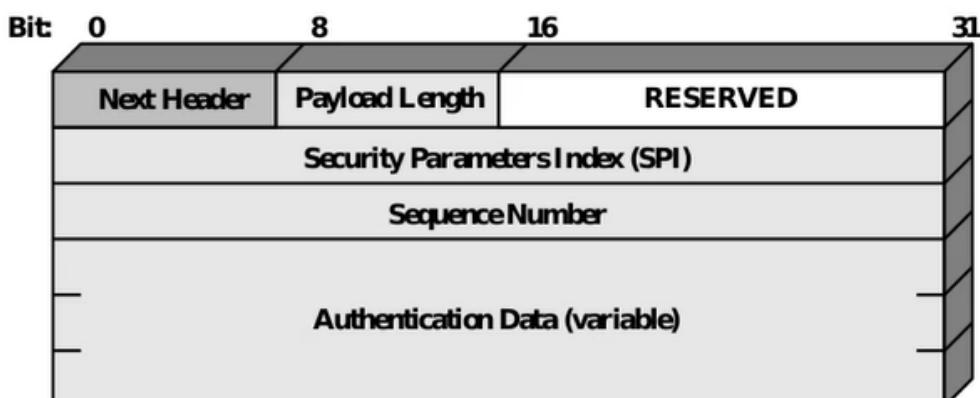
HOST e 2 fra i GATEWAY):



Authentication Header (AH)

Authentication header

- Sender authentication
- Integrity for packet contents and IP header
- Sender and receiver must share a secret key
 - This key is used in HMAC computation
 - The key is set up by IKE key establishment protocol and recorded in the SA
 - SA also records protocol being used (AH) and mode (transport or tunnel) plus hashing algorithm used

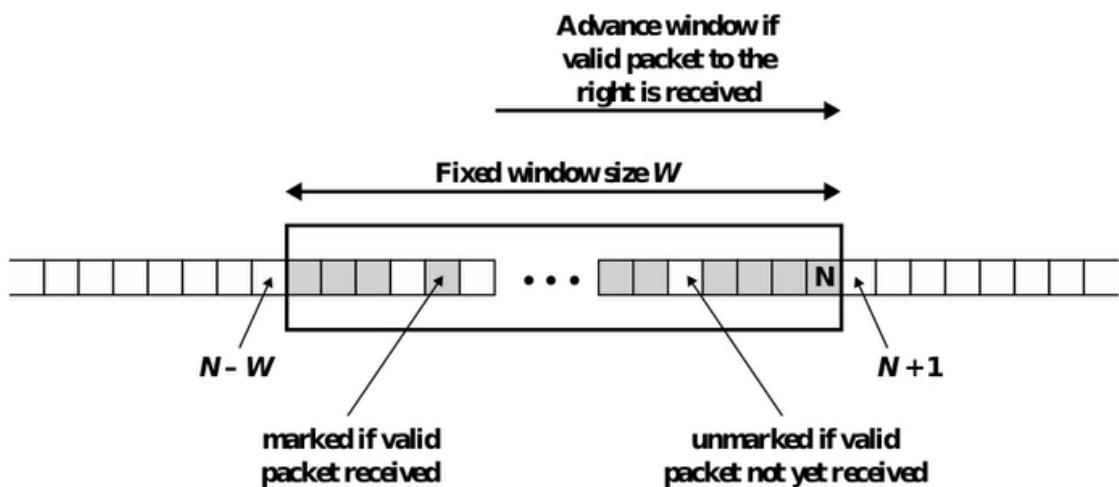


L'authentication header fornisce l'autenticazione del mittente e il servizio di integrità relativamente al pacchetto e all'header IP almeno in modalità transport.

- I 2 end-system dovranno condividere una chiave segreta tramite IKE tipicamente usata per il calcolo della **HMAC**.
- Il **Next Header** è simile al campo protocol di IPv4 dunque mi dice quale tipo di header troverò subito dopo (un header TCP, un header UDP oppure un ulteriore header IP).
- Il payload conterrà i dati.
- RESERVED contiene alcuni bit per implementazioni future del protocollo
- L'SPI individua la SA, vuol dire che è un informazione che il *mittente scrive* e il *destinatario legge* in modo da capire come trattare il pacchetto. Ci saranno alcuni byte che dipenderanno dall'algoritmo usato per fornire autenticazione.
- Il numero di sequenza che viene usato per la protezione anti-replay, ovvero la capacità di scartare pacchetti già ricevuti:

AH: anti-replay

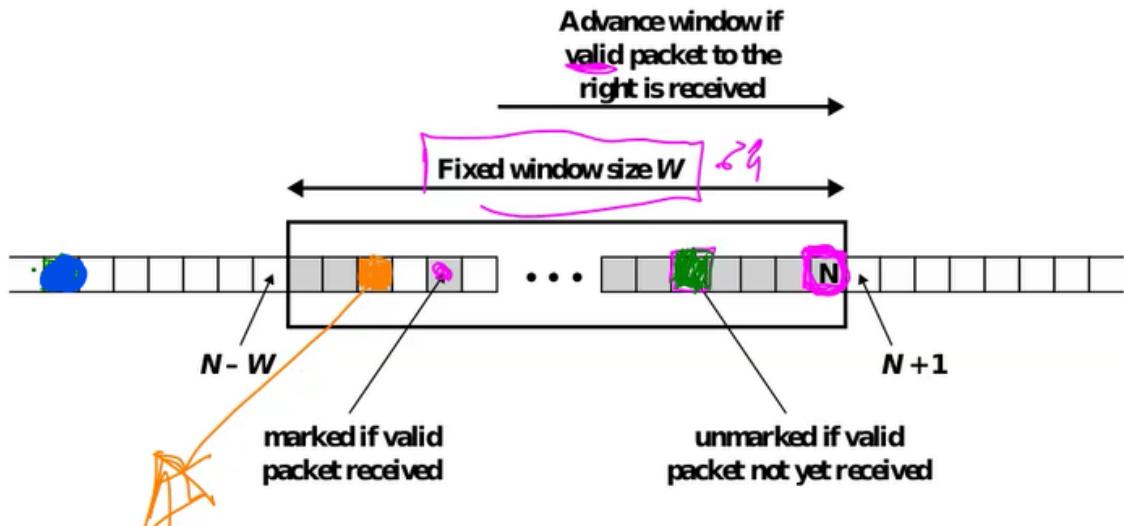
- When SA is established, sender initializes 32-bit counter to 0, increments by 1 for each packet
 - If wraps around $2^{32}-1$, new SA must be established
- Recipient maintains a sliding 64-bit window



Quando viene stabilita una SA tra due host, il mittente inizializza un contatore a 0 che viene incrementato per ogni pacchetto inviato. Quando raggiungo la fine del contatore devo stabilire una nuova SA e negoziare nuove chiavi. Il mittente inserisce questo numero su 32 bit nell'header AH (uguale in ESP), il ricevitore mantiene una **finestra mobile** che avanza ogni volta che arriva un pacchetto valido con un numero di sequenza più elevato. Supponendo che **N** sia il numero di sequenza più alto che ha ricevuto. Il pacchetto è valido se controllando il MAC verifica che sia corretto e non è stato modificato. Supponendo che tale pacchetto **N** sia l'ultimo pacchetto non in senso temporale ma come pacchetto con numero più elevato ed ho una finestra di dimensione fissa (ad esempio **W** = 64).

Il ricevitore considera gli ultimi **W** pacchetti e marca i corrispondenti numeri di sequenza se ha

ricevuto un pacchetto valido con quel numero di sequenza (quelli in grigio sono validi). Quelli in bianco non sono marcati in quanto non ha ancora ricevuto un pacchetto valido con quel particolare numero di sequenza. Supponiamo che mi arrivi di nuovo un pacchetto con il numero di sequenza in arancione:

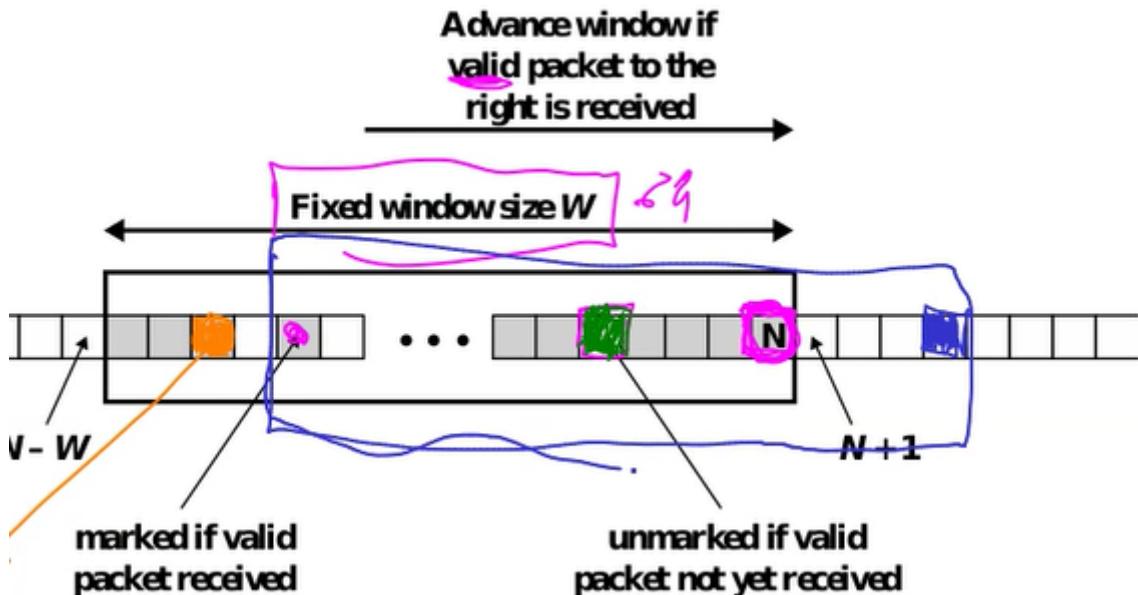


quel pacchetto non mi serve in quanto lo avevo già quindi IPSec lo ha già passato al livello superiore.

Se invece mi arriva il pacchetto in verde che prima era bianco, controllo che il codice MAC sia corretto, se sì marco il pacchetto come ricevuto.

Se invece mi arriva un pacchetto con il sequence number in blu oppure un qualsiasi pacchetto il cui codice MAC non è corretto, questo viene scartato.

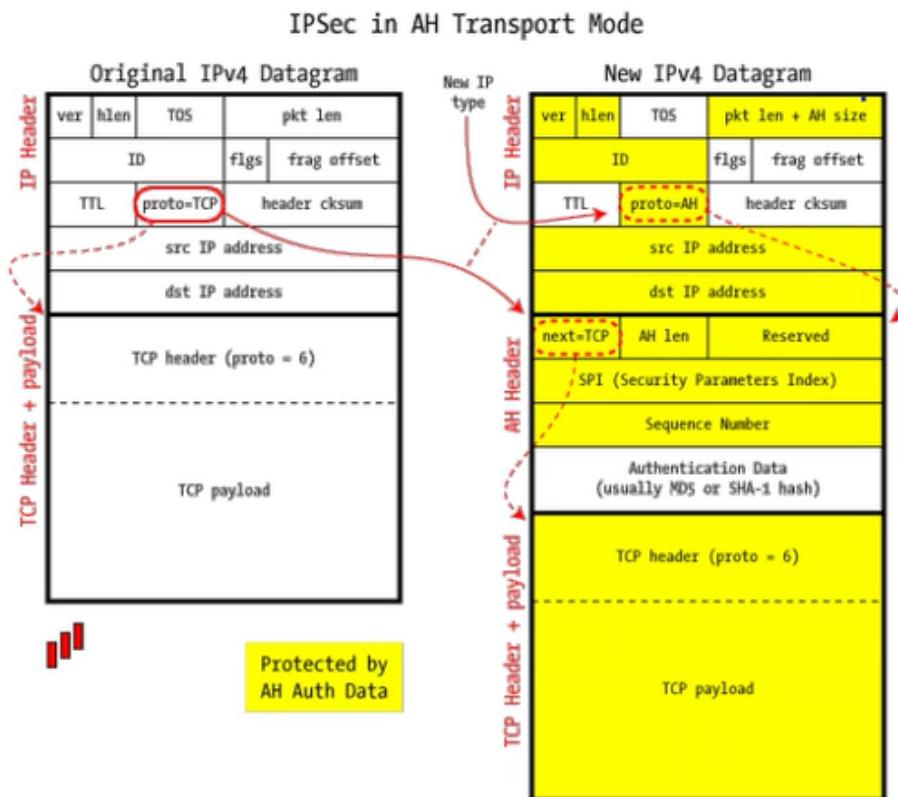
Se ricevo un pacchetto in avanti (in viola):



con un numero di sequenza maggiore di N , dopo averne verificato il MAC, mi avanza la finestra di 4 unità ed inizio a controllare che non arrivino duplicati all'interno della nuova finestra slittata in avanti di 4 sequence number.

AH Transport Mode

AH: transport mode



in questo esempio abbiamo un pacchetto IPv4 con header e payload TCP.

AH in modalità trasporto avrà un header IPv4, l'header AH e infine il payload. L'header più esterno è uguale a quello originario, chiaramente cambia la lunghezza in quanto devo tenere conto anche dell'inserimento dei campi di AH, il campo **protocol** di IP avrà non più valore TCP ma **AH**. Il **next = TCP** si è spostato all'interno dell'header AH.

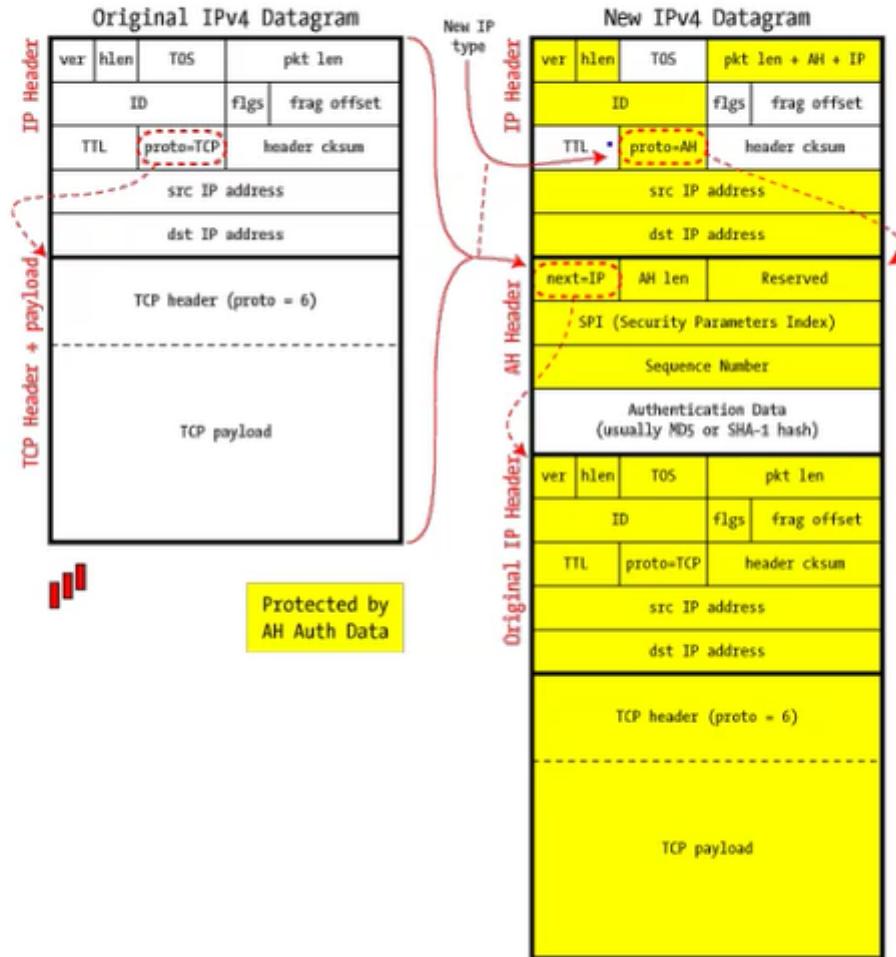
Le parti **evidenziate in giallo** sono quelle protette. I campi protetti sono sicuramente il payload, sicuramente anche alcuni campi dell'header AH e alcuni campi dell'header IPv4, MA NON TUTTI!

Quelli che vengono protetti sono quelli **IMMUTABILI** o **PREDICIBILI**: tutti quei campi che o non vengono modificati ma che sono modificati ma possono essere predetti in base al tipo di collegamento. L'esempio tipico è l'indirizzo di destinazione (dst IP address), in quanto se faccio source routing, l'IP di destinazione finale sarà l'host destinatario mentre il primo destinatario ovvero quello presente nell'header del pacchetto sarà il primo router che dovrà andare ad attraversare. Non posso proteggere il TTL perché viene modificato al passaggio dei router e non conosco quanti router attraverserà il mio pacchetto, ne posso proteggere la checksum in quanto cambiando il TTL cambierà sicuramente, come anche il ToS.

AH Tunnel Mode

AH: tunnel mode

IPSec in AH Tunnel Mode



Gli indirizzi dell'IP header in modalità tunnel saranno diversi rispetto agli indirizzi del pacchetto IP originale, mentre gli indirizzi IP sorgente e destinazione del payload di AH Tunnel Mode sono i medesimi presenti nel pacchetto IP originale.

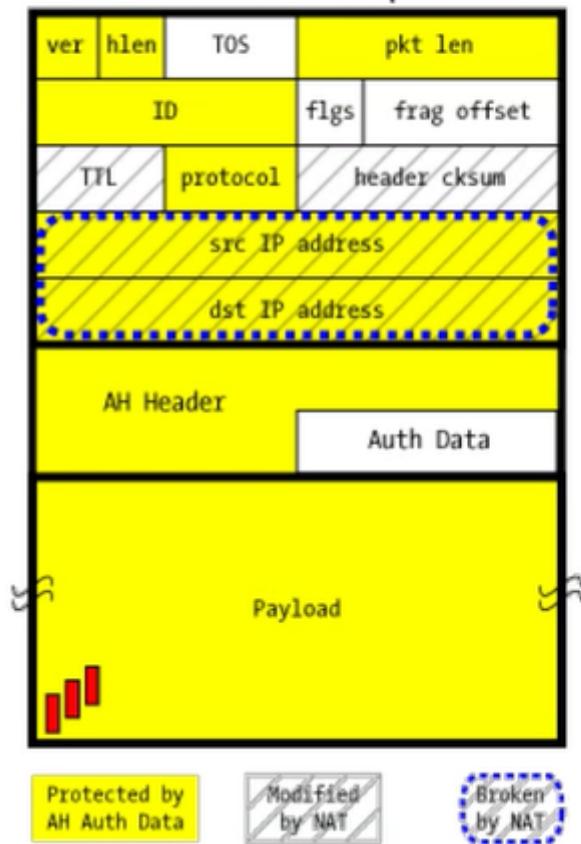
Il campo protocol di IP header mi dice che quello che segue è AH, mentre all'interno dell'AH header avrò invece IP. Il pacchetto IP originario è incapsulato nell'AH payload.

In questo caso viene autenticato l'intero pacchetto originario. Vengono autenticati anche i campi non modificabili o predicibili dell'header IP più esterno.

❓ E' un bene cercare di proteggere il più possibile o no?

AH & NAT

AH and NAT: Incompatible



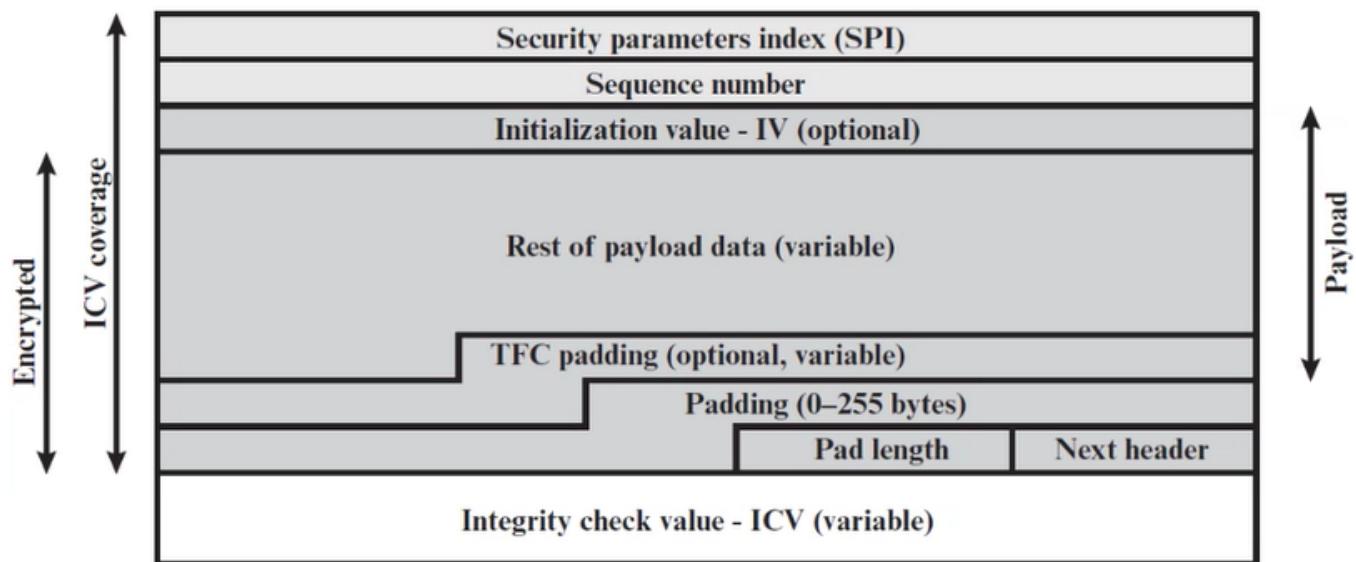
Nel caso del NAT questo sarebbe un problema in quanto gli indirizzi IP vengono modificati, però l'host che calcola la checksum crittografica li calcola con gli indirizzi originali, quindi poi passando per il NAT vengono cambiati, e quando il ricevitore ricalcola il checksum con gli indirizzi cambiati il checksum gli sembrerà errato.

20/05/2023

Encapsulating Security Payload (ESP)

Encapsulating security payload

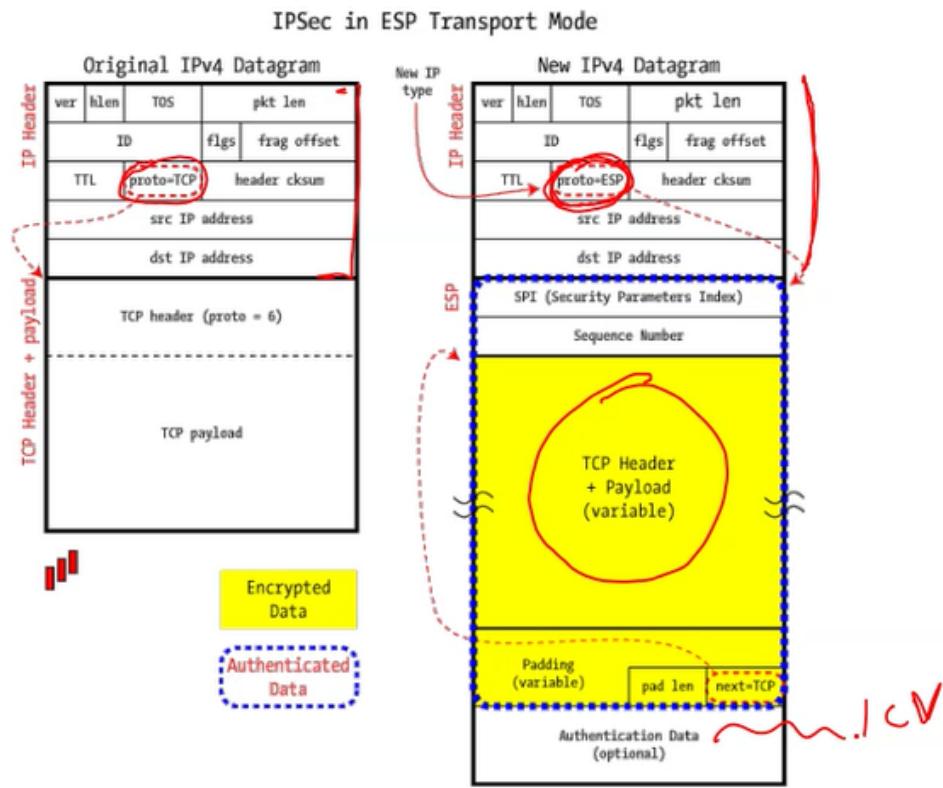
- Adds new header and trailer fields to packet



- L'SPI serve per individuare la Security Association
- Il Sequence Number serve per difendersi dal replay attack
- Il payload potrà essere o il payload del pacchetto IP originario o l'intero pacchetto IP originario a seconda se usiamo ESP in modalità transport o tunnel. Per qualche motivo si intendeva per payload anche l'IV che viene trasmesso al ricevitore nel caso in cui si usi ad esempio AES-CBC
- Il Pad lenght indica la lunghezza del padding
- L'**TFC padding** (traffic flow confidentiality) consente di evitare che un opposente capisca quale traffico sto effettuando facendo analisi statistica del traffico. L'opponente potrebbe essere anche il Service Provider che non vuole che venga usata la sua infrastruttura per particolari traffici di rete
- Il padding normale è necessario per allineare la struttura. Pensando di operare con un cifrario a blocchi di dimensione fissata, il payload dovrà avere una lunghezza multipla della lunghezza dei blocchi
- Il Next Header mi dà informazioni su quello che si trova nel Payload ovvero che si trova prima del Next Header stesso.
- L'ICV è il codice MAC per autenticazione e integrità di quello che lo precede. L'ICV è calcolato su tutta la versione cifrata del payload. E' stato fatto in questo modo in modo tale da controllare subito che l'ICV sia corretto ed evitare di effettuare un processing inutile nel caso di messaggi non integri o non autenticati.

ESP Transport Mode

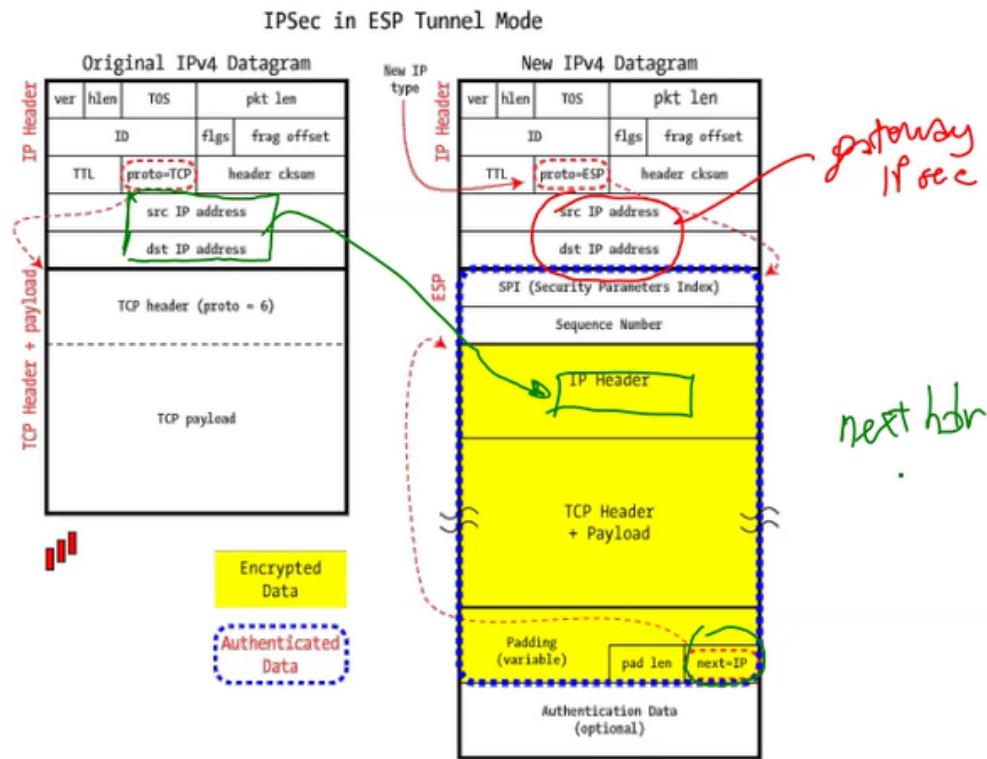
ESP: transport mode



Il campo protocollo in questo caso è `proto = ESP`. L'header ESP avrà un `next = TCP` in quanto il payload sarà un pacchetto TCP. Infine l'Authentication Data sarebbe l'ICV il quale potrebbe non esserci in quanto ESP fornisce encryption e/o authentication. In questo caso in cui abbiamo sia autenticazione che privacy, l'autenticazione copre campi ulteriori rispetto alla parte cifrata.

ESP Tunnel Mode

ESP: tunnel mode



Il gateway di frontiera userà un header con indirizzi IP esterni tipicamente diversi rispetto a quelli del pacchetto originario, infatti il src e dest IP address dell'IP header nel pacchetto ESP saranno quelli dei gateway che implementano il tunnel IPSec.

Il campo protocol esattamente come prima è ESP mentre il campo next-header di ESP sarà settato ad IP.

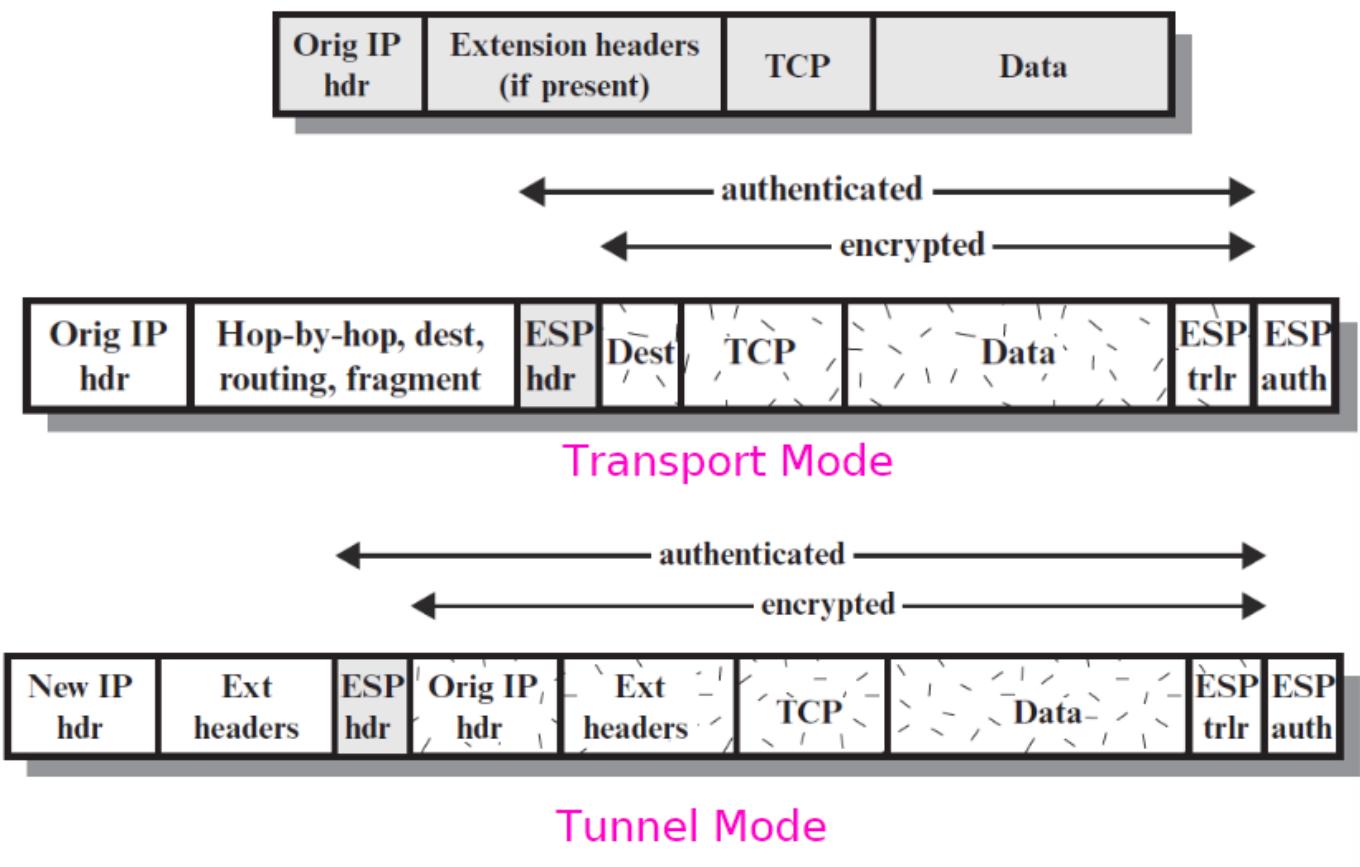
⚠ Nei pacchetti IPSec il campo che indica il tipo di pacchetto presente nel payload è detto *next-header* questo perché IPSec è stato sviluppato nell'ambito di IPv6, il quale usa la terminologia *next-header* per indicare il tipo di pacchetto del livello protocollare superiore, in modo analogo ad IPv4 che invece usa il termine *protocol*

ESP & IPv6

IPv6 ha un header di lunghezza fissa ma esistono degli extension header aggiuntivi che permettono di implementare funzioni aggiuntive come proprio ESP. Rispetto ad IPv4 cambiano proprio questi extension-header che possono essere visti come gli equivalenti delle opzioni di IPv4 però con maggiori gradi di libertà. Queste opzioni sono tipicamente spezzate in diverse parti come quelle per il routing, l'hop-by-hop, per la frammentazione che devono per forza essere in chiaro. Altre opzioni come quelle per il destinatario possono essere implementate al di fuori di ESP in chiaro oppure incapsulate in ESP

e dunque visibili solo all'endpoint finale in base alla necessità:

ESP & IPv6

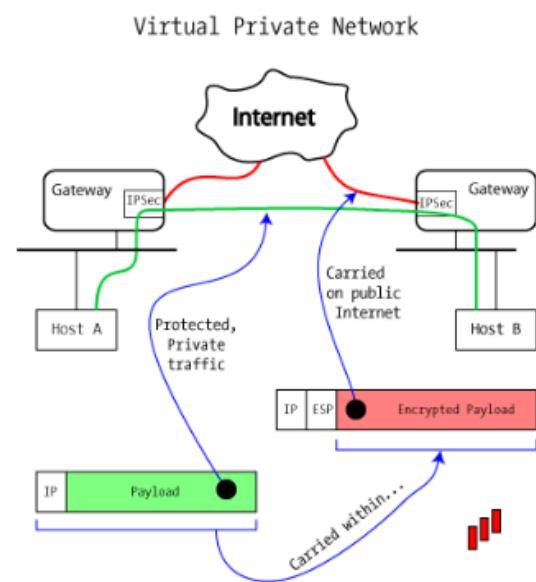


In modalità tunnel cambia che aggiungo un header più esterno con eventuali extension header se necessari che sono diversi rispetto all'extension header del pacchetto IP originario.

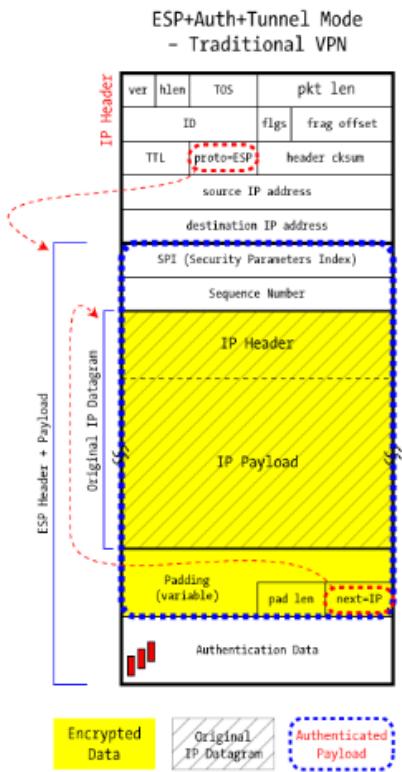
L'impiego classico di ESP è quello di realizzare le VPN anche se a volte viene usato anche il TLS:

IPSec: VPNs

- ESP is often used to implement a VPN (Virtual Private Network)
 - Packets go from internal network to a gateway with TCP / IP headers for address in another network
- Entire packet hidden by encryption
 - Including original headers so destination addresses are hidden
 - Receiving gateway decrypts packet and forwards original IP packet to receiving address in the network that it protects
- This is known as a **VPN tunnel**
 - Secure communication between parts of the same organization over public untrusted Internet



IPSec: VPN



IPSec: Services

	AH	ESP	ESP (+Auth)
Access Control	X	X	X
Datagram Integrity	X		X
Data Originator Authentication	X		X
Anti-Replay	X	X	X
Confidentiality		X	X
Flow Level Confidentiality		X	X

In tutti i casi ho un controllo degli accessi, AH fornisce integrità e autenticazione che se vogliamo sono

due servizi diversi che però vengono svolti usando un unica funzione crittografica e anche Anti-Replay. ESP fornisce sia Anti-Replay che confidenzialità a livello di dati e di flusso ma può fornire anche autenticazione della sorgente e integrità a livello di datagramma se combinato con algoritmi di auth. **Se vogliamo IPSec svolge anche delle funzioni di firewall in quanto i pacchetti potrebbero essere scartati prima di passarli ai livelli protocollari superiori grazie alla funzionalità di Access Control.**

Algoritmi crittografici

Cryptographic Algorithms (RFC 7321)

- RFC 7321 specifies the cryptographic algorithms that MUST be implemented, and provides guidance about ones that SHOULD or SHOULD NOT be implemented
- All AES modes are for **128-bit AES**
 - 192-bit and 256-bit AES MAY be supported for those modes
- Providing both confidentiality and authentication offers the best security
- If confidentiality is not needed, providing authentication can still be useful
 - The IPsec community prefers ESP with NULL encryption over AH
 - AH is still required in some protocols and operational environments when there are security-sensitive options in the IP header, such as source routing headers; ESP inherently cannot protect those IP options
- **Confidentiality without authentication is not effective and therefore SHOULD NOT be used**

La RFC 7321 specifica gli algoritmi da implementare fornendo delle linee guida su algoritmi da implementare o meno. In particolare per AES era obbligatoria implementare la variante a 128 bit mentre le chiavi a lunghezza superiore possono essere supportati. Se un algoritmo non è citato esplicitamente viene considerato MAY ovvero tutto ciò che non è vietato è possibile implementarlo, mentre tutto ciò che è MUST deve essere implementato al fine di avere un software compatibile con i protocolli in esame nella specifica RFC.

Se la confidenzialità non è necessaria ma solo autenticazione allora potrei usare AH oppure ESP senza cifratura. In generale la comunità IPSec preferisce ESP con cifratura nulla. In alcuni casi se voglio proteggere anche dei campi dell'header devo usare AH, in quanto ESP non protegge queste opzioni del pacchetto IP. **In generale però la confidenzialità senza autenticazione non è efficace e pertanto non dovrebbe essere usata.**

La seguente tabella indica per ciascun algoritmo se deve essere implementato o meno.
Lo SHOULD+ è un rafforzativo l'idea ovvero è quella di lasciare per ora SHOULD ma sapendo che in una prossima RFC è possibile che si passerà a un livello superiore (come MUST). Per AH non avrebbe senso non fare autenticazione in quanto AH non fa altro rispetto all'autenticazione.

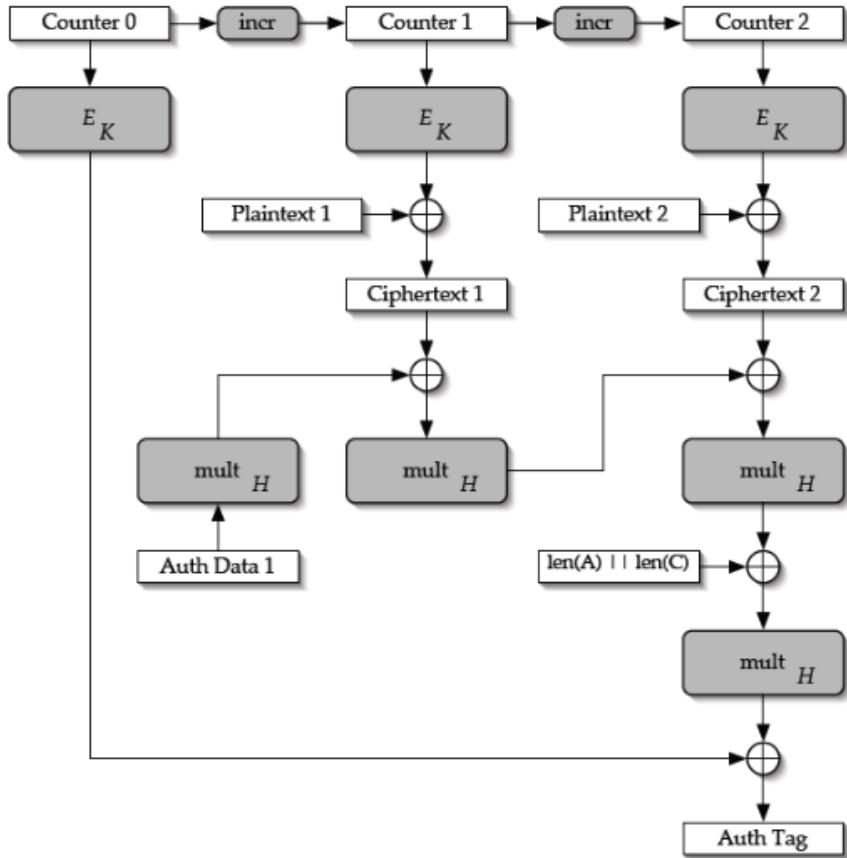
Cryptographic Algorithms

- **ESP Authenticated Encryption**
 - SHOULD+ AES-GCM with a 16 octet ICV
 - MAY AES-CCM (Counter with CBC-MAC mode)
- **ESP Encryption Algorithms**
 - MUST NULL (preferred over AH due to NAT traversal)
 - MUST AES-CBC
 - MAY AES-CTR
 - MAY 3DES-CBC
 - MUST NOT DES-CBC
- **ESP Authentication Algorithms**
 - MUST HMAC-SHA1-96
 - SHOULD+ AES-GMAC with AES-128
 - SHOULD+ AES-XCBC-MAC-96
 - MAY NULL
- The requirements for **AH** are the same, except NULL

RFC 7321 (August 2014) - Cryptographic Algorithm Implementation
Requirements and Usage Guidance for Encapsulating Security Payload (ESP)
and Authentication Header (AH)

Galois Counter Mode

Galois/Counter Mode(GCM)



Sono presenti degli Authorization Data e Authorization Tag dunque questo algoritmo permette di cifrare e autenticare anche dati diversi.

L'evoluzione che si è avuta dal punto di vista degli algoritmi va in 2 direzioni opposte:

Algorithms Evolution (RFC 8221)

- RFC 8221 Cryptographic Algorithm Implementation Requirements and Usage Guidance for ESP and AH (October 2017)
 - To enable ESP and AH to benefit from cryptography that is up to date while making IPsec interoperable
 - The algorithm implementation requirements and usage guidance may need to change over time to adapt to the changing world
 - The selection of mandatory-to-implement algorithms was removed from the main IKEv2 specification [RFC7296] and placed in a separate document
- Algorithms need to be suitable for a wide variety of CPU architectures and device deployments ranging from high-end bulk encryption devices to small, low-power IoT devices
 - Requirement levels that are marked as "IoT" apply to IoT devices and to server-side implementations that might presumably need to interoperate with them, including any general purpose VPN gateways
- IPsec sessions may have very long lifetime and carry multiple packets, so there is a need to move to **256-bit keys** in the long term
 - The requirement level for 128-bit keys and 256-bit keys is MUST
 - 192-bit keys remain at the MAY level

Per AES è diventato obbligatorio usare ad esempio chiavi a 256 bit secondo l'RFC 8221.

Algorithms Evolution

- **Encryption Must Be Authenticated:** Encryption without authentication is not effective and MUST NOT be used
- IPsec offers three ways to provide both encryption and authentication
 - ESP with an **Authenticated Encryption with Associated Data (AEAD) cipher**
 - ESP with a non-AEAD cipher + authentication
 - ESP with a non-AEAD cipher + AH with authentication
- **Changes from RFC 7321**
 - The status for **256-bit keys** has been raised from MAY to MUST

Algorithm	RFC 7321	RFC 8221
ENCR_AES_GCM_16	SHOULD+	MUST
ENCR_AES_CCM_8	MAY	SHOULD (IoT)
ENCR_AES_CTR	MAY	MAY(<i>not mentioned</i>)
ENCR_3DES	MAY	SHOULD NOT
AUTH_HMAC_SHA1_96	MUST	MUST-
AUTH_AES_128_GMAC	SHOULD+	MAY
AUTH_NONE	MAY	MUST / MUST NOT

Infine ad Aprile 2023 è stata rilasciata l'RFC 9395 che specifica sia per IKE che IPsec quali algoritmi

sono stati deprecati:

RFC 9395 (April 2023)

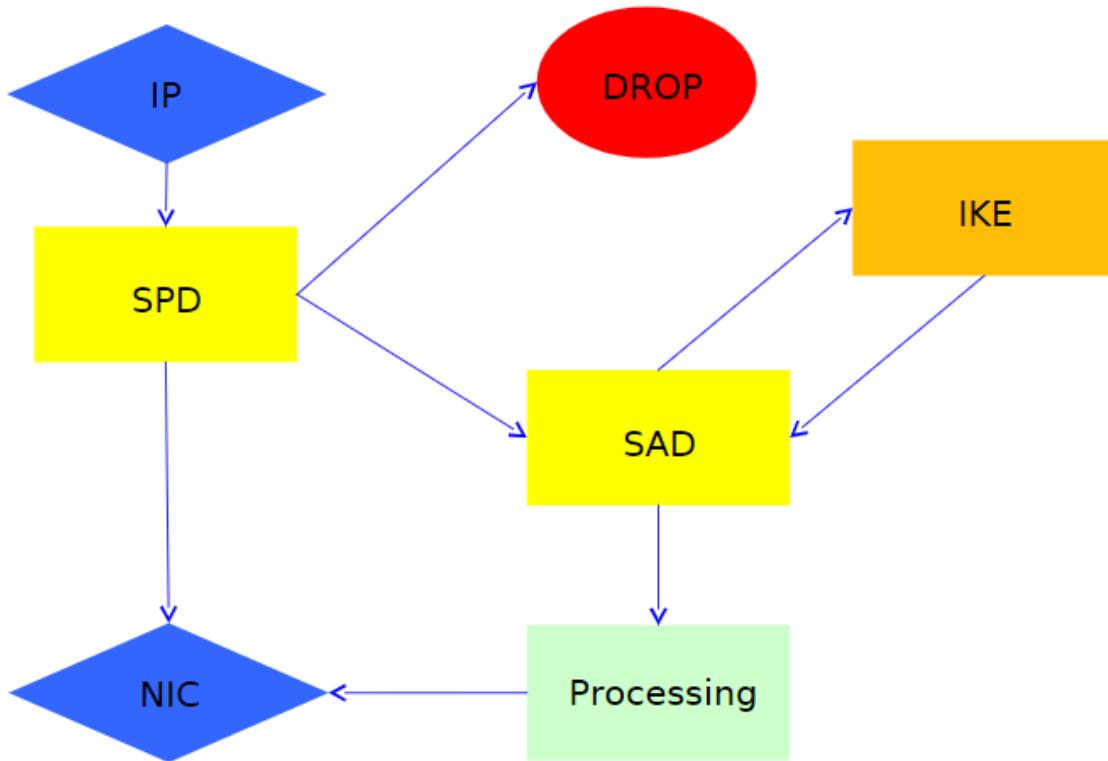
- RFC 9395 - Deprecation of the Internet Key Exchange Version 1 (IKEv1) Protocol and Obsoleted Algorithms
- Internet Key Exchange Version 1 (IKEv1) has been deprecated, and RFCs 2407, 2408, and 2409 have been moved to Historic status
- Since IKEv2 replaced IKEv1 over 15 years ago, IKEv2 has now seen wide deployment, and it provides a full replacement for all IKEv1 functionality
- No new modifications or new algorithms have been accepted for IKEv1 for at least a decade
- IKEv2 addresses various issues present in IKEv1, such as IKEv1 being vulnerable to amplification attacks
- This document updates RFCs 8221 and 8247 to reflect the usage guidelines of old algorithms that are associated with IKEv1 and are not specified or commonly implemented for IKEv2

Deprecated Obsolete Algorithms:

- Encryption Algorithms: RC5, IDEA, CAST, Blowfish, and the unspecified 3IDEA, ENCR_DES_IV64, and ENCR_DES_IV32
- PRF Algorithms: the unspecified PRF_HMAC_TIGER
- Integrity Algorithms: HMAC-MD5-128
- Diffie-Hellman groups: none

Architettura di IPSec

IPSec: Architecture



Supponiamo che io abbia un pacchetto IP, nel **Security Policy Database** è definito in modo simile a un firewall cosa fare con ciascun pacchetto ricevuto. Se viene ricevuto un pacchetto IP normale viene passato alla scheda di rete (NIC). Se nel SPD è indicata l'azione DROP allora il pacchetto deve essere scartato, altrimenti se dice di usare IPSec, bisogna usare un **Security Association Database** al cui interno troverò le regole su cosa fare con quel pacchetto, per poi passarlo infine alla scheda di rete. Se non dovessero esserci entry nel SAD bisognerà richiamare il protocollo IKE per capire come comportarsi.

Security Policy Database

Security Policy database

SPD														
Src IP	PFP	Dst IP	PFP	Proto	PFP	Src Port	PFP	Dst Port	PFP	Action	Proto/ Mode	Other	P	
192.168.0.1	1	192.168.1.1	1	TCP	1	*	0	*	0	PROTECT	ESP/Tunnel	AES-256		
192.168.0.0/24	1	192.168.1.0/24	1	TCP	1	*	0	*	0	BYPASS				

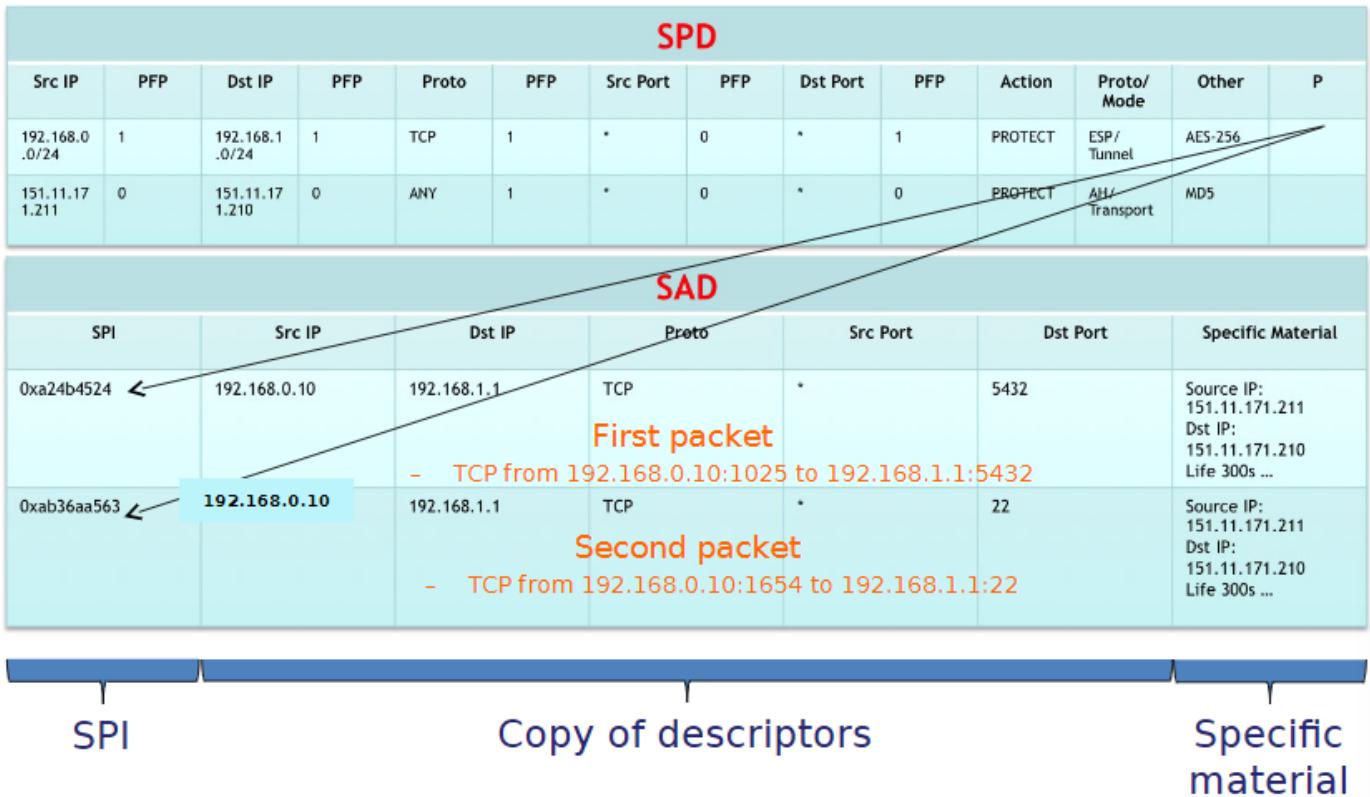
Rule part:

- Traffic descriptors
- Populate From Packet (PFP) bit

Action Pointer

Avremo una parte che individua i pacchetti (IP sorgente e destinazione, porta sorgente e destinatario e il protocollo usato) anche detti **Traffic Descriptors**. Per comprendere l'utilità dei campi **P** e **PFP** dobbiamo considerare il SAD:

Security Association DB



Il puntatore **P** dell'SPD punta alle regole che mi dicono come fare le singole operazioni contenute nel SAD. Il puntatore può puntare contemporaneamente a diverse regole del SAD.

Supponiamo di dover trasmettere questi pacchetti indicati in arancione nella figura e capire come questi pacchetti vadano a modificare il contenuto del SAD. **Essendo numeri di porta diversa si tratta di Security Association Rule diverse anche se appartenenti allo stesso flusso sorgente-destinazione.** Capiamo allora come il PFP interviene in quanto pur essendo lo stesso flusso si tratta sicuramente di applicazioni diverse e dunque possono richiedere tipi di protezione diversi.

Supponiamo che inizialmente il SAD sia vuoto. Quando il primo pacchetto mi arriva, interrogo IKE e creo la SA che avrà un suo identificativo e una copia dei descrittori del traffico. Per il primo pacchetto ricevuto:

TCP from 192.168.0.10:1025 to 192.168.1.1:5432

dato che nell'SPD associato al **Src IP** 192.168.0.0/24 c'è un PFP = 1 significa che andrà creata una SA per ogni pacchetto ricevuto sulla base del diverso IP sorgente mentre dato che il PFP della Src Port è 0 non andrà creata una SA diversa sulla base della diversa porta.

Il secondo pacchetto:

TCP from 192.168.0.10:1654 to 192.168.1.1:22

potrà usare la stessa SA del primo pacchetto?

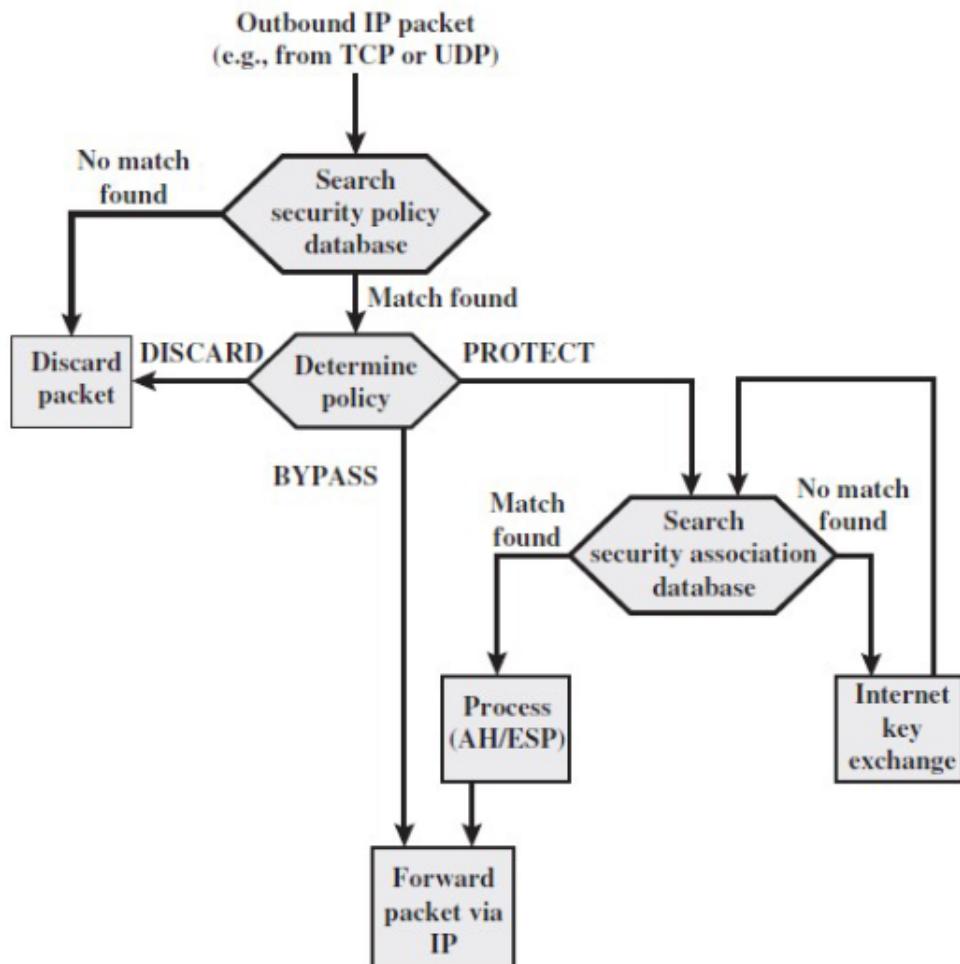
Dato che in questo caso la porta di destinazione è diversa, la SA association di prima andava bene per porta di destinazione 5432.

A seconda della necessità di voler proteggere diversamente flussi differenti andrà settato il PFP all'occorrenza.

A seconda del valore di PFP devo negoziare o meno tramite IKE le SA.

Volendo proteggere in modo diverso i vari flussi, il PFP permette di differenziare la gestione del traffico IPSec sulla base del traffico effettuato.

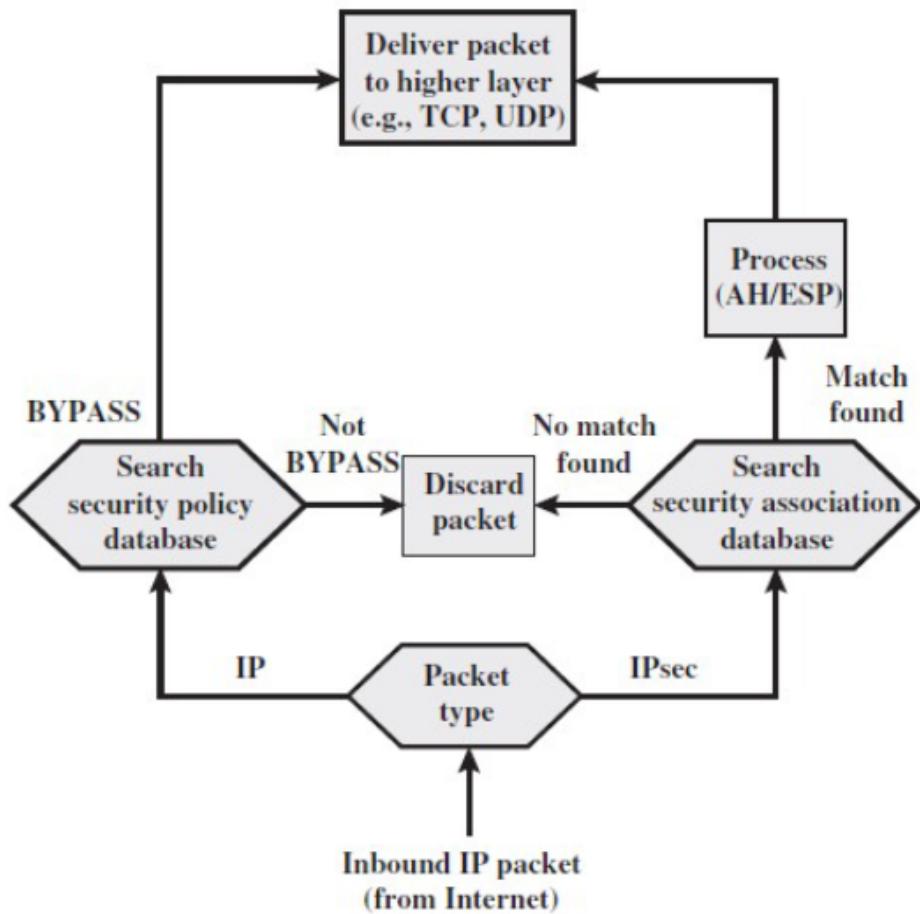
Outbound packets



Supponiamo che il livello di trasporto abbia passato a livello di rete un segmento. Se nell'SPD non esiste alcuna regola che corrisponde a quel flusso di pacchetti allora scarto quel pacchetto. Se trovo un match devo capire cosa fare con quel pacchetto. Se la policy è **BYPASS** il pacchetto è un normale pacchetto IP e viene passato direttamente alla scheda di rete. Se la policy è **PROTECT** devo cercare nel SAD se trovo la regola corrispondente posso processare quel pacchetto e poi passarlo alla scheda di rete per trasmetterlo. Se non trovo nulla nel SAD posso chiedere a IKE di fornirmi i dati e procedere come nel caso precedente.

Se arriva un pacchetto invece:

Inbound packets



Controllo se è IP, in tal caso bisogna guardare l'SPD per capire cosa fare con il traffico in ingresso, se dovrebbe essere protetto e invece non lo è dovrà essere scartato. Se si tratta invece di un pacchetto IPsec vado a cercare la SA corrispondente, se non la trovo vuol dire che non è un pacchetto già negoziato e dunque lo scarto altrimenti dovrà essere processato e poi passato alla NIC.

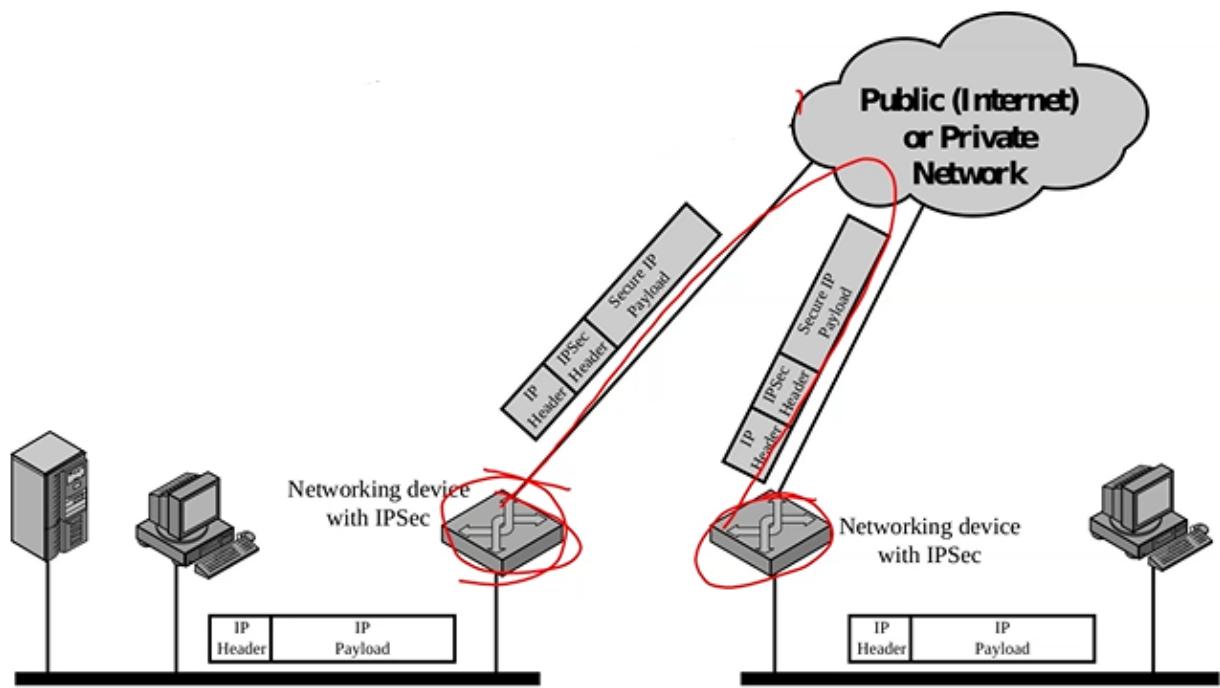
Internet Key Exchange

Internet Key Exchange (IKE)

- IKE performs mutual authentication between two parties and establish an IKE SA that includes shared secret information that can be used to efficiently establish SAs for ESP or AH and a set of cryptographic algorithms
- IKEv1
 - Oakley Key Determination Protocol: it is a key exchange protocol based on the Diffie-Hellman algorithm, but providing added security. Oakley does not dictate specific formats
 - Internet Security Association and Key Management Protocol (ISAKMP): ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes
- IKEv2
 - Not backward compatible

Effettua una autenticazione mutua tra le 2 parti e crea una **IKE SA** che poi viene usata per creare le SAs di ESP o AH.

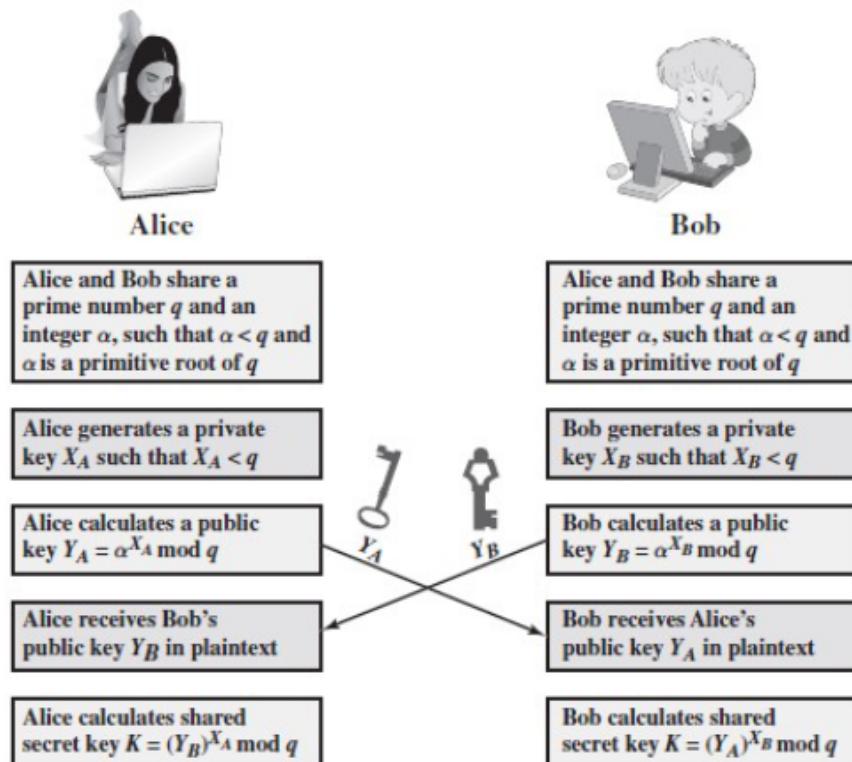
IPSec può essere usato anche senza IKE, ad esempio pensando di avere due reti, che gestiamo noi in quanto amministratori di rete, possiamo configurare le chiavi manualmente:



Il punto di partenza è Diffie-Hellmann:

Key Determination Protocol

- IKE key determination is a refinement of the Diffie-Hellman key exchange algorithm



Features of Diffie-Hellman

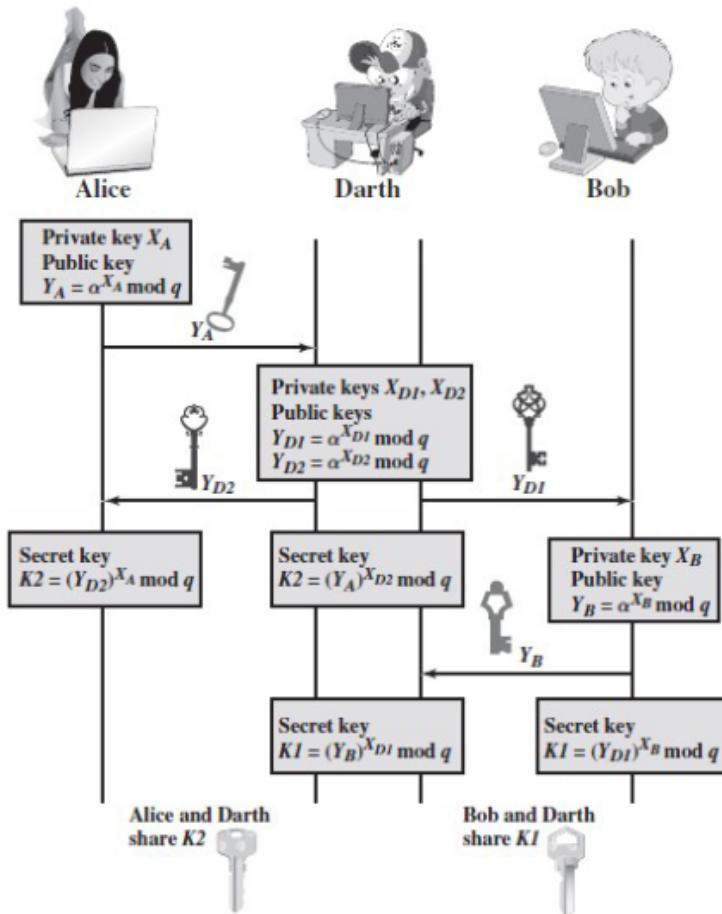
- The Diffie-Hellman algorithm has two attractive features
 - Secret keys are created only when needed
 - No pre-existing infrastructure (only an agreement on global parameters q and α)
- But also several weaknesses
 - No information about the identities of the parties
 - Vulnerability to a man-in-the-middle attack (see next slide)
 - Vulnerability to a clogging attack, in which an opponent requests a high number of keys
 - The algorithm is computationally intensive
- IKE key determination is designed to retain the advantages of Diffie-Hellman, while countering its weaknesses

I difetti di DH sono dovuti:

- alla possibilità di **Man In The Middle**: alla mancanza di autenticazione, in quanto ricevo il parametro da un certo indirizzo IP ma non ho alcun binding tra identità e pacchetti quindi sono

soggetto ad attacchi di tipo MITM

Man-in-the-middle attack



- **Clogging attack** (to clog = intasare): è un tipo di attacco DOS in cui costringo gli utenti a calcolare delle chiavi condivise segrete che però vuol dire effettuare dei calcoli computazionali abbastanza onerose

L'idea di IKE è quella di migliorare DH:

IKE key determination

- The IKE key determination algorithm is characterized by **five important features**
 - Use of cookies to thwart clogging attacks
 - Negotiation of a *group*
 - in essence, specifies the global parameters
 - Use of *nonces against* replay attacks
 - Exchange of Diffie-Hellman public key values
 - Authentication of the Diffie-Hellman exchange to thwart man-in-the-middle attacks
- i **gruppi**: chiavi più o meno lunghe ed eventualmente algoritmi in base al livello di sicurezza che voglio avere
- uso di **nonces** per difesa da attacchi di tipo replay
- **autenticazione** per difendersi da attacchi MITM

Cookies

Use of cookies

- Clogging attack
 - An opponent forges a legitimate source address and sends a public Diffie-Hellman key to the victim
 - The victim performs exponentiation to compute the secret key
- Cookie exchange
 - Each side sends a pseudorandom number in the initial message, which the other side acknowledges
 - This acknowledgement must be repeated in the first message of the Diffie-Hellman key exchange
 - If the source address was forged, the opponent gets no answer
- Basic requirements of cookie generation
 - It depends on the specific parties
 - The issuing entity will use some local secret information that cannot be deduced from any particular cookie
 - Generation and verification methods must be fast
- Recommended method for creating cookies
 - fast hash (eg., MD5) over the IP src and dest addresses, UDP src and dest port and a locally generated secret value

Viene inserito nel primo messaggio che mando un numero pseudocasuale (**cookie**) che sarà indirizzato solo verso la destinazione legittima e l'attaccante non potrà conoscerlo.

Uno dei modi classici per calcolare questo cookie consiste nel calcolare un hash veloce MD5 sulla base degli indirizzi IP dei due host e da qualche informazioni segreta.

IKEv2

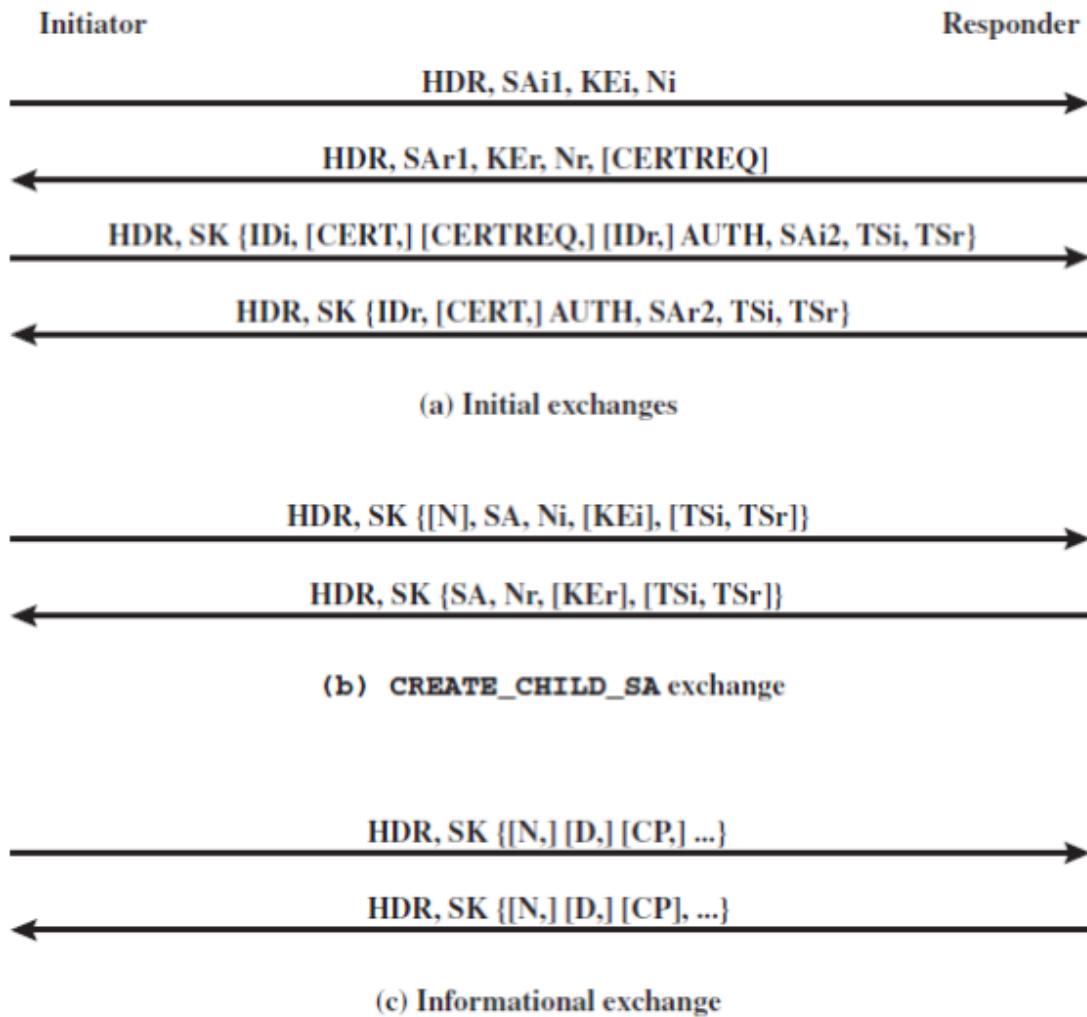
Sfrutta una comunicazione dei messaggi in "coppia" ovvero viene inviato il messaggio e si attende una risposta.

IKEv2 exchange

- IKE normally listens and sends on UDP port 500
 - IKE messages may also be received on UDP port 4500
 - An IPsec endpoint that discovers a NAT between it and its correspondent MUST send all subsequent traffic from port 4500
- Messages are exchanged in pairs
 - Request/response pair (or exchange)
- Messages have sequence number
 - It is responsibility of the initiator to ensure reliability
 - If the response is not received within a timeout, the requester needs to retransmit the request (or abandon the connection)
- Message size
 - IKEv2 messages are intended to be short, but...
 - contain structures with no hard upper bound on size (e.g., digital certificates)
 - IP fragmentation

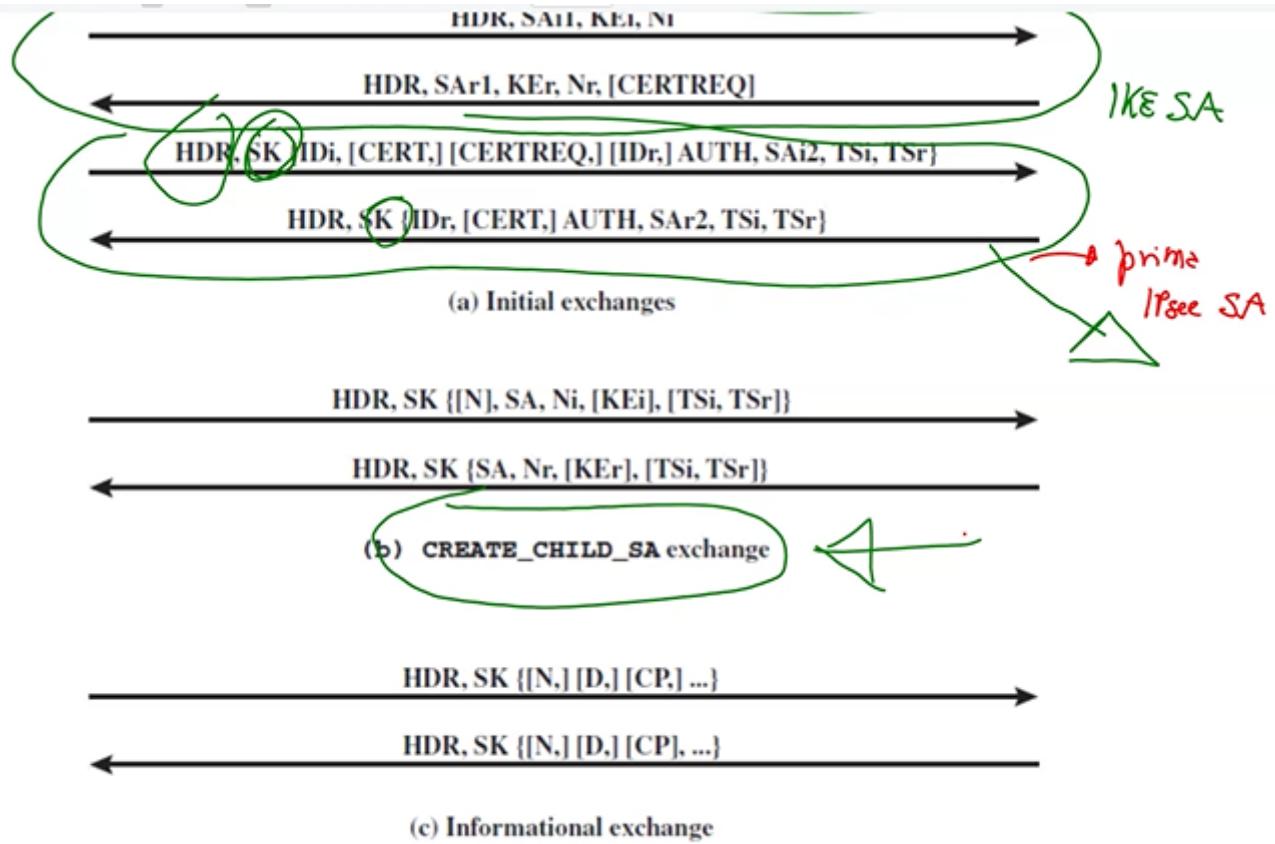
prevede uno scambio di messaggi diverso in cui inizialmente i primi messaggi sono in chiaro:

IKEv2 exchange

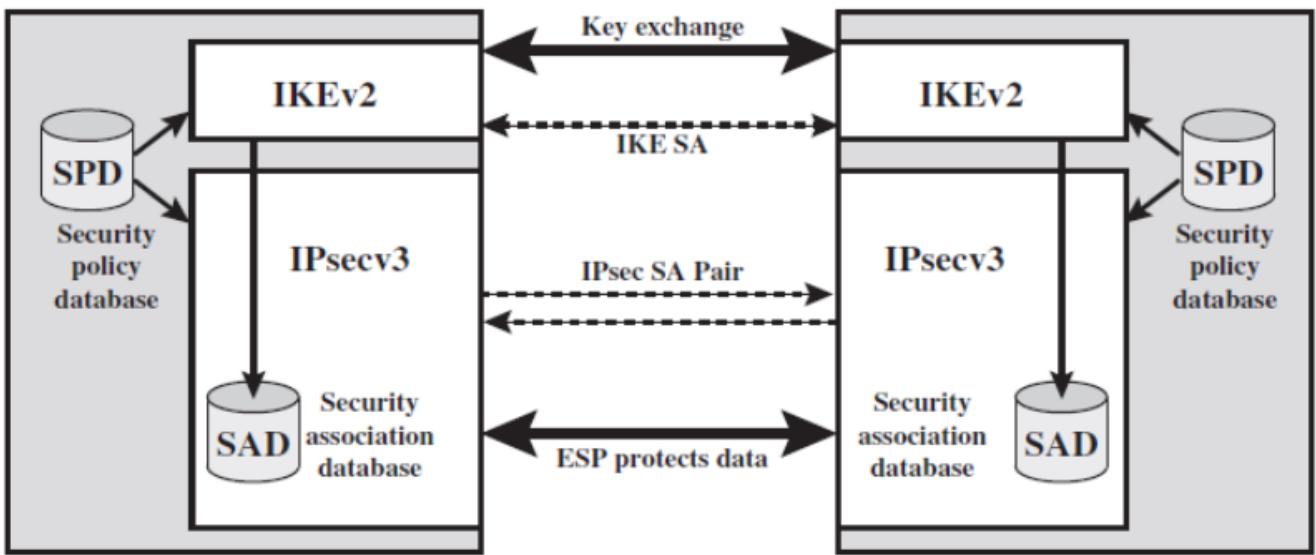


la seconda coppia di messaggi (in cui è indicato SK) sta per confidenzialità e autenticazione. L'header di IKE ha un suo formato (HDR). Dopo i primi 4 messaggi è possibile creare una SA. Dopo il primo scambio ho una IKE SA. Dopo il secondo scambio ho la prima IPsec SA. Infine uso la

CREATE_CHILD_SA quando le chiavi IKE vanno riaggiornate:



IKE SA

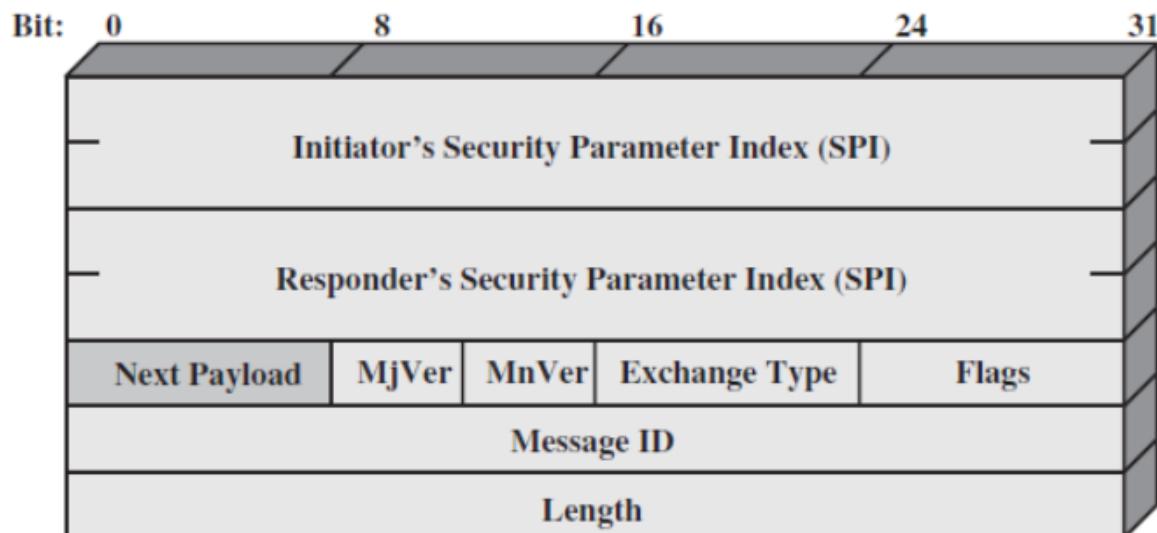


- IKE SA defines parameters for a secure channel between the peers
 - All subsequent IKE messages exchange are protected by encryption and message authentication

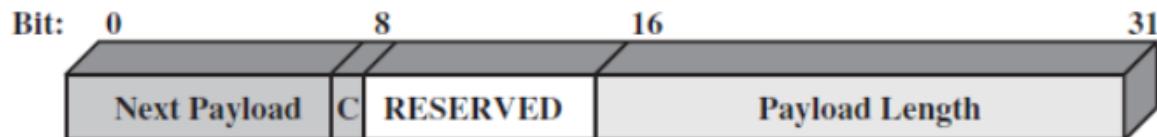
Se non trovo la regola nell'SPD chiamo IKE, il quale crea una IKE SA che è BIDIREZIONALE e servono per scambiarmi le chiavi e creare le SA di IPSec che invece sono unidirezionali.

L'header di IKE:

IKE Header format



(a) IKE header



(b) Generic Payload header

è su 64 bit .



Outline of the course

- Basics of Network Attacks
- Ethical Hacking
 - Introduction
 - Social Engineering
 - Scanning and Information Gathering
 - Vulnerability Scan
 - Network Attacks
 - Modern Attacks
- Hands-on Session

Network Security

Security services

Confidentiality



Security services: Confidentiality

Confidentiality has been defined by the International Organization for Standardization (ISO) in ISO-17799 as “ensuring that information is accessible only to those authorized to have access”

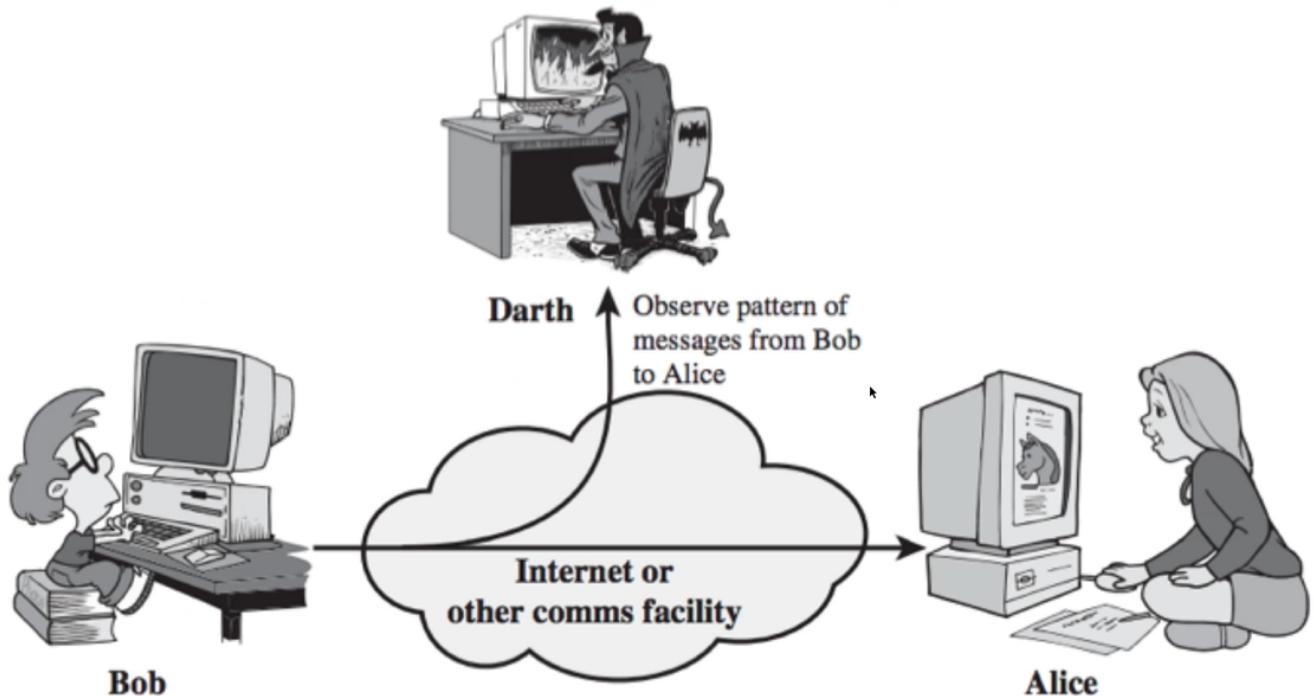
- There exist several kinds of confidentiality
 - Connection confidentiality
 - Connectionless confidentiality
 - Selective-field confidentiality
 - Traffic-Flow confidentiality
- Mechanisms able to guarantee confidentiality:
 - Criptography
- Related Attacks
 - Eavesdropping, sniffing

l'informazione è accessibile solo a chi è esattamente autorizzato ad accedere. La crittografia garantisce la confidenzialità.

Gli attacchi relativi sono attacchi passivi ovvero non implicano la generazione di pacchetti di traffico

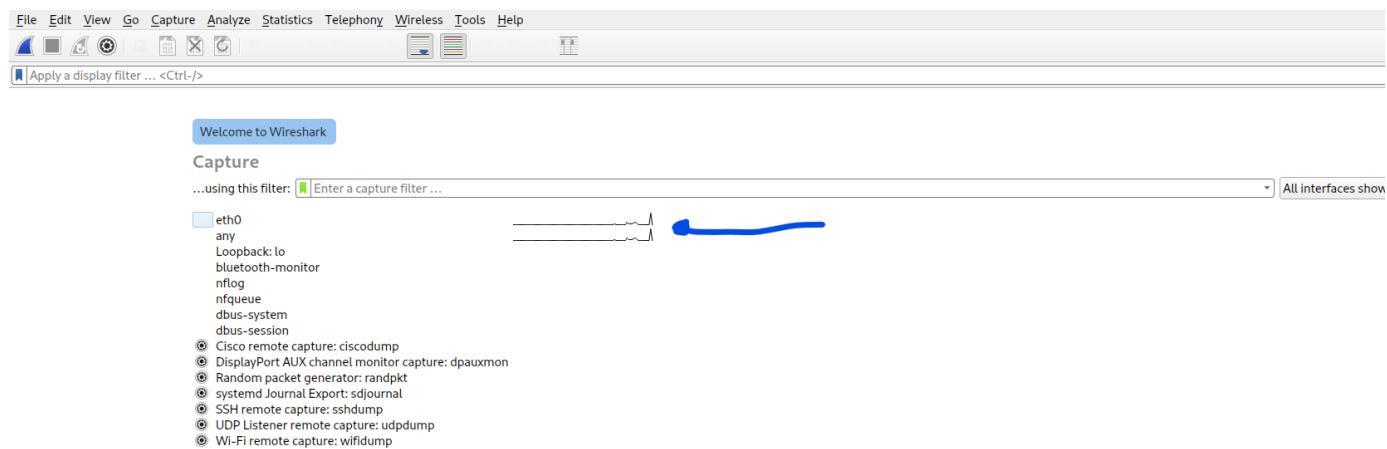


Passive Attacks: Eavesdropping



Supponendo di essere nel caso fortunato perché ci troviamo nel caso di riuscire ad analizzare il traffico, esistono dei tool come Wireshark per sniffare il traffico.

Selezioniamo l'interfaccia di rete che fa traffico:



Nella finestra di Wireshark ci sono diverse sezioni. Si tratta di un analizzatore di protocollo. Nella finestra ci sono diverse informazioni riassuntive:

No.	Time	Source	Destination	Protocol	Length	Info
15099	38.205958	192.168.77.176	52.112.166.1	UDP	1228	50052 → 3481 Len=1186
15100	38.205961	192.168.77.176	52.112.166.1	UDP	1228	50052 → 3481 Len=1186
15101	38.205965	192.168.77.176	52.112.166.1	UDP	1228	50052 → 3481 Len=1186
15102	38.205968	192.168.77.176	52.112.166.1	UDP	1228	50052 → 3481 Len=1186
15103	38.205971	192.168.77.176	52.112.166.1	UDP	1228	50052 → 3481 Len=1186
15104	38.206039	192.168.77.176	52.112.166.1	UDP	1228	50052 → 3481 Len=1186
15105	38.206041	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15106	38.206043	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15107	38.206044	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15108	38.206046	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15109	38.206048	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15110	38.206050	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15111	38.206069	192.168.77.176	52.112.166.1	UDP	176	50015 → 3479 Len=134
15112	38.216237	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15113	38.216241	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15114	38.216244	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15115	38.216248	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15116	38.216252	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15117	38.216255	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15118	38.216257	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15119	38.216261	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15120	38.216263	192.168.77.176	52.112.166.1	UDP	1230	50052 → 3481 Len=1188
15121	38.216266	192.168.77.176	52.112.166.1	UDP	1226	50052 → 3481 Len=1184
15122	38.216310	192.168.77.176	52.112.166.1	UDP	1226	50052 → 3481 Len=1184
15123	38.216311	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15124	38.216313	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15125	38.216315	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15126	38.216317	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15127	38.216319	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15128	38.226314	192.168.77.176	52.112.166.1	UDP	160	50015 → 3479 Len=118
15129	38.226538	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15130	38.226542	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15131	38.226544	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15132	38.226547	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197
15133	38.226552	192.168.77.176	52.112.166.1	UDP	1239	50052 → 3481 Len=1197

> Frame 15119: 1230 bytes on wire (9840 bits), 1230 bytes captured (9840 bits) on interface en9, id 0
 > Ethernet II, Src: ASIXElec_cc:e1:2b (f8:e4:3b:cc:e1:2b), Dst: TP-Link_26:f0:76 (10:27:f5:26:f0:76)
 > Internet Protocol Version 4, Src: 192.168.77.176, Dst: 52.112.166.1
 > User Datagram Protocol, Src Port: 50052, Dst Port: 3481
 > Data (1188 bytes)

I dettagli del pacchetto sono divisi per ciascun livello della pila TCP/IP.

l'header MAC

34 0.188554471	216.58.209.36	192.168.64.128	TCP	60 80 → 51882 [FIN, PSH, ACK] Seq=17964 Ack=80 Win=64239 Len=0
35 0.188570654	192.168.64.128	216.58.209.36	TCP	54 51882 → 80 [ACK] Seq=80 Ack=17965 Win=63540 Len=0

> Frame 35: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
 > Ethernet II, Src: VMware_7f:2c:4a (00:0c:29:7f:2c:4a), Dst: VMware_f2:7a:28 (00:50:56:f2:7a:28)
 > Destination: VMware_f2:7a:28 (00:50:56:f2:7a:28)
 > Source: VMware_7f:2c:4a (00:0c:29:7f:2c:4a)
 Type: IPv4 (0x0800)
 > Internet Protocol Version 4, Src: 192.168.64.128, Dst: 216.58.209.36
 > Transmission Control Protocol, Src Port: 51882, Dst Port: 80, Seq: 80, Ack: 17965, Len: 0

l'header IPv4

35 0.188570654	192.168.64.128	216.58.209.36	TCP	54 51882 → 80 [ACK] Seq=80 Ack=17965 Win=63540 Len=0
> Frame 35: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0 > Ethernet II, Src: VMware_7f:2c:4a (00:0c:29:7f:2c:4a), Dst: VMware_f2:7a:28 (00:50:56:f2:7a:28) > Internet Protocol Version 4, Src: 192.168.64.128, Dst: 216.58.209.36				

0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSGP: CS0, ECN: Not-ECT)
 Total Length: 40
 Identification: 0x0000 (0)
 > 010. = Flags: 0x2, Don't fragment
 ...0 0000 0000 0000 = Fragment Offset: 0
 Time to Live: 64
 Protocol: TCP (6)
 Header Checksum: 0x9048 [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.64.128
 Destination Address: 216.58.209.36
 > Transmission Control Protocol, Src Port: 51882, Dst Port: 80, Seq: 80, Ack: 17965, Len: 0

Authentication



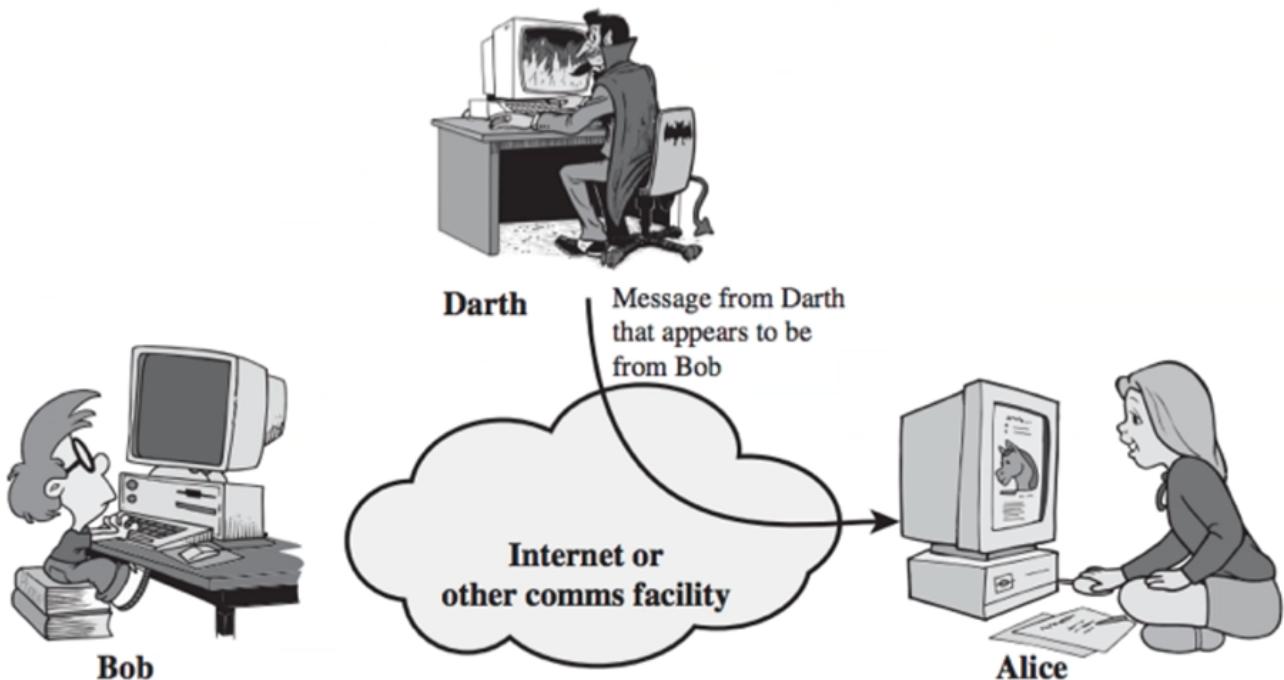
Security services: Authentication

Authentication is the act of establishing or confirming something (or someone) as authentic, that is, that claims made by or about the subject are true

- It can be required at
 - Data origin authentication
 - Peer entity authentication
- Mechanisms able to guarantee Information authenticity:
 - A difficult-to-reproduce physical artifact, such as a seal, signature, watermark, special stationery, or fingerprint
 - A shared secret, such as a passphrase, in the content of the message
 - An electronic signature; public key infrastructure is often used to cryptographically guarantee that a message has been signed by the holder of a particular private key
- Related Attacks
 - Masquerade, spoofing, traffic generation



Active Attacks: masquerade





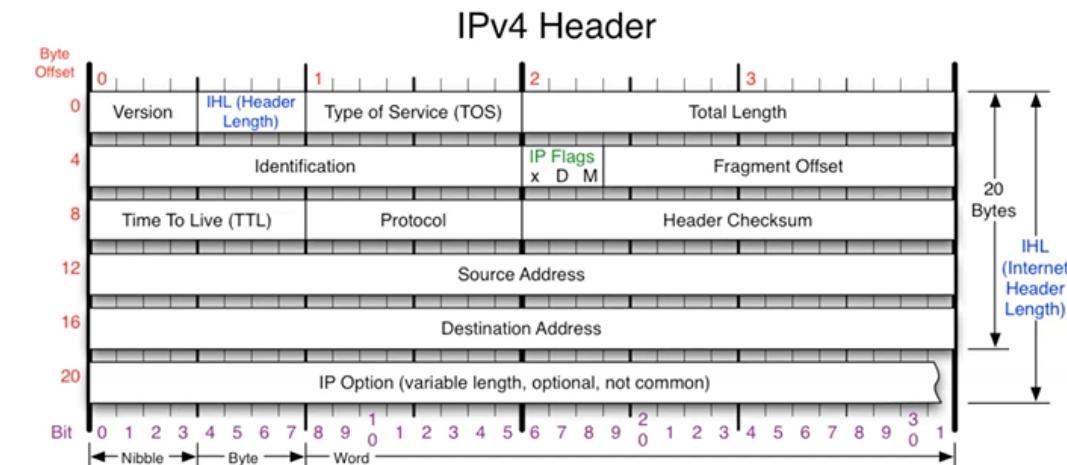
Security services: Integrity

Data integrity is data that has a complete or whole structure. All characteristics of the data including business rules, rules for how pieces of data relate, dates, definitions and lineage must be correct for data to be complete

- There exist several kinds of integrity
 - Connection integrity with/without recovery
 - Connectionless integrity
 - Selective fields integrity
- Mechanisms able to guarantee integrity:
 - Hash functions (trivial example: CRC of IP header)
 - Data authentication
 - Digital signature
- Related Attacks
 - Man-in-the-Middle (MitM)

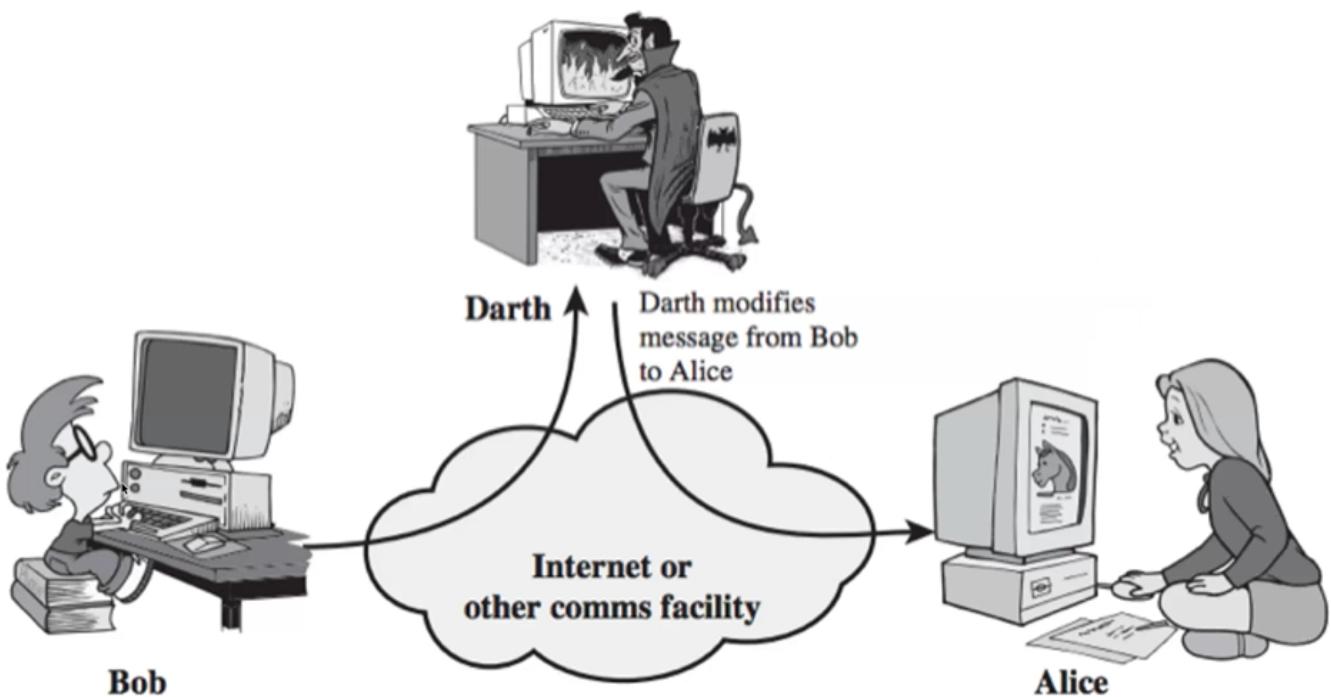


Integrity: a trivial example

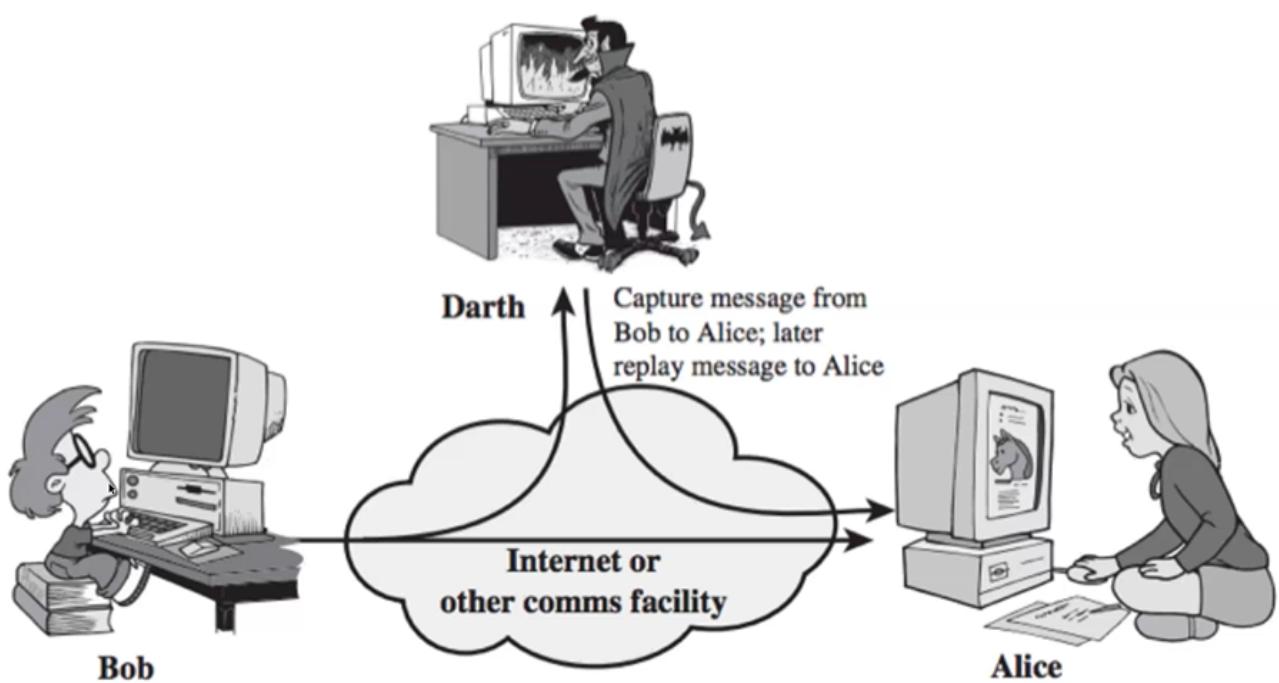


Version	Protocol	Fragment Offset	IP Flags
Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.	IP Protocol ID. Including (but not limited to): 1 ICMP 17 UDP 57 SKIP 2 IGMP 47 GRE 88 EIGRP 6 TCP 50 ESP 89 OSPF 9 IGRP 51 AH 115 L2TP	Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.	x 0x80 reserved (evil bit) D 0x40 Do Not Fragment M 0x20 More Fragments follow
Header Length	Total Length	Header Checksum	RFC 791
Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.	Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.	Checksum of entire IP header	Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Active Attacks: message modification



Active Attacks: replay





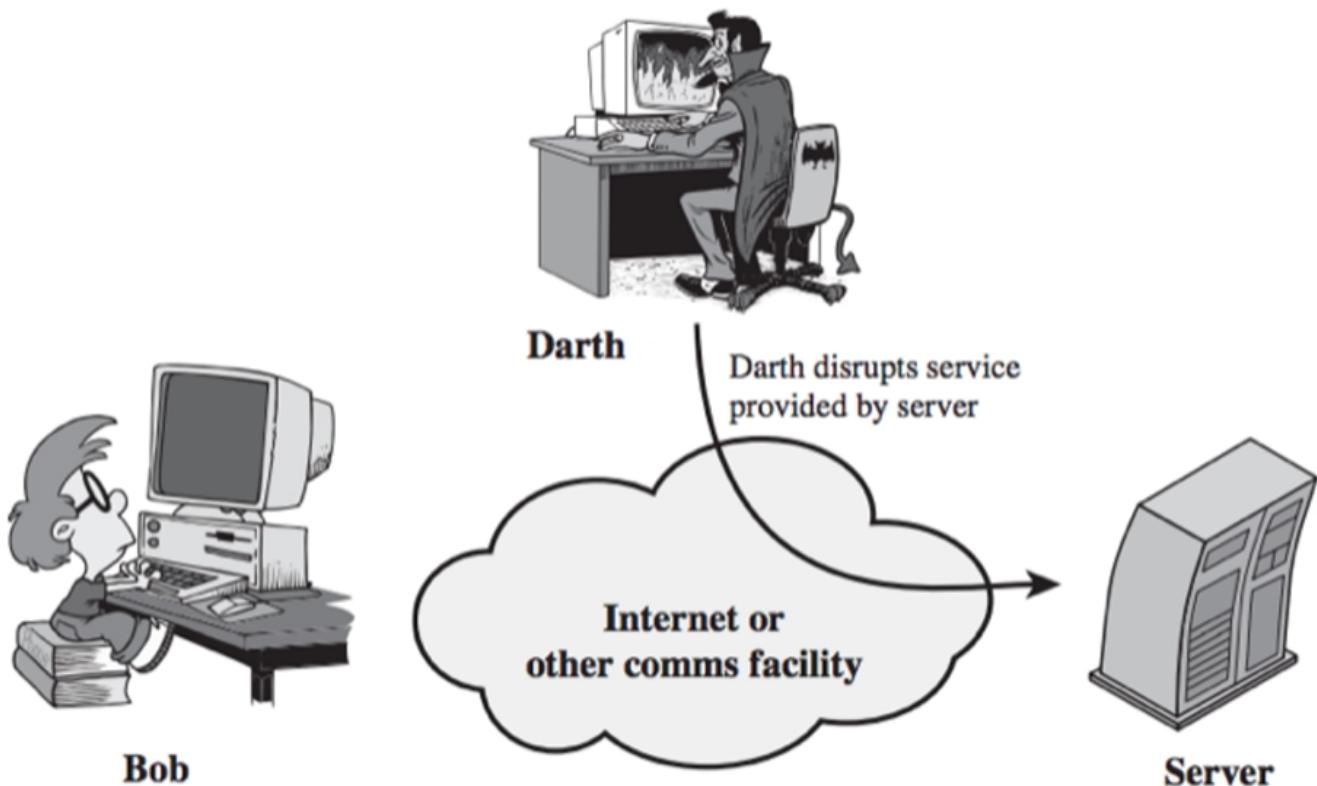
Security services: Availability

The degree to which a system, subsystem, or equipment is operable and in a committable state at the start of a mission, when the mission is called for at an unknown, i.e., a random, time. Simply put, availability is the proportion of time a system is in a functioning condition

- Mechanisms able to guarantee availability:
 - ???
- Related Attacks
 - Denial of Service (DoS)
 - Distributed Denial of Service (DDoS)



Active Attacks: denial of service



Non-repudiation



Security services: Non Repudiation

Non-repudiation is the concept of ensuring that a party in a dispute cannot repudiate, or refute the validity of a statement or contract

- There exist several kinds of non-repudiation
 - Origin non-repudiation
 - Destination non-repudiation
- Mechanisms able to guarantee non repudiation:
 - Digital Signature (only partly)
 - Trusted Third Party
- **Related attacks**

Access Control



Security services: Access Control

The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions can occur, and what those accessing the resource are allowed to).

- Mechanisms able to guarantee access control:
 - Access Control List
 - Login/Password
 - Firewall
- **Related Attacks**
 - Unauthorized access
 - Privilege escalation (User-to-Root - U2R)

Ciclo di vita di un attacco di rete



LifeCycle of a Network Attack

1. Information Gathering
2. Scanning
3. Gaining an access
4. Maintain the access
5. Clean evidences

It is important to avoid every and each distinct phase

- Security at all possible levels
 - Host-level security and network-level security
 - Network security and access security

Information Gathering



Information Gathering

- **Social engineering** is the act of manipulating people into performing actions or divulging confidential information, rather than by breaking in or using technical hacking techniques
 - **Passive:** involves acquiring information without directly interacting with the target (e.g., searching public records or news releases)
 - **Active:** involves interacting with the target directly by any means (e.g., telephone calls to the help desk or technical dept.)
- Several techniques:
 - Pretexting
 - Diversion theft
 - ^I Phishing
 - ...

Il comando `host` restituisce l'indirizzo IP associato al server web target:

```
[root@kali]# host www.unipi.it
www.unipi.it is an alias for wwwnew2.unipi.it.
wwwnew2.unipi.it has address 131.114.21.42

[root@kali]#
```

Lanciando il comando whois sull'indirizzo IP ottenuto:

```
[root@kali]# whois 131.114.21.42
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2023, American Registry for Internet Numbers, Ltd.
# Home

NetRange:      131.114.0.0 - 131.117.255.255
CIDR:         131.114.0.0/15, 131.116.0.0/15
NetName:       RIPE-ERX-131-114-0-0
NetHandle:     NET-131-114-0-0-1
Parent:        NET131 (NET-131-0-0-0-0)
NetType:       Early Registrations, Transferred to RIPE NCC
OriginAS:
Organization: RIPE Network Coordination Centre (RIPE)
RegDate:      2004-02-04
Updated:      2011-09-26
Comment:      These addresses have been further assigned to users in the RIPE NCC region. Contact information can be found in the RIPE database at http://www.ripe.net/
whois:
Ref:          https://rdap.arin.net/registry/ip/131.114.0.0
ResourceLink: https://apps.db.ripe.net/search/query.html
ResourceLink: whois.ripe.net

OrgName:      RIPE Network Coordination Centre
OrgId:        RIPE
Address:      P.O. Box 10096
City:         Amsterdam
StateProv:
PostalCode:   1001EB
Country:      NL
RegDate:
Updated:      2013-07-29
Ref:          https://rdap.arin.net/registry/entity/RIPE

ReferralServer: whois://whois.ripe.net
ResourceLink: https://apps.db.ripe.net/search/query.html

OrgAbuseHandle: ABUSE3850-ARIN
OrgAbuseName:  Abuse Contact
```

Questo IP appartiene al CIDR indicato in blu che è di appartenenza del RIPE:

```
NetRange:      131.114.0.0 - 131.117.255.255
CIDR:         131.114.0.0/15, 131.116.0.0/15
NetName:       RIPE-ERX-131-114-0-0
NetHandle:     NET-131-114-0-0-1
Parent:        NET131 (NET-131-0-0-0-0)
NetType:       Early Registrations, Transferred to RIPE NCC
OriginAS:
Organization: RIPE Network Coordination Centre (RIPE)
RegDate:      2004-02-04
Updated:      2011-09-26
Comment:      These addresses have been further assigned to users in the RIPE NCC region. Contact information can be found in the RIPE database at http://www.ripe.net/
whois:
Ref:          https://rdap.arin.net/registry/ip/131.114.0.0
ResourceLink: https://apps.db.ripe.net/search/query.html
ResourceLink: whois.ripe.net

OrgName:      RIPE Network Coordination Centre
OrgId:        RIPE
Address:      P.O. Box 10096
City:         Amsterdam
StateProv:
PostalCode:   1001EB
```

La rete appartiene a una rete detta PISA-NET che ha 2^{16} indirizzi:

```
inetnum: stem      131.114.0.0 - 131.114.255.255
netname:        PISA-NET
descr:          UNI-Pisa
country:        IT
org:            ORG-UDSD34-RIPE
sponsoring-org: ORG-GIRa1-RIPE
admin-c:        SS103-RIPE
tech-c:         MD26948-RIPE
tech-c:         SB30015-RIPE
status:         LEGACY
remarks:        This prefix is statically assigned
remarks:        To notify abuse mailto: cert@garr.it
remarks:        GARR - Italian academic and research network
mnt-by:         RIPE-NCC-LEGACY-MNT
mnt-by:         GARR-LIR
mnt-irt:        IRT-GARR-CERT
created:        2002-04-11T16:41:50Z
last-modified:  2022-08-26T11:33:48Z
source:         RIPE

organisation:   ORG-UDSD34-RIPE
org-name:       Universita' di Pisa
country:        IT
org-type:       OTHER
address:        Lungarno Pacinotti, 43/44
address:        56126
address:        Pisa
address:        I
abuse-c:        AG16225-RIPE
mnt-ref:        GARR-LIR
mnt-by:         GARR-LIR
created:        2014-04-09T12:30:58Z
last-modified:  2022-12-01T17:31:47Z
source:         Burp    RIPE # Filtered

person:         Maurizio Davini
address:        UNI-Pisa
address:        Via Livornese 1289/1291
address:        I-56122-56122
address:        Italy
phone:          +39 050 2210960
nic-hdl:        MD26948-RIPE
mnt-by:         GARR-LIR
```

questo è importante perchè io potrei voler attaccare un IP all'interno di quella rete e non necessariamente l'IP originario.

Ottengo inoltre informazioni interessanti per tecniche di social engineering:

```
person: Maurizio Davini
address: UNI-Pisa
address: Via Livornese 1289/1291
address: I-56122-56122
address: Italy
phone: +39 050 2210960
nic-hdl: MD26948-RIPE
mnt-by: GARR-LIR
created: 2021-03-08T06:58:19Z
last-modified: 2022-01-28T13:25:00Z
source: RIPE # Filtered

person: Simone Badalassi
address: UNI-Pisa
address: Via Livornese 1289/1291
address: I-56122-56122
address: Italy
phone: +39 050 2210953
nic-hdl: SB30015-RIPE
mnt-by: GARR-LIR
created: 2022-08-26T11:33:29Z
last-modified: 2022-08-26T11:33:29Z
source: RIPE # Filtered

person: Stefano Suin
address: UNI-Pisa
address: Via Livornese 1289/1291
address: I-56122-Pisa
address: Italy
phone: +39 050 2210962
fax-no: +39 050 2212560
nic-hdl: SS103-RIPE
mnt-by: GARR-LIR
created: 1970-01-01T00:00:00Z
last-modified: 2022-01-28T13:26:02Z
source: RIPE # Filtered
```

I server di mail per chi ha server onpremise di posta, possono essere un punto di ingresso vulnerabile.

Usando il comando `nslookup` con `set type=MX`:

```
[root@kali)-[/home/kali]
# nslookup
> set type=MX
> www.unipi.it
;; communications error to 192.168.64.2#53: timed out
Server:      192.168.64.2
Address:     192.168.64.2#53

Non-authoritative answer:
www.unipi.it    canonical name = wwwnew2.unipi.it.

Authoritative answers can be found from:
unipi.it
    origin = ns1.unipi.it
    mail addr = postmaster.unipi.it
    serial = 2023052501
    refresh = 3600
    retry = 300
    expire = 1209600
    minimum = 3600
> set type=A
> ns1.unipi.it
;; communications error to 192.168.64.2#53: timed out
Server:      192.168.64.2
Address:     192.168.64.2#53

Non-authoritative answer:
Name:  ns1.unipi.it
Address: 131.114.21.10
> ns2.unipi.it
;; communications error to 192.168.64.2#53: timed out
Server:      192.168.64.2
Address:     192.168.64.2#53

Non-authoritative answer:
Name:  ns2.unipi.it
Address: 131.114.21.5
>
```

da cui ricaviamo anche l'esistenza di alcuni indirizzi di DNS server di Unipi come l'ns1 e l'ns2.

09/06/2023

Per automatizzare la ricerca dei vari server web contenuti nel dominio unipi.it.

Proviamo a scaricare la pagina web di www.unipi.it:

```

└─(kali㉿kali)-[~]
$ wget www.unipi.it
--2023-06-09 12:20:04-- http://www.unipi.it/
Resolving www.unipi.it (www.unipi.it)... 131.114.21.42
Connecting to www.unipi.it (www.unipi.it)|131.114.21.42|:80... connected.
HTTP request sent, awaiting response ... 301 Moved Permanently
Location: https://www.unipi.it/ [following]
--2023-06-09 12:20:04-- https://www.unipi.it/
Connecting to www.unipi.it (www.unipi.it)|131.114.21.42|:443... connected.
HTTP request sent, awaiting response ... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html
[ ⇌
2023-06-09 12:20:05 (948 KB/s) - 'index.html' saved [55775]

└─(kali㉿kali)-[~]
$ ls
Desktop Documents Downloads index.html Music Pictures Public Templates Videos
eulibllovvv
└─(kali㉿kali)-[~]
$ 

```

Tutti i link all'interno della pagina iniziano con un tag href quindi posso filtrarli isolando le righe contenenti il tag HTML href:

```

└─(kali㉿kali)-[~]
$ cat index.html | grep href
<base href="https://www.unipi.it/" />
<link href="/templates/homepagetemplate/favicon.ico" rel="shortcut icon" type="image/vnd.microsoft.icon" />
<link href="/index.php/component/fpss/module/949?format=feed&type=rss" rel="alternate" type="application/rss+xml" title="Frontpage Slideshow (by JoomlaWorks) RSS Feed" />
<link href="/index.php/component/fpss/module/949?format=feed&type=atom" rel="alternate" type="application/atom+xml" title="Frontpage Slideshow (by JoomlaWorks) Atom Feed" />
<link rel="stylesheet" href="/components/com_k2/css/k2.css" type="text/css" />
<link rel="stylesheet" href="/templates/homepagetemplate/wright/bootstrap/css/bootstrap.min.css" type="text/css" />
<link rel="stylesheet" href="/templates/homepagetemplate/wright/bootstrap/css/bootstrap-responsive.min.css" type="text/css" />
<link rel="stylesheet" href="/templates/homepagetemplate/wright/fontawesome/css/font-awesome.min.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/templates/homepagetemplate/wright/css/typography.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/templates/homepagetemplate/wright/css/joomla25.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/templates/homepagetemplate/wright/css/joomla25.responsive.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/templates/homepagetemplate/css/1_template.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/templates/homepagetemplate/css/2_template.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/templates/homepagetemplate/css/style-Homepage.css" type="text/css" />
<link rel="stylesheet" href="/templates/homepagetemplate/css/ricerca.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/modules/mod_unipisocial/assets/css/social.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/modules/mod_box_carousel/assets/css/boxini.css" type="text/css" />
<link rel="stylesheet" href="/plugins/content/phocadownload/assets/css/phocadownload.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/modules/mod_rseventspro_upcoming/style.css" type="text/css" />
<link rel="stylesheet" href="/templates/homepagetemplate/html/mod_fpss/JJ-Rasper/css/template.css.php?width=900&mp;height=300&sidebarWidth=240&timer=0&thumbnailViewportWidth=65&thumbnailViewportHeight=40&mid=949" type="text/css" />
<link rel="stylesheet" href="/modules/mod_maximenuck/themes/css3megamenu/css/maximenuck.php?monid=maximenuck" type="text/css" />
<link rel="stylesheet" href="/modules/mod_maximenuck/templatelayers/beez_20-position1.css" type="text/css" />
<link rel="stylesheet" href="/modules/mod_maximenuck/assets/maximenuresponsiveck.css" type="text/css" />
<link rel="stylesheet" href="/templates/homepagetemplate/html/mod_maximenuck/overridemenuck.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/plugins/system/fmalertcookies/assets/css/bootstrap.min.css" type="text/css" />
<link rel="stylesheet" href="https://www.unipi.it/plugins/system/fmalertcookies/assets/css/custom.css" type="text/css" />
<link rel="stylesheet" href="/templates/js_wright/css/explorer.css" type="text/css" />

```

A noi servono tutti gli indirizzi che sono del dominio di secondo livello unipi.it, dunque il dominio di secondo livello deve rimanere unipi.it:

```
File Actions Edit View Help
news
news Trash
news
news
tutte-le-news
www.unipi.it
studenti">STUDENTI<
matricolandomosi.unipi.it
www.unipi.it
ricerca">Ricerca<
bandi-ricerca" >Bandi ricerca<
personale">Personale<
docenti2" >Docenti<
strutture">Strutture<
dipartimenti" >Dipartimenti<
concorsi-gare-e-bandi">Bandi e Concorsi<
venditeimmobiliari.unipi.it
amministrazione
www.unipi.it
www.unipi.it
www.unipi.it
www.unipi.it
www.unipi.it
sostenibile.unipi.it">
sostenibile.unipi.it">
www.unipi.it
www.unipi.it
www.facebook.com
twitter.com
www.youtube.com
instagram.com
<a href="#" target="\_blank">
www.linkedin.com
www.indicepa.it
documenti-ateneo
unimap.unipi.it
documenti-ateneo
documenti-ateneo
note-legali">Note legali<
documenti-ateneo
urp">Urps<
documenti-ateneo
amministrazione
atti-di-notifica">Atti di notifica<
alboufficiale.unipi.it
statuto-regolamenti
documenti-ateneo

[(kali㉿kali)-[~]
$ cat index.html | grep href= | cut -d "/" -f3
```



con il seguente comando prendiamo il terzo campo.

Adesso possiamo filtrare per il solo dominio unipi.it eliminando i duplicati:

```
[(kali㉿kali)-[~]
$ cat index.html | grep href= | cut -d "/" -f3 | grep unipi.it | sort -u
alboufficiale.unipi.it
matricolandomosi.unipi.it
sostenibile.unipi.it">
unimap.unipi.it
venditeimmobiliari.unipi.it
www.unipi.it
```

Creiamo un file bash per estrarre gli IP dei nomi estratti:

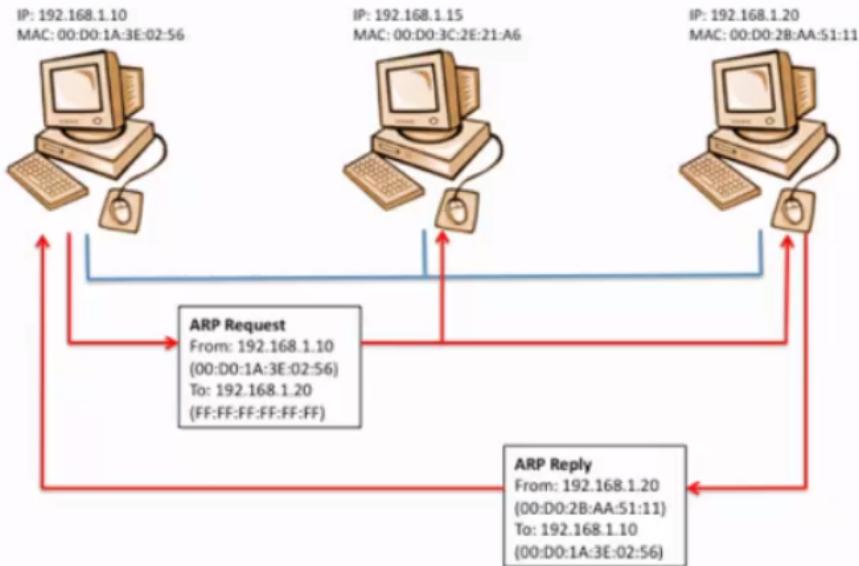
```
for hostname in $(cat server_web.txt); do
    host $hostname | grep "has address" | cut -d " " -f4
done
```

```
└─(kali㉿kali)-[~]
$ ./find_ip.sh
131.114.72.200
131.114.142.31
131.114.142.31
131.114.142.38
131.114.142.31
131.114.21.42
```

Per fare una ricerca esaustiva andare quanto meno a riapplicare tutto questo alle singole pagine.

Attacchi di livello 2 TCP/IP

Recall: ARP protocol



ARP attacks

- **ARP poisoning**

- An attacker C can send a spoofed ARP reply (gratuitous ARP) message to a given host A, “saying” to be host B

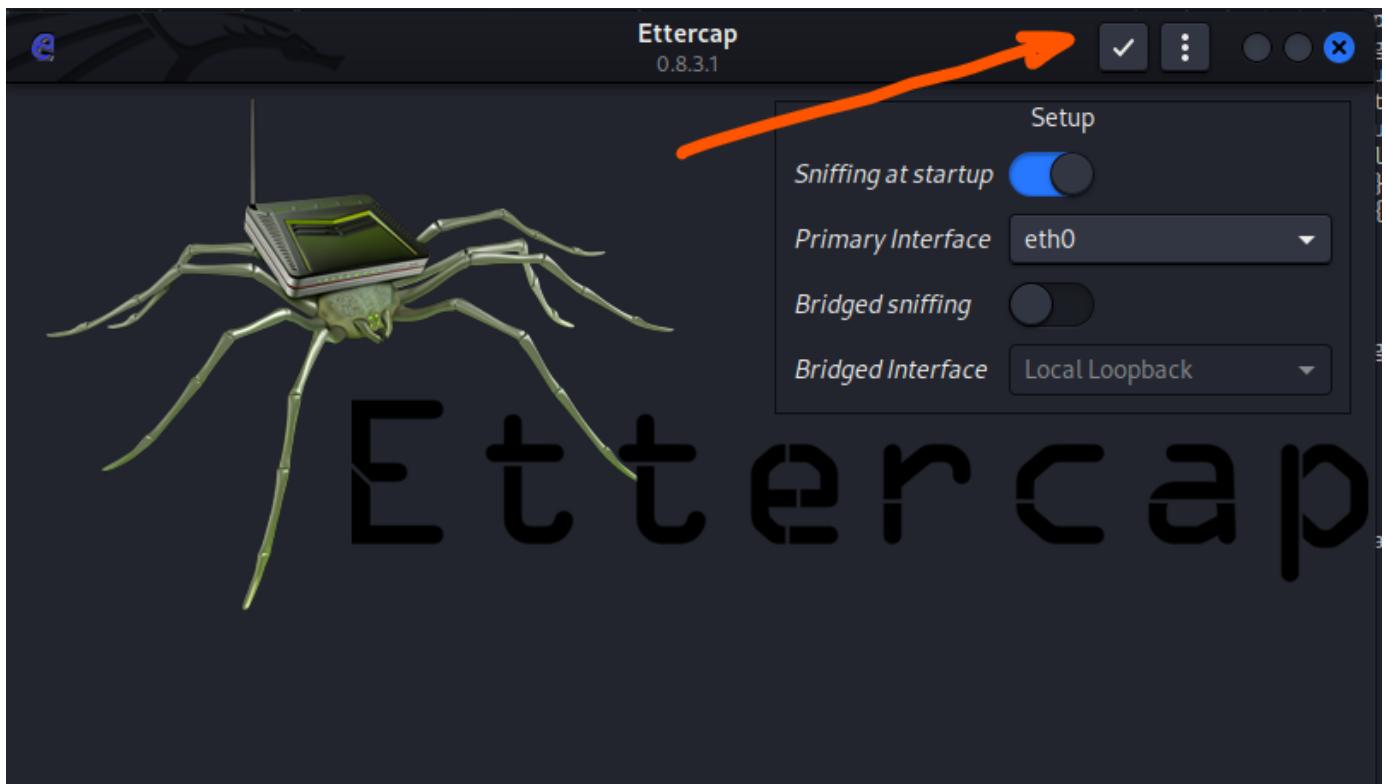
- **MAC flooding**

- By MAC flooding a switch’s ARP table with spoofed ARP replies, the attacker can overload switches, making them enter in “forwarding mode”

ettercap

permette di effettuare un attacco di tipo MITM permettendo di fare ARP poisoning.

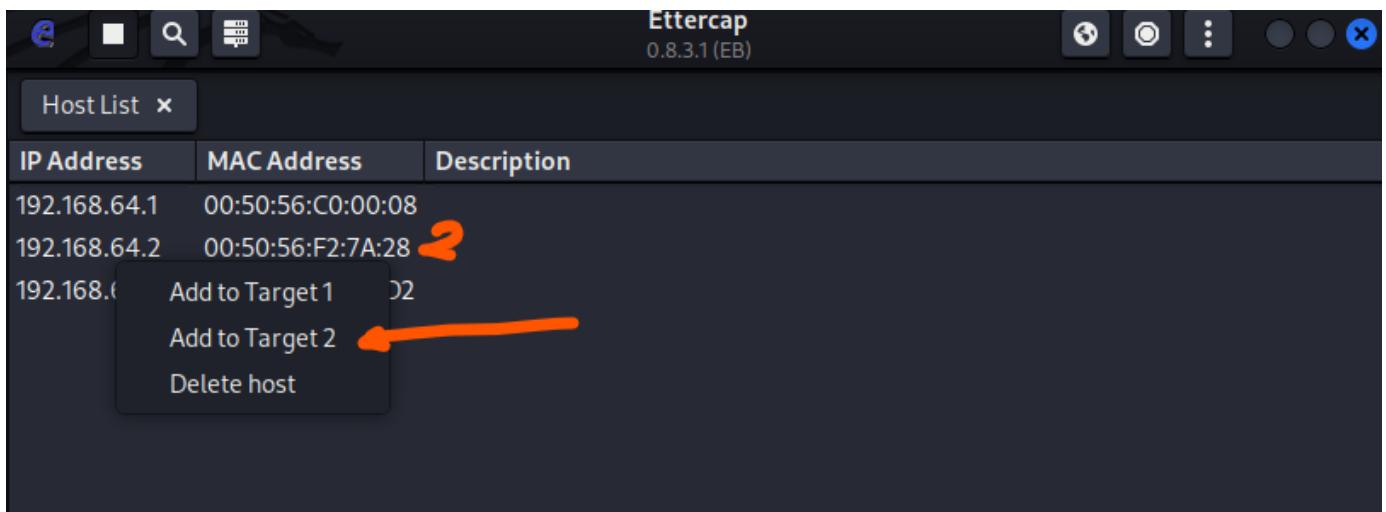
```
sudo ettercap -G
```



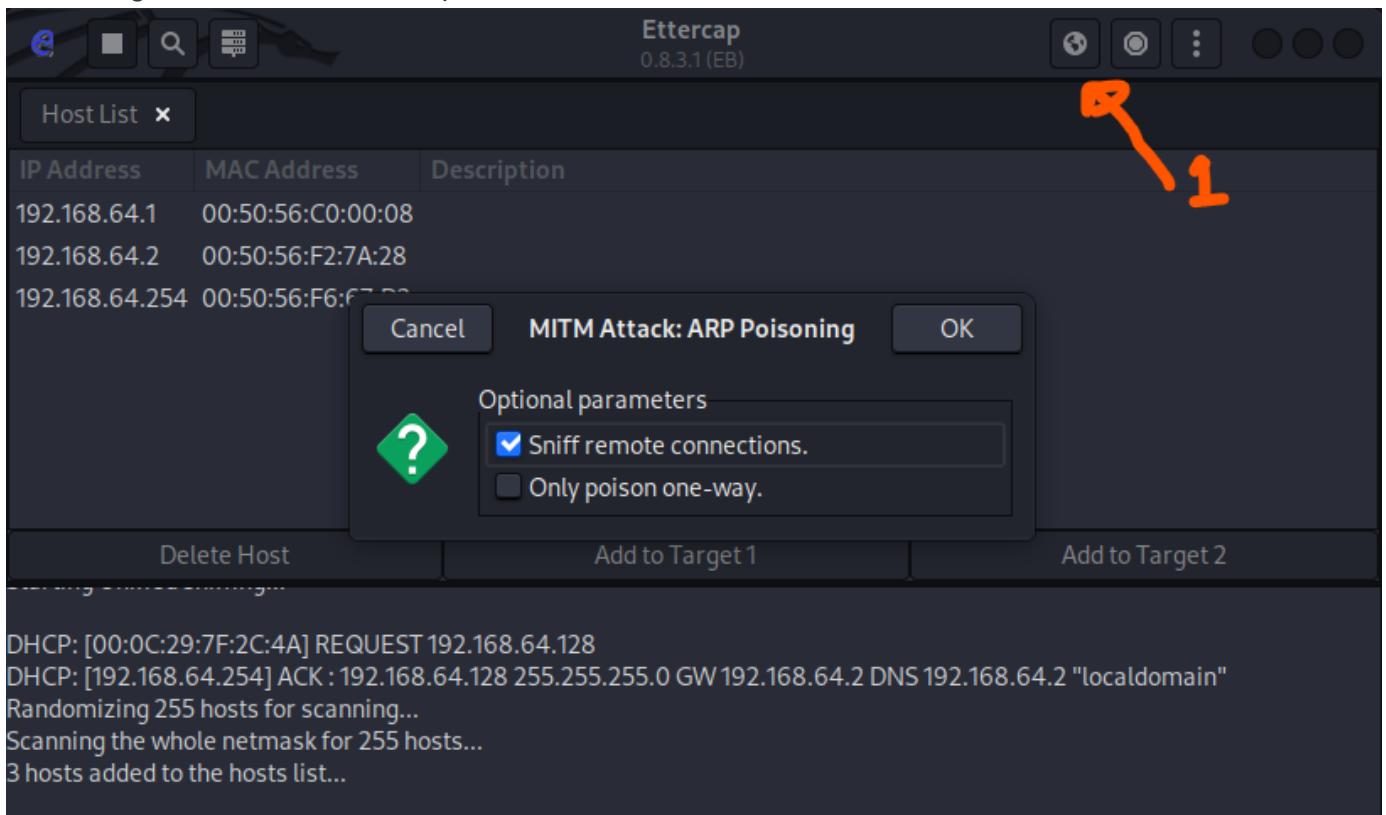
The screenshot shows the Ettercap 0.8.3.1 application window with the title 'Ettercap 0.8.3.1 (EB)' at the top. On the left, there is a 'Hosts' menu with options: 'Hosts list', 'Enable IPv6 Scan' (unchecked), 'Scan for hosts', 'Load hosts from file ...', and 'Save hosts to file ...'. A red arrow points to the 'Scan for hosts' option. In the center, there is a terminal-like window displaying network traffic and configuration messages. At the bottom, there is a 'Host List' table with three rows:

IP Address	MAC Address	Description
192.168.64.1	00:50:56:00:00:08	Add to Target 1
192.168.64.254	00:50:56:00:00:08	GW

A red arrow points to the 'Add to Target 1' entry in the host list table. The table also includes a 'Delete host' option at the bottom.



Per rimanere trasparenti evitando di rimuovere pacchetti dalla rete, è possibile abilitare il packet forwarding sulla macchina sulla quale si sta facendo il MITM:



```
vim /etc/sysctl.conf
sysctl -p
cat /proc/sys/net/ipv4/ip_forward
```

Una volta avviato l'attacco, tramite un tool di network sniffing come Wireshark il traffico ICMP (è stato usato come esempio un ping tra il Windows 10 e il Windows Server) è possibile vederlo a differenza di prima che non era possibile, questo grazie ad ettercap che attua il MITM.

17/06/2023

L'attacco di tipo DOS o DDOS è un attacco contro il servizio di disponibilità (availability) mirato a rendere non accessibile un servizio agli utenti legittimi, come ad esempio un attacco ad un web server per impedire che altri utenti possano connettersi al web server.

Il primo attacco di quasi DOS della storia avvenne nel 1903 quando Marconi a Londra doveva fare la dimostrazione del telegrafo su lunga distanza e prima che iniziasse la trasmissione vera e propria, un sedicente mago iniziò a mandare un segnale (la parola RATTI) che infastidiva la comunicazione.



Denial of Service

- A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users.
- The five basic types of attack are:
 - Consumption of computational resources, such as bandwidth, disk space, or processor time
 - Disruption of configuration information, such as routing information
 - Disruption of state information, such as unsolicited resetting of TCP sessions
 - Disruption of physical network components
 - Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately

Hanno un problema in quanto sono difficili da evitare in quanto se l'attacco DOS inonda di richieste lecite bisogna iniziare a fare analisi dei pacchetti, ma cambiando l'IP sorgente si può bypassare questo problema. Gli attacchi DOS ancora oggi sono abbastanza critici su internet.

Ci sono 5 tipologie di attacchi DOS:

- consumo della banda oppure la potenza di calcolo o lo spazio sul disco della macchina attaccata
- poisoning delle informazioni di routing ovvero iniettare in rete dei messaggi fasulli che creano un loop topologico o annullano le rotte funzionanti esistenti all'interno di una rete
- attacchi che si basano sul cercare di far cambiare le informazioni di stato di una macchina. Ad esempio TCP è un protocollo di trasporto che prevede l'instaurazione di una connessione, dunque se riesco a convincere uno dei due estremi della connessione che la connessione in realtà è terminata posso indurre il reset della connessione TCP
- attacco a livello fisico come per esempio fare jamming su una connessione wireless
- disturbo della comunicazione con segnali di rumore

Possiamo anche inviare del traffico UDP mandandone in quantità tale da saturare le risorse del destinatario. Se non c'è nessun dispositivo di sicurezza un attacco semplice come questo può essere molto letale:



UDP flooding

- UDP flood attack can be initiated by sending a large number of UDP packets to random ports on a remote host.
As a result the host will:
 - Check for the application listening at that port;
 - See that no application listens at that port;
 - Reply with an ICMP Destination Unreachable packet.
- For a large number of UDP packets, the victimized system will be forced into sending many ICMP packets, eventually leading it to be unreachable by other clients

L'ICMP è un protocollo usato per controllare lo stato dei devices sulla rete e può essere sfruttato per alcuni attacchi come:

- **Smurf:** il ping si basa sullo scambio di 2 pacchetti Echo request/reply, l'idea è quello di far ricevere dalla vittima un numero enorme di Echo Request magari mettendo come IP sorgete degli IP random in modo da non ottenere la risposta e rischiare di saturare la propria banda durante l'attacco oppure più intelligentemente inviare un pacchetto Echo Request con IP sorgente l'IP della vittima e IP destinatario l'IP di broadcast. Potenzialmente questo pacchetto può raggiungere tutte le macchine del mondo. Idealmente tutti i devices connessi in rete rispondono e inviano un Echo Reply alla vittima che si trova a ricevere moltissimi pacchetti. Questo tipo di attacco non è più facile da attuare in quanto la maggior parte dei pacchetti broadcast vengono scartati dai router e inoltre le macchine possono essere configurate per non rispondere ai pacchetti ICMP broadcast.
- **Ping of Death:** è un attacco diverso concettualmente, si basa su un singolo pacchetto lecito ma più lungo del normale, 65536 byte che vengono frammentati ovviamente e una volta deframmentati sulla macchina vittima il SO crashava alla ricezione del pacchetto. I sistemi operativi odierni non



ICMP flooding

- A ping flood is a simple denial-of-service attack where the attacker overwhelms the victim with ICMP Echo Request (ping) packets
- **Smurf attack**
 - A Smurf attack consists of sending a large amount of ICMP echo request traffic to IP broadcast addresses, all of which have a spoofed source IP address of the intended victim
 - Most OSs can be configured not to answer to ICMP packets sent to broadcast IP address, but this doesn't prevent the victim to be attacked
- **Ping of Death**
 - A ping of death (abbreviated "POD") is a type of attack on a computer that involves sending a malformed or otherwise malicious ping to a computer.
 - Several OSs crash, when receiving an ICMP Echo request 65536 bytes long (fragmented, since it is longer than the maximum allowed packet length)



SYN flooding

- The SYN flood is a well known type of attack and is generally not effective against modern networks.
- It works if a server allocates resources after receiving a SYN, but before it has received the ACK.
- There are two methods, but both involve the server not receiving the ACK:
 - A malicious client can skip sending this last ACK message
 - Or by spoofing the source IP address in the SYN

The technology often used in 1996 for allocating resources for half open TCP connections involved a queue which was often very short (e.g., 8 entries long) with each entry of the queue being removed upon a completed connection, or upon expiry (e.g., after 3 minutes).

endpoint si scambiano dei pacchetti con cui sincronizzarsi sui parametri della connessione. Questa fase di instaurazione della connessione si basa sullo scambio di 3 pacchetti: SYN con alcuni parametri per gestire la connessione, SYN+ACK e infine il client manderà un pacchetto di ACK. Questo scambio di messaggi comporta che il server alla ricezione del messaggio di SYN si memorizzerà alcune informazioni di stato che serviranno per capire come stabilire le connessioni con i client. Chiaramente per realizzare ciò l'attaccante doveva non inviare mai il pacchetto ACK altrimenti la sua connessione sarebbe passata in un buffer per le connessioni aperte solitamente più grande.



SYN cookies

SYN cookies provide protection against the SYN flood by eliminating the resources allocated on the target host.

Daniel J. Bernstein, the technique's primary inventor, defines SYN Cookies as "particular choices of initial TCP sequence numbers by TCP servers."

- SYN Cookies allows a server to avoid dropping connections when the SYN queue fills up
- The server sends back the appropriate SYN+ACK response to the client but discards the SYN queue entry
- If the server then receives a subsequent ACK response from the client, the server is able to reconstruct the SYN queue entry using information encoded in the TCP sequence number

Un modo per evitare il SYN flooding è l'uso dei SYN cookies. All'interno dell'header TCP c'è un campo che si chiama sequence number che serve per capire quale è la posizione del pacchetto nel flusso e che quando mando l'ACK posso dire quale pacchetto ho ricevuto e quale invece no. Il sequence number è gestito in modo tale per cui è un numero random in quanto alcuni attacchi si basano sull'indovinare il sequence number della vittima. L'idea proposta dal Bernstein fu quella di sfruttare il fatto di poter scegliere il sequence number iniziale randomico in modo da non dover memorizzare alcuna informazione lato server in quanto tutte le informazioni sono all'interno del sequence number. Il pacchetto SYN inviato dall'attaccante avrà un sequence number, il pacchetto della vittima SYN+ACK avrà un sequence number del tutto random in quanto non implementa questa cosa ma è random ma all'interno del campo ACK ci sarà il valore dell'ACK per cui mi sta dicendo che ha ricevuto il pacchetto.



SYN cookies

- SYN cookie calculation: $n = t(5\text{bits}) | m(3\text{bits}) | s(24\text{bits})$, where
 - t = A slowly-incrementing timestamp (typically `time()` logically right-shifted 6 positions, which gives a 64 second resolution)
 - m = The maximum segment size (MSS) value that the server would have stored in the SYN queue entry
 - s = The result of a cryptographic secret function computed over the server IP address and port number, the client IP address and port number, and the value t . The returned value "s" must be a 24-bit value.
- When the server receives back an ACK (the seq number will be $n + 1$), it performs the following operations:
 - Checks the value t against the current time to see if the connection is expired.
 - Recomputes s to determine whether this is, indeed, a valid SYN Cookie.
 - Decodes the value m from the 3-bit encoding in the SYN Cookie, which it then can use to reconstruct the SYN queue entry
- Drawbacks:
 - the server is limited to only 8 unique MSS values
 - the server must reject all TCP options
 - a connection may freeze when the final ACK of the three-way handshake is lost and the client first awaits data from the server
- The newer TCP Cookie Transactions (TCPCT) standard is designed to overcome these shortcomings of SYN cookies

Christian Callebari - Ethical Hacking

Slide 59

Quando il server target riceve il pacchetto, il sequence number che manda all'interno del campo acknowledge sarà $n+1$ dunque quando poi riceve il terzo pacchetto il server deve effettuare alcuni controlli ovvero che il valore del timestamp sia sempre lo stesso, un tempo molto lento di 64 secondi che si stima non cambi di molto tra un pacchetto e il successivo e poi il valida l'hash.

Il problema è che il campo `s` del SYN cookie tiene conto solo di indirizzo IP e porte mentre all'interno del pacchetto di SYN e delle informazioni di stato che il server dovrebbe memorizzare ci sono anche le opzioni del TCP. Il campo opzione per definizione è un campo variabile.

Host scan



Host scan

- The result of a scan on a port is usually generalized into one of three categories:
 - **Open or Accepted:** The host sent a reply indicating that a service is listening on the port
 - **Closed or Denied or Not Listening:** The host sent a reply indicating that connections will be denied to the port (e.g., ICMP port unreachable message)
 - **Filtered, Dropped or Blocked:** There was no reply from the host

La finalità è in prima battuta quello di effettuare un analisi delle porte e dei servizi aperti sulla macchina target.

Lo stato delle porte può essere:

- aperto: la macchina ha un servizio che è in ascolto su quella porta
- chiuso: la macchina risponde con un messaggio ICMP destination port unreachable se la porta è chiusa
- filtrata: la macchina non mi risponde con nulla in quanto droppa il pacchetto ricevuto.



Host scan

- A host scan can be performed in several ways:
 - **FIN scanning:** useful if the SYN packets are blocked by the firewall. Closed ports reply to a FIN packet with the appropriate RST packet, whereas open ports ignore the packet on hand
 - nmap -sF 192.168.1.10
 - **Xmas scanning:** TCP packets with all the flags set
 - nmap -sX 192.168.1.10



Host scan

- A host scan can be performed in several ways:
 - **SYN scanning** also known as half-open scanning
 - nmap -sS 192.168.1.10
 - **UDP scanning**
 - nmap -sU 192.168.1.10
 - **ACK scanning:** can be useful in the case packet filtering blocks packets without the ACK flag set. It does not exactly determine whether the port is open or closed, but whether the port is filtered or unfiltered (useful to detect the presence of a firewall)
 - nmap -sA 192.168.1.10

Le scansioni possono essere fatte usando un:

- SYN scanning: mando un TCP SYN, il destinatario mi risponde con un pacchetto SYN+ACK, capisco che la porta è aperto e poi mando un pacchetto di RST
- UDP scanning: è incerta come scansione e alle volte anche poco precisa in quanto potrei non ricevere una risposta oppure perchè a livello di servizi molto spesso la prima connessione è in TCP e poi parlano in UDP se necessario
- ACK scanning: posso anche mandare un pacchetto di ACK
- FIN scanning: utile se i pacchetti SYN sono bloccati dal firewall e mi permette di capire quali porte sono chiuse in quanto ad un pacchetto di FIN in genere il SO su una porta chiusa risponde con un pacchetto di RST
- XMAS scanning: un pacchetto TCP con tutti i flag settati

Il comando da lanciare per effettuare lo scan degli host presenti nella rete:

→ **Public** nmap -sn -sL 192.168.77.0/24

In Wireshark applicare il seguente filtro Berkley per filtrare il traffico proveniente e diretto alla sola macchina attaccante verso la rete o l'IP target:

```
(ip.src == 192.168.1.105 || ip.dst == 192.168.1.105) && (ip.src == 192.168.1.0/24 || ip.dst == 192.168.1.0/24)
```

Il comando di nmap che fa il ping scan:

```
nmap -sn 192.168.1.0/24
```

analizzandolo in Wireshark usando un filtro di tipo ARP vedremo che ha scambiato traffico ARP:

No.	Time	Source	Destination	Protocol	Length Info
22	0:00:00.000000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
23	0:00:00.055000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
24	0:00:00.100000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
25	0:00:00.145000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
26	0:00:00.190000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
27	0:00:00.235000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
28	0:00:00.280000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
29	0:00:00.325000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
30	0:00:00.370000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
43	0:00:00.415000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
44	0:00:00.460000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
45	0:00:00.505000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
46	0:00:00.550000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
47	0:00:00.595000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
48	0:00:00.640000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
49	0:00:00.685000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
50	0:00:00.730000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
51	0:00:00.775000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
52	0:00:00.820000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
53	0:00:00.865000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
54	0:00:00.910000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
55	0:00:00.955000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
56	0:00:01.000000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
57	0:00:01.045000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
58	0:00:01.090000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
59	0:00:01.135000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
60	0:00:01.180000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
61	0:00:01.225000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
62	0:00:01.270000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
63	0:00:01.315000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
64	0:00:01.360000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105
65	0:00:01.405000	Vmware_7f:2c:4a	Broadcast	ARP	42 Who has 192.168.1.105 Tell 192.168.1.105

nmap ha fatto in realtà ARP scan.

Alcune macchine potrebbero non rispondere ai pacchetti di ping (come i firewall), rispondere invece all'ARP è inevitabile e necessario pertanto nmap invia delle richieste ARP e non di ping.

Nel caso di scansioni fatte su reti esterne non locali, non potendo usare l'ARP manderà dei pacchetti di SYN sulle porte 80 e 443 (solitamente quelle aperte più frequentemente).

Ad esempio potremmo voler forzare nmap ad effettuare un ping scan usando il comando:

```
nmap -PE 192.168.1.0/24
```

No.	Time	Source	Destination	Protocol	Length Info
57	0:00:05.7451	192.168.1.254	192.168.1.104	ICMP	98 Echo (ping) request id=0x324e, seq=0/0, tt
58	0:00:05.7491391	192.168.1.104	192.168.1.254	ICMP	98 Echo (ping) reply id=0x324e, seq=0/0, tt
59	0:00:05.7490838	192.168.1.254	192.168.1.104	ICMP	98 Echo (ping) request id=0x354e, seq=0/0, tt
60	0:00:05.749326478	192.168.1.104	192.168.1.254	ICMP	98 Echo (ping) reply id=0x354e, seq=0/0, tt

Scan technique

```
--*-CLAWERULE. PLEASE RUN PATH TO EACH HOST
```

SCAN TECHNIQUES:

```
-sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans  
-sU: UDP Scan  
-sN/sF/sX: TCP Null, FIN, and Xmas scans  
--scanflags <flags>: Customize TCP scan flags  
-sI <zombie host[:probeport]>: Idle scan  
-sY/sZ: SCTP INIT/COOKIE-ECHO scans  
-sO: IP protocol scan  
-b <FTP relay host>: FTP bounce scan
```

PORT SPECIFICATION AND SCAN ORDER:

```
-p <port ranges>: Only scan specified ports  
Fv -n22 -n1-65535 -n 11-53 111 127 T-21-25 80 139 8080 800
```

- -sS: manda un pacchetto di SYN
- -sT: apre la connessione
- -sA: che senso ha inviare un pacchetto di ACK? In realtà nessuna nella maggior parte dei casi ma alle volte è un pacchetto permesso sul firewall quindi in casi particolari può tornare utile
- -sV dice all'host di dirmi qualcosa di più sul servizio che è in ascolto su quella porta instaurando la connessione con un SYN,SYN-ACK,ACK e poi una volta ottenute delle informazioni di livello superiore ad esempio tramite il protocollo SSH o SMB o HTTP chiude la connessione mandando un pacchetto di FIN

30/06/2023

❓ Cos'è un Intrusion Detection System?

Sono dei "firewall" con dei controlli più dettagliati. Vedremo Snort che è un IDS open-source.

Why an intrusion detection system?

- Network security mainly means PREVENTION
 - Physical protection for hardware
 - Passwords, access tokens, etc. for *authentication*
 - Access control list for authorization
 - Cryptography for secrecy
 - Backups and redundancy for authenticity
 - ... and so on

BUT ...

... Absolute security cannot be guaranteed!

What is an Intrusion Detection System?

- Prevention is suitable when
 - Internal users are trusted
 - Limited interaction with other networks
- Need for a system which acts when prevention fails

Intrusion Detection System

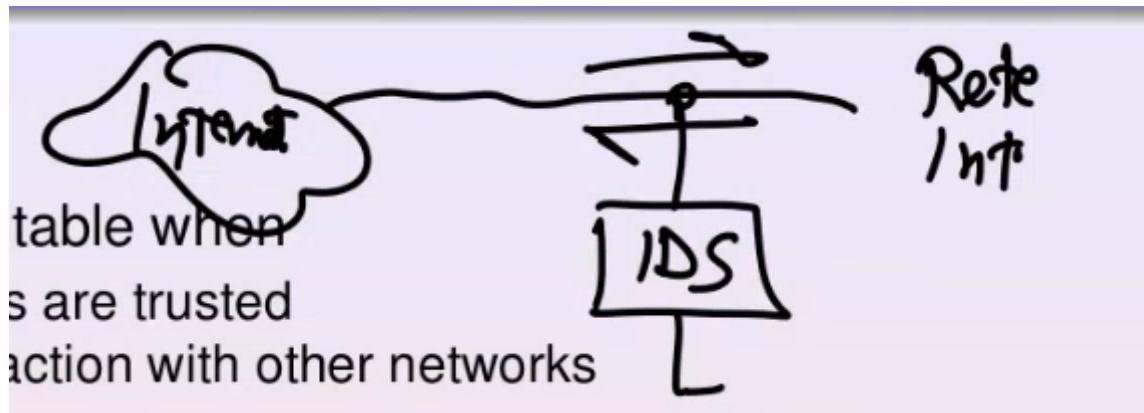
An intrusion detection system or IDS is a software/hardware tool used to detect unauthorized access to a computer system or a network

Quando la prevenzione è efficiente?

1. Quando gli utenti interni sono fidati
2. Quando ci sono poche interazioni con le altre reti

La differenza tra IDS e IPS sta nel modo di agire di questi sistemi o meglio come sono posizionati rispetto al traffico.

Se supponiamo di avere la nostra rete collegata ad Internet, l'IDS rileva solamente se ci sono attacchi sul traffico da/verso Internet:



L'IPS si pone invece in serie e non in parallelo tra le 2 reti, dunque l'IPS potrà far passare il traffico legittimo e scartando il traffico malevolo. IPS e IDS usano dunque algoritmi analoghi per analizzare il traffico ma la loro differenza dipende dal modo di agire e dalla loro posizione.

Questo primo lavoro del 1980 è la base teorica dell'intrusion detection. Gli intruder (attaccanti) possono essere categorizzati in 3 categorie distinte:

A taxonomy of the intruders

Intruders can be classified as

- **Masquerader:** an individual who is not authorized to use the computer and who penetrates a system's access control to exploit a legitimate user's account
- **Misfeasor:** a legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access, but misuses his/her privileges
- **Clandestine User:** an individual who seizes supervisory control of the system and uses the control to evade auditing and access controls or to suppress audit collection

Anderson, Computer Security Monitoring and Surveillance, Tech Rep 1980

Esistono inoltre diversi tipi di intrusione come descritte da Dorothy Denning nel 1997:

A taxonomy of the intrusions

- **Eavesdropping and Packet Sniffing:** passive interception of network traffic
- **Snooping and Downloading**
- **Tampering and Data Diddling:** unauthorized changes to data or records
- **Spoofing:** impersonating other users
- **Jamming or Flooding:** overwhelming a system's resources
- **Injecting Malicious Code**
- **Probing**
- **Exploiting Design or Implementation Flaws:** as buffer overflow
- **Cracking Passwords and Keys**

Denning, Cyberspace Attacks and Countermeasures, New York, 1997

- Eavesdropping and Packet Sniffing: con Wireshark possiamo catturare i pacchetti nella rete ad esempio
- Snooping and Downloading: si intende la cattura non solo dei singoli pacchetti ma anche riuscire a sbirciare sulla tastiera mentre un collega digita la password oppure mentre scrive una mail per ottenere delle informazioni puntuali
- Tampering and Data Diddling: impersonificazione di altri utenti o macchine tramite il cambio dell'indirizzo sorgente dei pacchetti che vado a generare
- Jamming e Flooding: attacchi di tipo DOS in cui vado a consumare tutte le risorse di calcolo di un computer
- Cracking di password e chiavi: esistono delle librerie che usano dei dizionari comuni
- Probing: nmap è un esempio di strumento che permette di scoprire informazioni riguardo il OS, le sue vulnerabilità e i servizi aperti

Gli IDS sono classificabili in base a una serie di parametri.

Host-based vs network-based

Host based VS Network based

Host based IDS

- Aimed at detecting attacks related to a specific host
- Architecture/Operating system dependent
- Processing of high level information (e.g. system calls)
- Effective in detecting insider misuse

Network based IDS

- Aimed at detecting attacks towards hosts connected to a LAN
- Architecture/Operating system independent
- Processing data at lower level of granularity (packets)
- Effective in detecting attacks from the “outside”

Misuse-based vs Anomaly-based

Quali meccanismi utilizzo per capire se sono in presenza di un attacco?

Misuse based IDS VS Anomaly based IDS

Misuse based IDS

Signature-based rule-based

- Identifies intrusion by looking for patterns of traffic or of application data presumed to be malicious
- Pattern of misuses are stored in a database
- Effective in detecting only “known” attacks

Anomaly based IDS

- Identifies intrusions by classifying activity as either anomalous or normal
- Need a training phase to recognize normal activity
- Able to detect “new” attacks
- Generates more false alarms than a misuse based IDS

Stateless vs Stateful

Stateless IDS VS Stateful IDS

Stateless IDS

- Treats each event independently of others
- Simple system design
- High processing speed

Stateful IDS

- Maintains information about past events
- The effect of a certain event depends on its position in the events stream
- More complex system design
- More effective in detecting distributed attacks

Centralized vs Distributed

Centralized IDS VS Distributed IDS

Centralized IDS

- All the operations are performed by the same machine
- More simple to realize
- Only one point of failure

Distributed IDS

- Composed of several components
 - **Sensors** which generate security events
 - **Console** to monitor events and alerts and control the sensors
 - Central **Engine** that records events and generate alarms
- May need to deal with different data formats
- Need of a secure communication protocol (IPFIX)

Abbiamo visto tanti tipi di IDS ma al giorno d'oggi l'interesse maggiore è per quelli network based in quanto sono gli unici efficaci per identificare attacchi DDOS inoltre l'interesse è verso quelli di tipo

Misuse Based in quanto la maggior parte degli attacchi sono ben noti e presenti in internet al netto di quelli nuovi (0-day) in effetti Snort è misuse-based.

IDS State of the Art

- Focus is on Network based IDSs (The only ones effective in detecting Distributed Denial of Service - DDoS)
- State of the art IDSs are Misuse Based
 - Most attacks are realized by means of software tools available on the Internet
 - Most attacks are “well-known” attacks

BUT ...

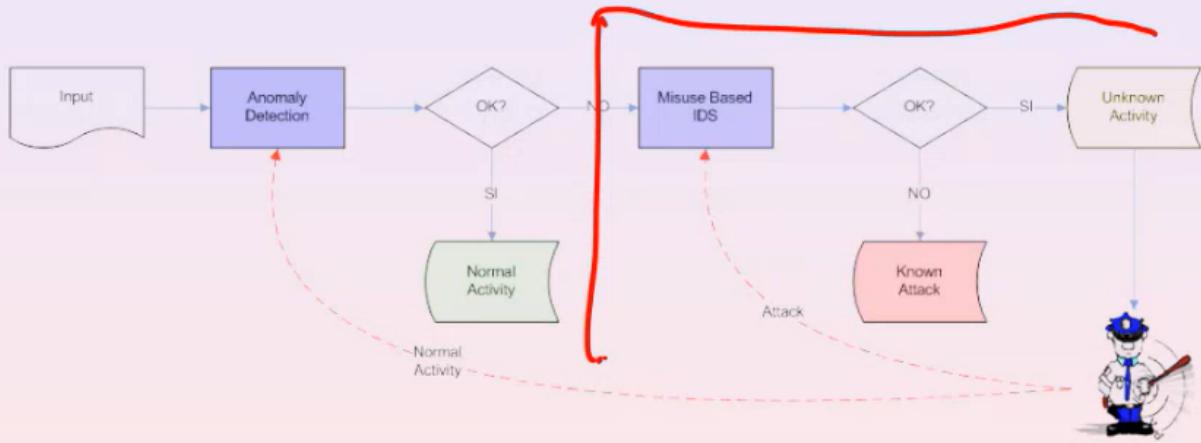
... The most dangerous attacks are those written ad hoc by the intruder!

The best choice?

- Combined use of both
 - HIDS (for insider attacks) & NIDS (for outsider attacks)
 - Misuse IDS (low False Alarm rate) & Anomaly IDS (for “new” attacks)
 - Stateless IDS (fast data process) & Stateful IDS (for “complex” attacks)
- Distributed IDS
 - Not a single point of failure
 - More effective in monitoring large networks

Pensando ad un network IDS l'input potrà essere il file .pcap, a cui applico l'algoritmo di anomaly detection. Se usassi un anomaly based tutta la parte in rosso sarebbero attacchi.

The best choice?



A questo sottoinsieme di traffico potrei applicare un algoritmo misuse-based.

Honey-Pot

- A honey-pot is a “trap” set to detect, deflect or counteract attempts of intrusions
- It generally consists of a computer, data, or a network site that appears to be part of a network, but which is actually isolated
- It seems to store important “information” which can be easily get by an intruder
- It is not “protected”
- It is potentially risky for the “real” network
- There exist some Honey-Pot detection systems

Ecco alcune definizione di base statistiche sugli allarmi, intrusioni e probabilità di falsi negativi e positivi:

Basic definitions

- $A = \text{alarm}$
- $\neg A = \text{not an alarm}$

- $I = \text{attack (intrusion)}$
- $\neg I = \text{not an attack}$

• $P(A|I) = \text{False positive probability}$

• $P(\neg A|I) = \text{False negative probability}$

$P(A|I)$ = *False Alarm*
 $P(A|I)$ = *detection Rate*

Basic probabilistic relations

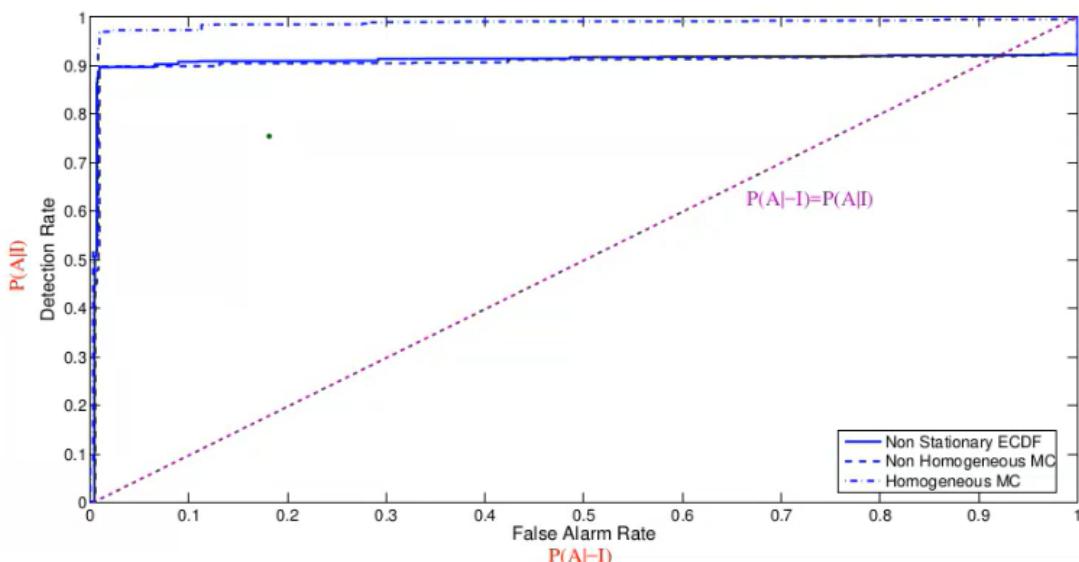
- $P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$
- $P(B) = \sum_i P(A_i) \cdot P(B|A_i)$

$\cup A_i$ = *evento certo*

Vorrei avere dunque un detection rate alto con un basso livello di falsi allarmi.

Questo aspetto viene caratterizzato tramite la curva ROC:

ROC (Receiver Operating Characteristics) Curve



sull'asse delle x abbiamo il rate di falso allarme mentre sull'asse delle y abbiamo il detection (o hit) rate.

Idealmente vorrei avere un detection rate pari a 1 con un false alarm rate pari a 0.

L'equazione della diagonale del piano è data dalla probabilità indicata su di essa, ovvero la probabilità

con cui genero un allarme sia che c'è intrusione che non c'è è sempre la stessa. Dunque voglio le curve stiano al di sopra della diagonale e idealmente raggiungano il punto (0,1).

Other performance indexes

- Detection Rate (DR)

$$DR = \frac{TP}{TP + FN}$$



- False Alarm Rate (FAR)

$$FAR = \frac{FP}{FP + TN}$$

- Accuracy (ACC)

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

- Execution time

L'accuracy sarà dunque il numero di volte che do una risposta corretta diviso il numero totale di eventi. Il tempo di esecuzione dell'algoritmo per poter produrre una risposta è fondamentale dunque considererò anche questo in fase di analisi.

Base Rate Fallacy

Let's suppose:

- $P(A|I) = 0.99$
 - $P(\neg A \mid \neg I) = 0.99$
 - we have 2 attacks a day over 10^6 pkts
- } accuratezza del 99%

$$\text{base rate} = P(I) = 1/500000$$

Applying the Bayes theorem:

$$\begin{aligned} P(I|A) &= \frac{P(I) \cdot P(A|I)}{P(I) \cdot P(A|I) + P(\neg I) \cdot P(A|\neg I)} = \\ &= \frac{1/500000 \cdot 0.99}{1/500000 \cdot 0.99 + (1 - 1/500000) \cdot 0.01} \end{aligned}$$

Thus $P(I|A) = 0.0002$

Supponiamo che il nostro sistema abbia un accuratezza del 99% e che la maggior parte del traffico sia lecito.

Il base rate è il rate con cui ricevo un pacchetto malevolo ovvero la probabilità di essere in presenza di un attacco.

Come si può vedere, nonostante i livelli elevati di accuratezza, dal momento che l'attacco è un evento raro la maggior parte degli allarmi sarà legato a un evento lecito.

Un pò di storia...

A bit of History

- ① First Generation IDSs (end of the 1970s)
 - The concept of IDS first appears in the 1970s and early 1980s (Anderson, Computer Security Monitoring and Surveillance, Tech Rep 1980) //
 - Focus on audit data of a single machine
 - Post processing of data
- ② Second Generation IDSs (1987)
 - Intrusion Detection Expert System (Denning, An intrusion Detection Model, IEEE Trans. on Soft. Eng., 1987)
 - Statistical analysis of data
- ③ Third Generation IDSs
 - Focus on the network
 - Real-time detection
 - Real-time reaction
 - Intrusion Prevention System

Snort

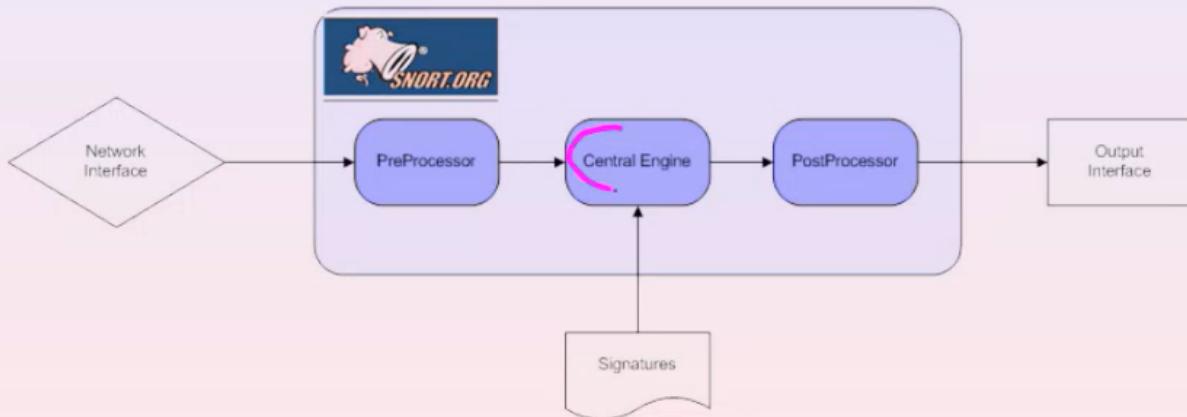
SNORT

Snort

- The most famous IDS
- Open source software tool
- Network based
- Signature based (Histogram based)
- Centralized architecture



- Spade (Statistical Packet Anomaly Detection Engine), the anomaly detection plug-in for Snort, is not supported any longer
- The rules database, as well as the system code, is available for download at the web site <http://www.snort.org>



L'architettura di Snort prevede un engine centrale che si poggia ad un database con le signature e le regole.

Il PreProcessor include una prima analisi del traffico tra cui un approccio stateful nel caso ad esempio di un port scan in quanto mandare un pacchetto verso una porta non è strano ma lo diventa se questi pacchetti vengono mandati continuamente verso diverse porte nella rete.

SNORT Architecture

Pre-Processor

- First security check
- Stateful approach (e.g. Port Scan)

Post-Processor

- Alarm generation
- It is possible to choice which action should be done by the system

Central Engine

- Check for known patterns (pkt level/flow level)
- Rules are organized in tree structures
- 80% of the total processing time

Esiste un file generale di configurazione chiamato snort.conf

SNORT Rules

Snort follows a “Unixy” configuration philosophy

- Configuration is plaintext
- Powerful and complex
- Snort configuration consists of:
 - Global configuration (snort.conf)
 - Optional *.rules file(s)

Il file snort.conf è dichiarato in questo modo:

SNORT Rules

snort.conf

```
ipvar HOME_NET 192.168.3.0/24
ipvar EXTERNAL_NET !$HOME_NET
ipvar DNS_SERVERS [192.168.3.1,192.168.3.10]
ipvar HTTP_SERVERS
[192.168.3.1,192.168.3.2,192.168.3.88]
portvar HTTP_PORTS 80
var RULE_PATH /usr/local/snortrules
...
include $RULE_PATH/local.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/bleeding-all.rules
Include $RULE_PATH/community-bot.rules
```



esistono 3 tipi di variabili:

- ipvar: variabili che fanno riferimento ad indirizzi IP
- portvar: variabili che fanno riferimento a porte di rete
- var: variabili di altro tipo

Introdurre le variabili facilita di molto la configurazione o modifica del file di configurazione.

Una regola include due parti: un header e un body.

```
    .  
    alert tcp any any -> any any (msg: ``Sample alert'';)
```

Header contains the following fields:

- Action (log, alert, pass, ...)
- Protocol (ip, tcp, udp, icmp, any)
- Src IP & Port
- Dst IP & Port
- Direction operator (“->”, “< >”)

The body is usually the complex part

- Begins and ends with “()”
- Series of “rule options” (keywords, with optional parameters) separated by “;”



- **Header:** contiene una parola che si riferisce all'azione, un campo protocol per poter filtrare il traffico, il primo any fa riferimento all'IP sorgente, il secondo any fa riferimento all'IP destinazione, poi la direzione che può essere unidirezionale -> o bidirezionale <>
- **Body:** è il corpo delle regole in blu che contiene una parola chiave come in questo caso msg che vuol dire che viene generato un allarme ogni volta che riceviamo un pacchetto TCP.

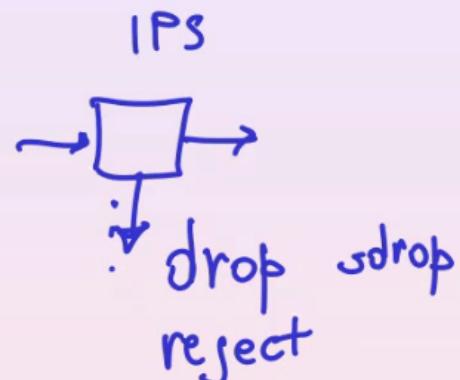
07/07/2023

SNORT Rules

```
alert tcp any any -> any any (msg: "Sample alert";)
```

Header contains the following fields:

- Action (log, alert, pass, ...)
- Protocol (ip, tcp, udp, icmp, any)
- Src IP & Port
- Dst IP & Port
- Direction operator ("->", "< >")



The body is usually the complex part

- Begins and ends with "()"
- Series of “rule options” (keywords, with optional parameters) separated by “;”

Esistono 3 regole per poter scartare i pacchetti nel caso di IPS:

- **drop**: scarta il pacchetto e genera il log relativo all'evento
- **reject**: scarta il pacchetto, genera il log MA risponde anche al mittente (ad esempio se si tratta di un pacchetto TCP chiude la connessione oppure manda un messaggio di Host Unreachable usando ICMP)
- **sdrop**: blocca il pacchetto critico ma non crea nessun log

le regole possono essere specificate usando 5 tipi di opzioni:

SNORT Rules

Five types of options

- Metadata
- Payload detection
- Non-payload detection
- Post-detection
- Thresholding and suppression

SNORT Rules

```
alert tcp $EXTERNAL_NET any -> 192.168.3.0/24 80
(msg: "Sample alert"; classtype:
web-application-activity;
reference:url,http://www.vorant.com/advisories/20060405.html;
sid:2000123; rev:1;)
```

- Use of “classtype” implies a default priority for each class
- Defaults for each class are in the manual
- Use the “priority” option to override these
- Each sid must be unique
- Choose a sid range > 4,000,000 to avoid conflicts with popular rule providers

Questa regola mi dice che un qualunque pacchetto che provenga dalla rete esterna, qualunque numero di porta, verso la rete /24, diretti verso la porta 80 viene generato un alert. *classtype* in questo esempio indica traffico di tipo web.

Se si vuole creare una regola conviene usare un ID con valore maggiore di 4000000 per evitare conflitti con i provider di regole.

SNORT Rules

- **Metadata:** provide snort with information about the rule itself or pass on information to the analyst e.g.:
 - “msg” specifies the human-readable alert message
 - “reference” includes a URL for more info
 - “classtype” and “priority” give some idea about the type of attack and the severity of the event
 - “sid” and “rev” uniquely identify the rule (including revisions & edits)
- **Payload detection:** Look inside the packet payload (not the packet headers)
 - “content” looks for a string of bytes
 - “nocase” modified content, makes it case insensitive
 - “offset” skips a certain number of bytes before searching
 - “pcre” allows the use of Perl-compatible regular expressions (support must be compiled in)

SNORT Rules

```
alert tcp $EXTERNAL_NET any -> 192.168.3.0/24 80
(msg: ``Sample alert'';
content: ``http|3a| //www.vorant.com/test.cgi?id=pwn3d'' ;
nocase;
offset:12; classtype: web-application-activity;
reference:url,http://www.vorant.com/advisories/20060405.html;
sid:2000123; rev:1;)
```

- Looks for the case-insensitive string “http://www.vorant.com/test.cgi?id=pwn3d” in all traffic matched by the rule header
- Skips the first 12 bytes of each packet before starting search, for efficiency
- Note inclusion of hex ASCII code for the “:”
- The “|3a|” notation is good for non-printable data (or “：“, which must not be used in content match)

il 3a è il codice ASCII tra pipe che sostituisce i [:] che è un carattere speciale di snort usato infatti per separare il nome dal parametro dal suo valore.

Questa regular expression ci dice che l'alert verrà generato sulle richieste di GET di una pagina htm

oppure html. il `.*` indica qualsiasi carattere ripetuto più volte mentre il `\.` serve per fare l'escaping del punto che fa parte dell'estensione del file (.htm o .html) che vogliamo cercare:

SNORT Rules

```
alert tcp $EXTERNAL_NET any -> 192.168.3.0/24 80
(msg: "Sample alert"; pcre: "/GET.*\.htm/i";
classtype: webapplication-activity;
reference:url,http://www.vorant.com/advisories/20060405.html;
sid:2000123; rev:1;)
```

- Alerts on all GET requests for an HTML page (.htm or .html both work)
- “i” option to pcre asks for case-insensitive matching
- A simple content match could be used, but sometimes...
 - content is not flexible enough to match the data
 - a single PCRE may be more clear than a bunch of individual content matches



Ecco un esempio di opzione non-payload:

SNORT Rules

```
alert tcp $EXTERNAL_NET any -> 192.168.3.0/24 80
(msg: "Sample alert"; flow: to_server,established;
pcre: "/GET.*\.htm/i"; classtype: web-application-activity;
reference:url,http://www.vorant.com/advisories/20060405.html;
sid:2000123; rev:1;)
```

- Technically a “non-payload” option
- “established” option specifies that the rule only alerts on valid TCP sessions
- “to_server” option further restricts matching to packets going to the “server”
- Snort assumes the “client” is the session initiator, so the server is the recipient



Snort in questo caso è in grado di costruirsi internamente la macchina a stati del TCP. Siamo nella fase

[established] dunque non consideriamo il 3-way handshake o la fase finale di chiusura e inoltre essendo il flow "to_server" ci interessano solamente i pacchetti inviati dal client al server.

? Come fa SNORT a cercare dei pattern di attacco all'interno dei payload?

String searching algorithm

- In a signature based IDS the content of each packet (headers+payload) is inspected to find “known” strings
- There is the need of an efficient string matching algorithm

String Matching Algorithm

A String searching algorithm, sometimes called string matching algorithm, is a string algorithm which tries to find a place where one or several strings (also called patterns) are found within a larger string or text

Ci vuole un algoritmo di ricerca delle stringhe all'interno di un testo largo.

Esistono 2 tipologie di algoritmi di ricerca, il primo fa riferimento alla ricerca di un pattern di una singola stringa e algoritmi multipattern in grado di cercare in parallelo più stringhe. L'algoritmo più famoso di

questa seconda tipologia è l'algoritmo di Aho-Corasik.

String searching algorithm

Mono-Pattern Algorithm

Effective for dictionaries of not more than 10 patterns
Serial test
Boyer-Moore

Multi-Pattern Algorithm

Effective for dictionaries of more than 10 patterns
Parallel test
Aho-Corasick

Algoritmo di Aho-Corasick

Si basa sulla costruzione di un albero e di muoversi all'interno dell'albero mentre scandisco i caratteri del mio testo target.

Aho-Corasick Algorithm

A classical algorithm is the Aho-Corasick Algorithm

- It solves the **Exact Set Matching Problems**
- It locates occurrences of any pattern of a set $P = \{P_1, \dots, P_k\}$ in target $T[1 \dots m]$
- It is based on the refinement of a **keyword tree** K
 - Each edge of K is labeled by a character
 - Any two edges out of a node have different labels

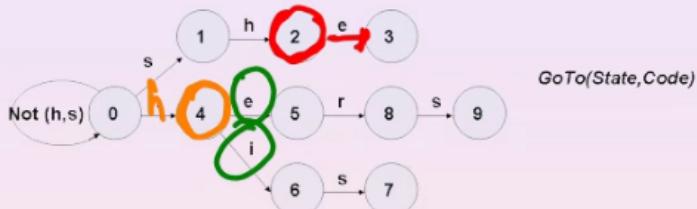
A.V. Aho and M.J. Corasick, "Efficient string matching: An aid to bibliographic search",
Communications of the ACM 18

Supponiamo che il pattern da cercare sia questo `{she, his, her, hers}`:

Aho-Corasick Algorithm Working

Let's $P = \{\text{she}, \text{he}, \text{his}, \text{hers}\}$

shX



State	1	2	3	4	5	6	7	8	9
Fail	0	4	5	0	0	0	1	0	1
State	3	5	7	9					
Output	She, he	he	his	hers					

$T = 'sihe'$

- GoTo(0,'s')=1
- GoTo(1,'i')=Fail
- Fail(1)=0
- GoTo(0,'i')=0

$T = 'sihe'$

- GoTo(0,'h')=4
- GoTo(4,'e')=5
- Output(5)=he

M. Pagano

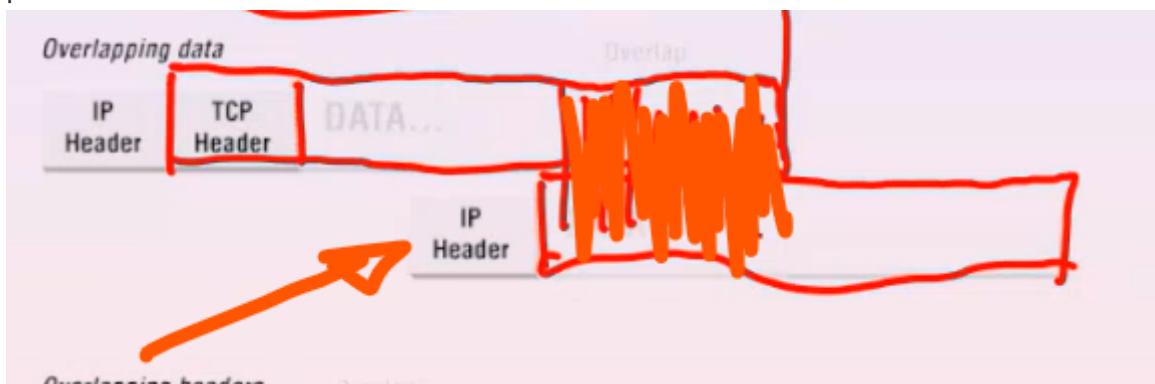
IDS

35 / 61

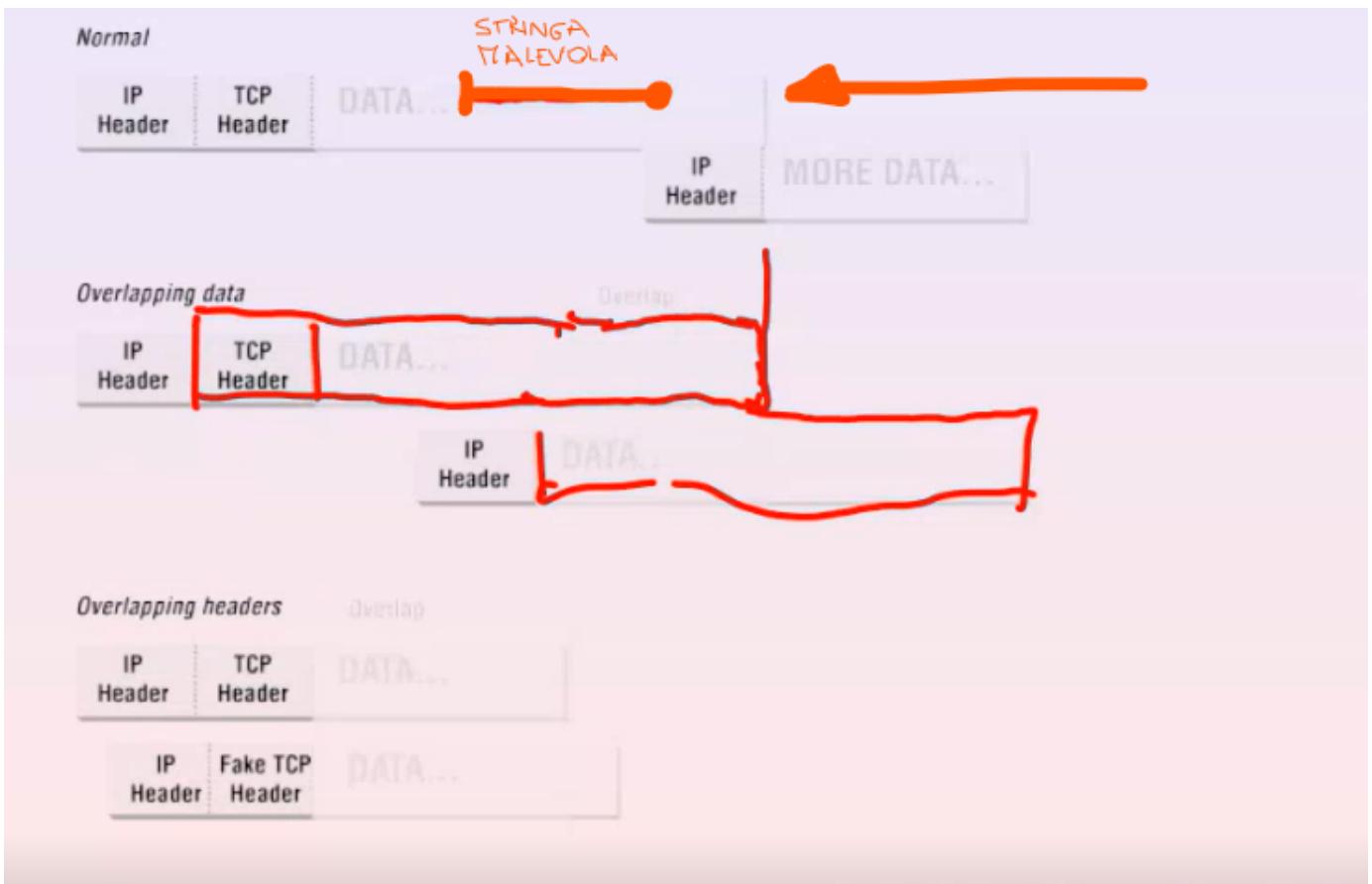
L'algoritmo si basa sulla creazione di un albero delle keywords che devo cercare e poi avremo due tabelle, una che mi dice cosa fare in caso di fail e in particolare mi dice in che stato tornare. L'ultima tabella è quella che mi dà le uscite ovvero quando raggiungo i vari stati significa che ho ottenuto almeno un output voluto per ciascuno stato in cui mi trovo.

Overlapping Attack

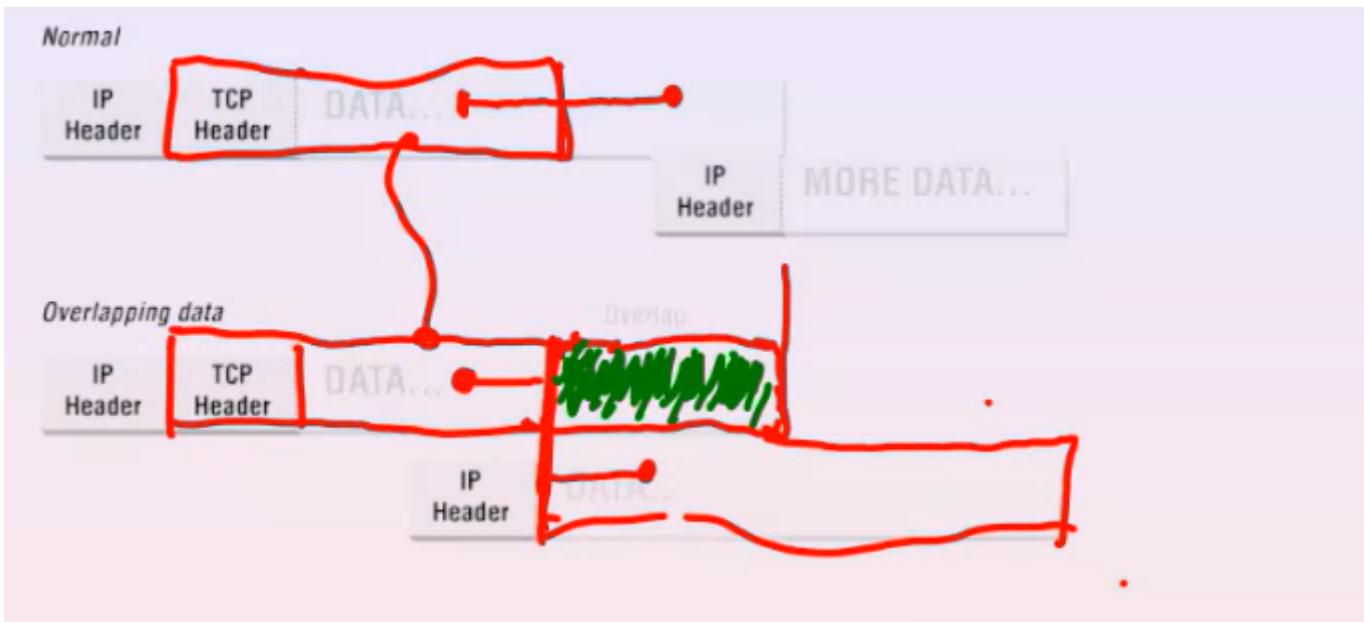
L'MTU in IPv4 mi permette di frammentare un pacchetto troppo grande. Nell'Header IPv4 ci sono le informazioni per sapere qual'è l'ultimo frammento dal quale devo riprendere a trasmettere i dati. Un attaccante potrebbe fare un'operazione per ingannare eventuali firewall o IDS dove il primo frammento sembra somigliare a quello iniziale mentre il secondo frammento ha una sovrapposizione di questa parte di dati in arancione:



Il senso di questo attacco è semplice. Supponiamo che nel pacchetto originale ci sia una stringa malevola che si trovi nella posizione in figura:



l'attaccante nel secondo caso, fino alla posizione da cui riprende il pacchetto sottostante ricopia i dati reali mentre nella rimanente parte del frammento ci mette dei byte casuali (quelli in verde):



dunque cambiando il contenuto in verde impedisco all'IDS di trovare la stringa malevola nel mio pacchetto. Quando l'IDS analizzerà il secondo frammento troverà solo la seconda parte della stringa che però non è malevola dunque passerebbero.

Tuttavia quando il ricevitore riceve i due frammenti, legge l'offset della seconda parte e posiziona i byte ricevuti con il secondo pacchetto al posto di quelli in verde e quando ricostruisce il pacchetto otterrà il

pacchetto originario con tutti i dati uniti:



Anomaly detection

Ovvero la costruzione di un modello del comportamento normale dell'host o della rete per fare poi un confronto con quello che monitoro.

Metrics and Models

Metrics

- *Event counter* ↗
- *Interval timer* ↗
- *Resource measure* ↗

Statistical models

- *Operational model*: abnormality is decided by comparing observations x_n of a RV x (representing a quantitative measure accumulated over a period) with a fixed threshold
- *Mean and standard deviation model*: abnormality is decided by checking if x_n falls inside the confidence interval
- *Multivariate model*: based on the correlations between two or more metrics
- *Markov process model*: based on the transition probabilities
- *Time series model*: takes into account order and inter-arrival time of the observations

Costruisco un modello dell'host o della rete e confronto il traffico in live con il modello costruito.

Possiamo avere un contatore di eventi per particolari tipologie di pacchetti oppure il tempo di intervallo tra pacchetti con certe caratteristiche.

Oppure ancora la misura delle risorse.

Statistical Approach: Traffic Descriptors

To identify some traffic parameters, which can be used to describe the network traffic and that vary significantly from the normal behavior to the anomalous one

Some examples

- Packet length
- Inter-arrival time
- Flow size
- Number of packets per flow
- ... and so on

Choice of the Traffic Descriptors

For each parameter we can consider

- Mean Value
- Variance and higher order moments
- Distribution function
- Quantiles
- ... and so on

The number of potential traffic descriptors is huge (some papers identify up to 200 descriptors)

GOAL

To identify as few descriptors as possible to classify traffic with an *acceptable* error rate

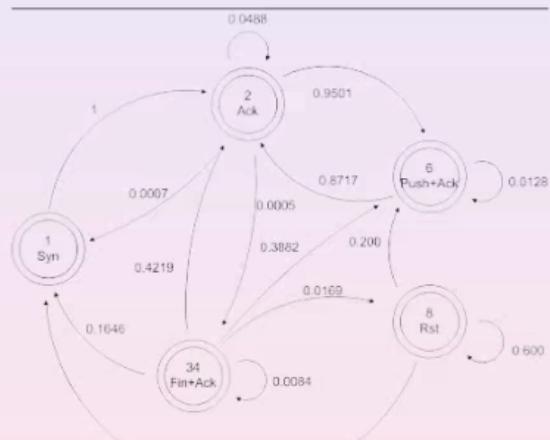
- The approach was first proposed by Denning and developed in the 1990s.
- Mainly used in two distinct environment
 - **HIDS**: to model the sequence of system commands used by a user
 - **NIDS**: to model the sequence of some specific fields of the packet (e.g. the sequence of the flags values in a TCP connection)
- The most classical approach: **Markov chains**
- In the last few years many works on **neural networks**

Markov Chain and TCP - Training phase

Calculate the transition probabilities

$$a_{ij} = P[q_{t+1} = j | q_t = i] = \frac{P[q_t = i, q_{t+1} = j]}{P[q_t = i]}$$

- Server side
- 3-way handshake
- psh flag
- closing



SSH Markov Chain

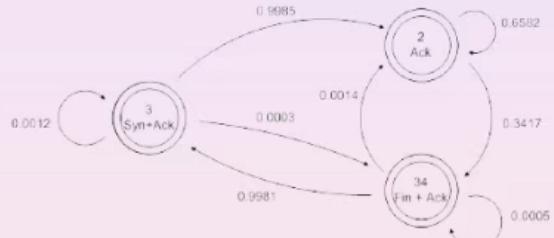
Markov Chain and TCP - Training phase

Calculate the transition probabilities

$$a_{ij} = \frac{P[q_{t+1} = j | q_t = i]}{P[q_t = i, q_{t+1} = j]} =$$

$$\frac{P[q_t = i]}{P[q_t = i]}$$

- Client side
- 3-way handshake
- ack flag
- closing



FTP Markov Chain

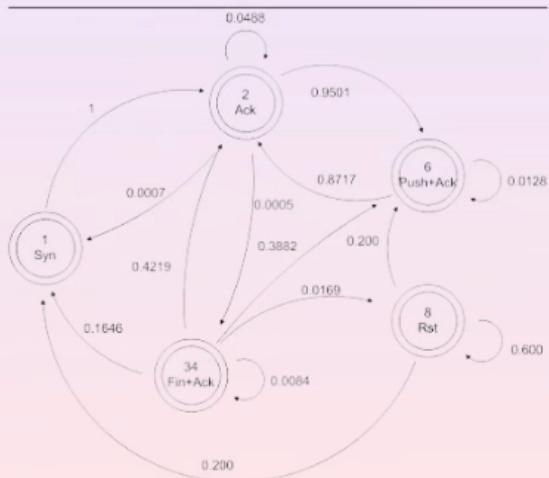
In generale bisogna sviluppare una catena di Markov per ciascuna applicazione sia lato server che lato client.

Markov Chain and TCP - Training phase

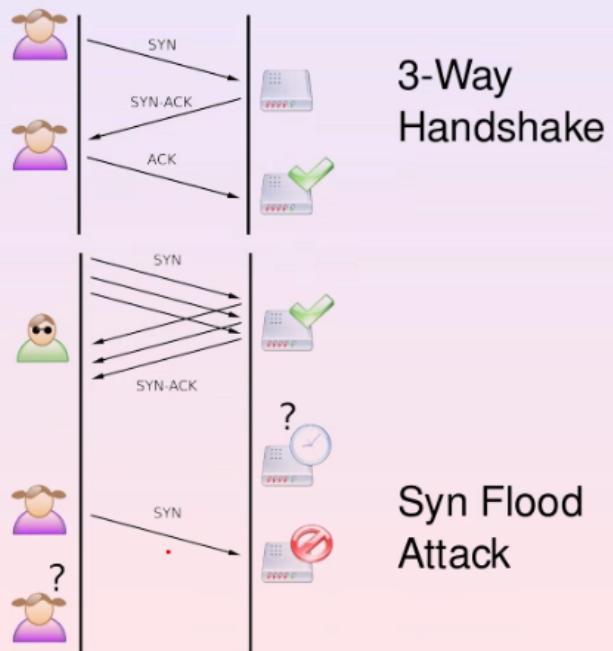
Calculate the transition probabilities

$$a_{ij} = \frac{P[q_{t+1} = j | q_t = i]}{P[q_t = i, q_{t+1} = j]} =$$

$$\frac{P[q_t = i]}{P[q_t = i]}$$



SSH Markov Chain



Syn Flood Attack

Poi bisognerà costruire la funzione di verosomiglianza:

- Given the observation $(S_{R+1}, S_{R+2}, \dots, S_{R+T})$
- The system has to decide between two hypothesis

H_0 : normal behaviour

H_1 : anomaly

- A possible statistic is given by the **logarithm of the Likelihood Function**

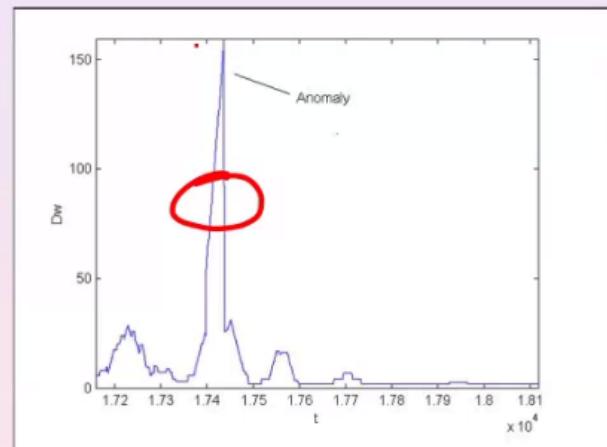
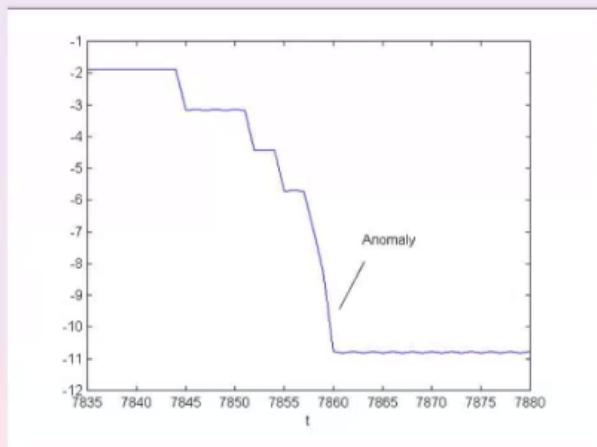
$$\text{LogLF}(t) = \sum_{t=R+1}^{T+R} \text{Log}(a_{S_t S_{t+1}})$$

- or by its **temporal “derivative”**

$$D_w(t) = \left| \text{LogLF}(t) - \frac{1}{W} \sum_{i=1}^W \text{LogLF}(t-i) \right|$$

e se c'è un'anomalia questa corrisponde a picchi o decrescite rapide:

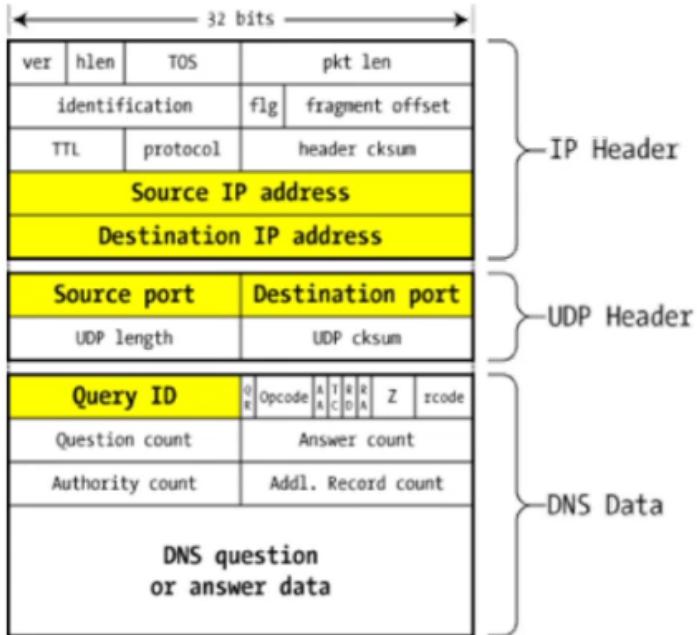
Markov Chain and TCP - Detection phase



DNS



DNS packet



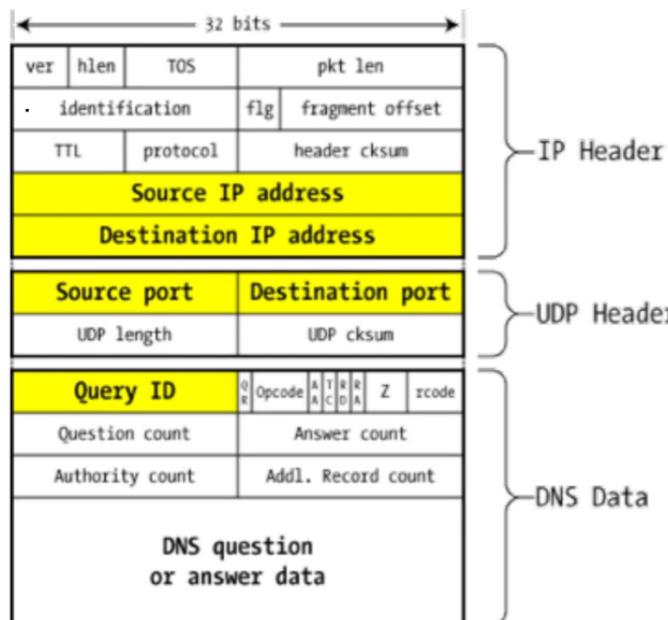
DNS packet on the wire

- **Query ID:** a unique identifier created in the query packet **QR (Query / Response):** Set to 0 for a query by a client, 1 for a response from a server
- **Opcode:** Set by client to 0 for a standard query; the other types aren't used in our examples
- **AA (Authoritative Answer):** Set to 1 in a server response if this answer is Authoritative, 0 if not
- **TC (Truncated):** Set to 1 in a server response if the answer can't fit in the 512-byte limit of a UDP packet response
- **RD (Recursion Desired):** The client sets this to 1 if it wishes that the server will perform the entire lookup of the name recursively
- **RA (Recursion Available):** The server sets this to indicate that it will (1) or won't (0) support recursion
- **Z: reserved**

I vari elementi restituiti da una response DNS sono denominati RESOURCE RECORD caratterizzati da un tipo (A corrisponde a un record IPv4).

Un pacchetto DNS è costituito dai seguenti campi:

- Query ID: un ID identificativo univoco della sessione di richiesta e risposta
- Authoritative Answer: nel caso in cui la risposta arriva da un server autoritativo per quel dominio sarà settato a 1
- Recursion Desired: indica al server DNS di riferimento di effettuare in autonomia l'intera ricerca ricorsiva
- Recursion Available: il server indica tramite questo bit se è disponibile per effettuare la ricerca ricorsiva oppure no

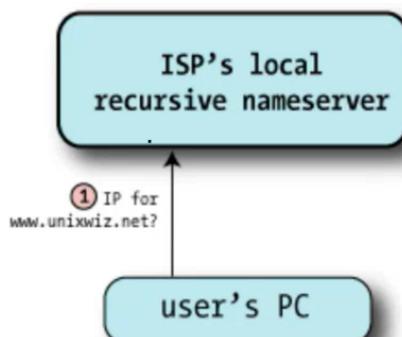


- **rcode:** Response code from the server: indicates success or failure
- **Question record count:** The client fills in the next section with a single "question" record that specifies what it's looking for: it includes the name (www_unixwiz.net), the type (A, NS, MX, etc.), and the class (virtually always IN=Internet)
- **Answer/authority/additional record count:** Set by the server, these provide various kinds of answers to the query from the client
- **DNS Question/Answer data:** This is the area that holds the question/answer data referenced by the count fields above

DNS packet on the wire

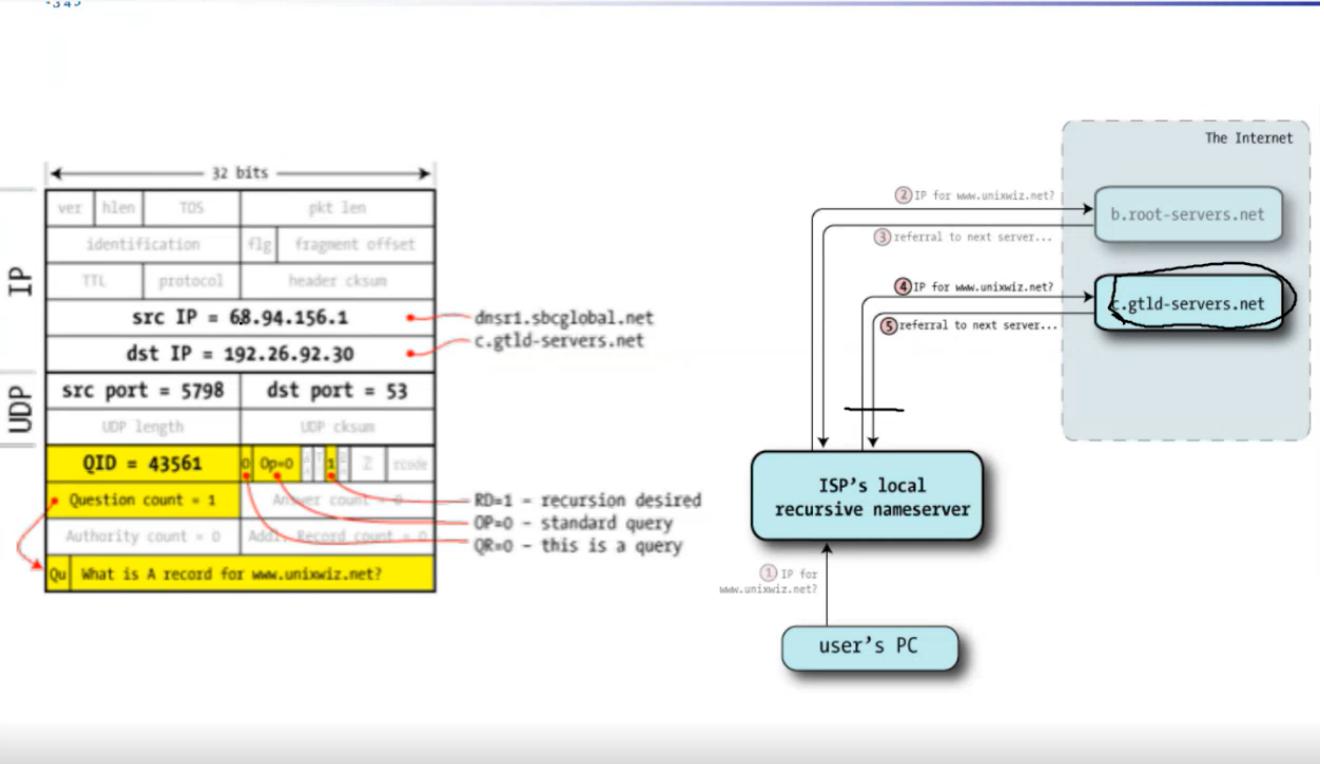
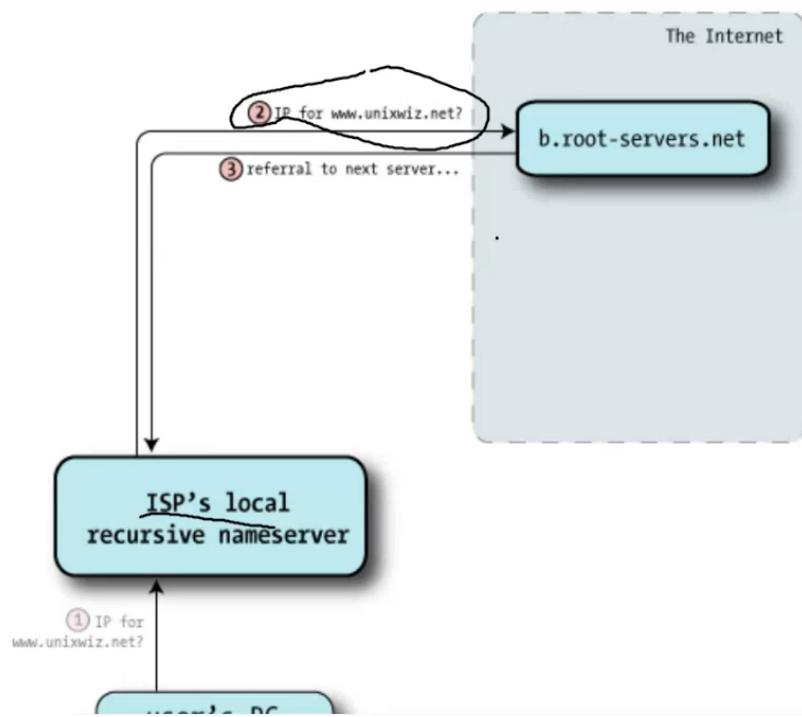


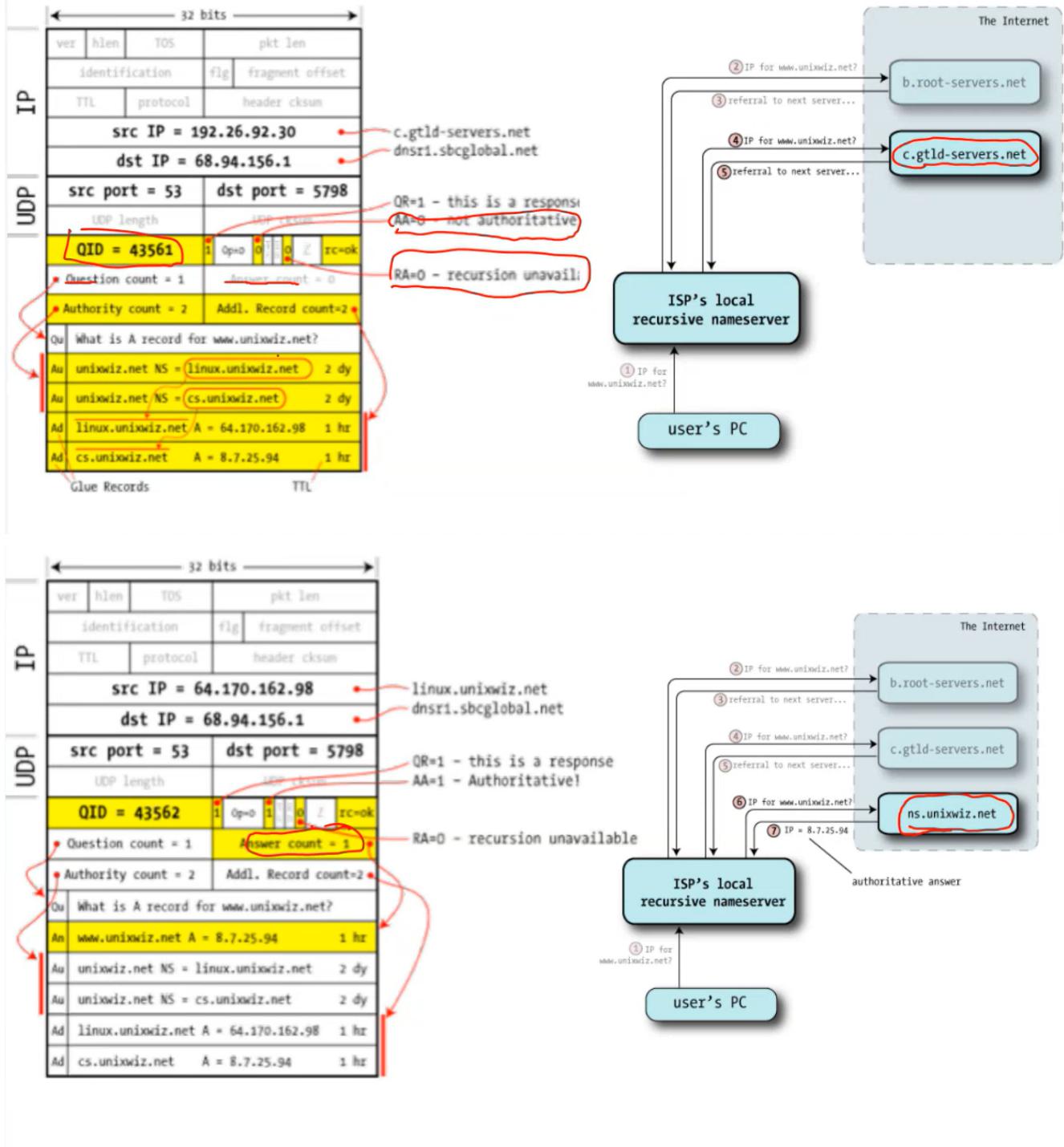
Standard DNS query





Standard DNS query





La cache evita di effettuare ogni volta una ricerca anche nel caso in cui quella richiesta DNS all'interno di una rete è stata già effettuata:



Standard DNS query

- **DNS Cache**

- Once we get an authoritative answer for a given name, we can save it in a local cache to use to satisfy future queries directly

- **DNS Cache TTL**

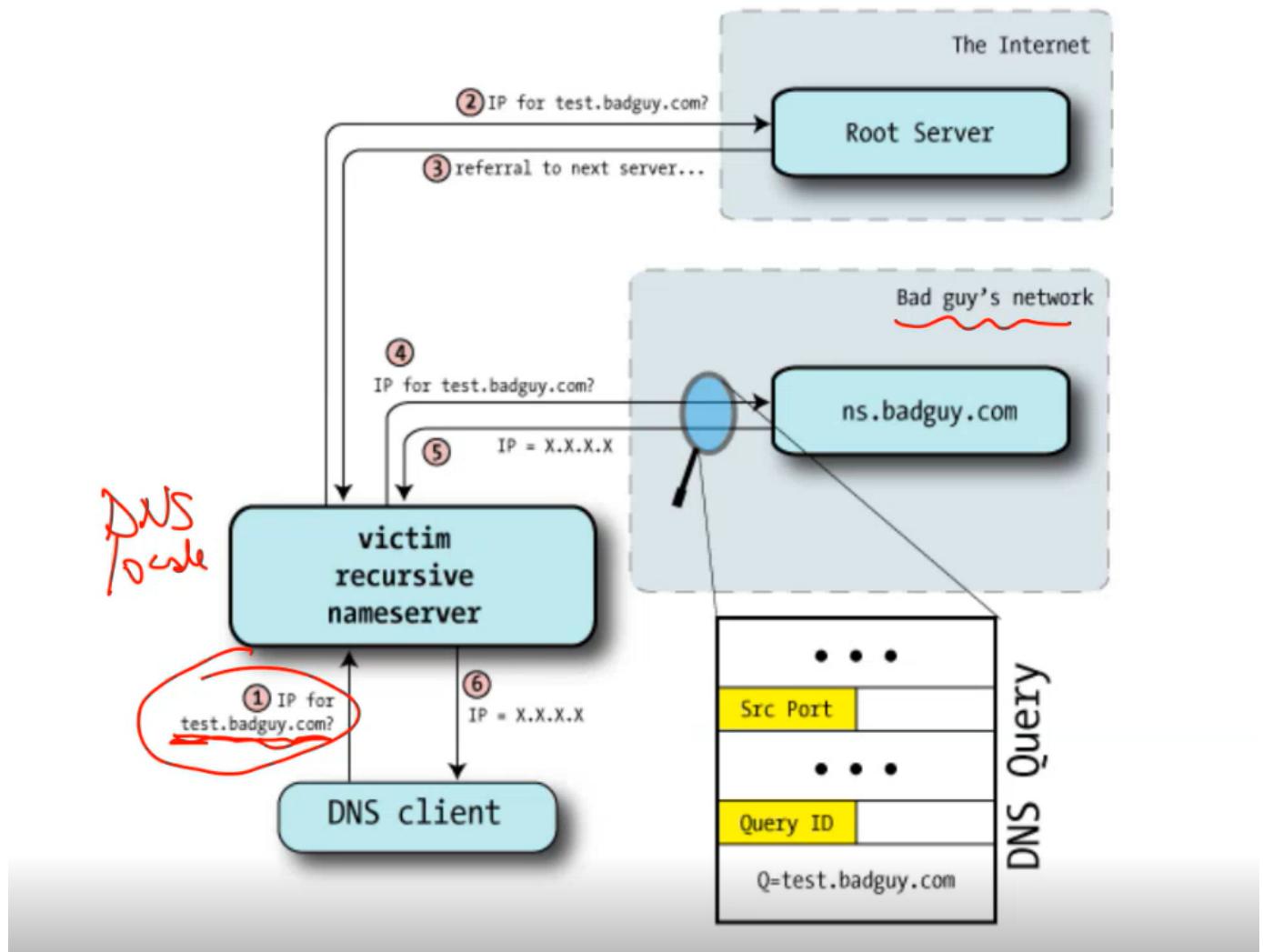
- Each entry in the DNS cache has a time-to-live, measured in seconds.
The administrator of the zone specifies this information for every resource record

Non necessariamente i DNS autoritativi devono far parte dello stesso dominio richiesto:

```
pagano@pagano-ThinkPad-X390-Yoga:~$ dig www.unipi.it +norecurse +trace +nodnssec  
  
; <>> DiG 9.16.1-Ubuntu <>> www.unipi.it +norecurse +trace +nodnssec  
;; global options: +cmd  
..  
77702 IN NS m.root-servers.net.  
77702 IN NS c.root-servers.net.  
77702 IN NS f.root-servers.net.  
77702 IN NS i.root-servers.net.  
77702 IN NS k.root-servers.net.  
77702 IN NS j.root-servers.net.  
77702 IN NS g.root-servers.net.  
77702 IN NS d.root-servers.net.  
77702 IN NS h.root-servers.net.  
77702 IN NS e.root-servers.net.  
77702 IN NS l.root-servers.net.  
77702 IN NS b.root-servers.net.  
77702 IN NS a.root-servers.net.  
;; Received 239 bytes from 127.0.0.53#53(127.0.0.53) in 88 ms  
  
it. 172800 IN NS d.dns.it.  
it. 172800 IN NS nameserver.cnr.it.  
it. 172800 IN NS a.dns.it.  
it. 172800 IN NS dns.nic.it.  
it. 172800 IN NS m.dns.it.  
it. 172800 IN NS r.dns.it.  
;; Received 460 bytes from 192.5.5.241#53(f.root-servers.net) in 20 ms  
  
unipi.it. 10800 IN NS ns2.unipi.it.  
unipi.it. 10800 IN NS ns1.garr.net.   
unipi.it. 10800 IN NS ns1.unipi.it.  
;; Received 163 bytes from 193.206.141.46#53(r.dns.it) in 20 ms  
  
www.unipi.it. 300 IN CNAME wwwnew2.unipi.it.  
wwwnew2.unipi.it. 300 IN A 131.114.21.42  
unipi.it. 86400 IN NS ns1.unipi.it.  
unipi.it. 86400 IN NS ns1.garr.net.  
unipi.it. 86400 IN NS ns2.unipi.it.  
;; Received 201 bytes from 131.114.21.10#53(ns1.unipi.it) in 28 ms
```

DNS Cache poisoning

L'attaccante non può fare in genere attacchi MITM ma può inviare dei pacchetti fake con le risposte errate agli host legittimi facendogli creare di aver ricevuto una risposta dal DNS autoritativo:

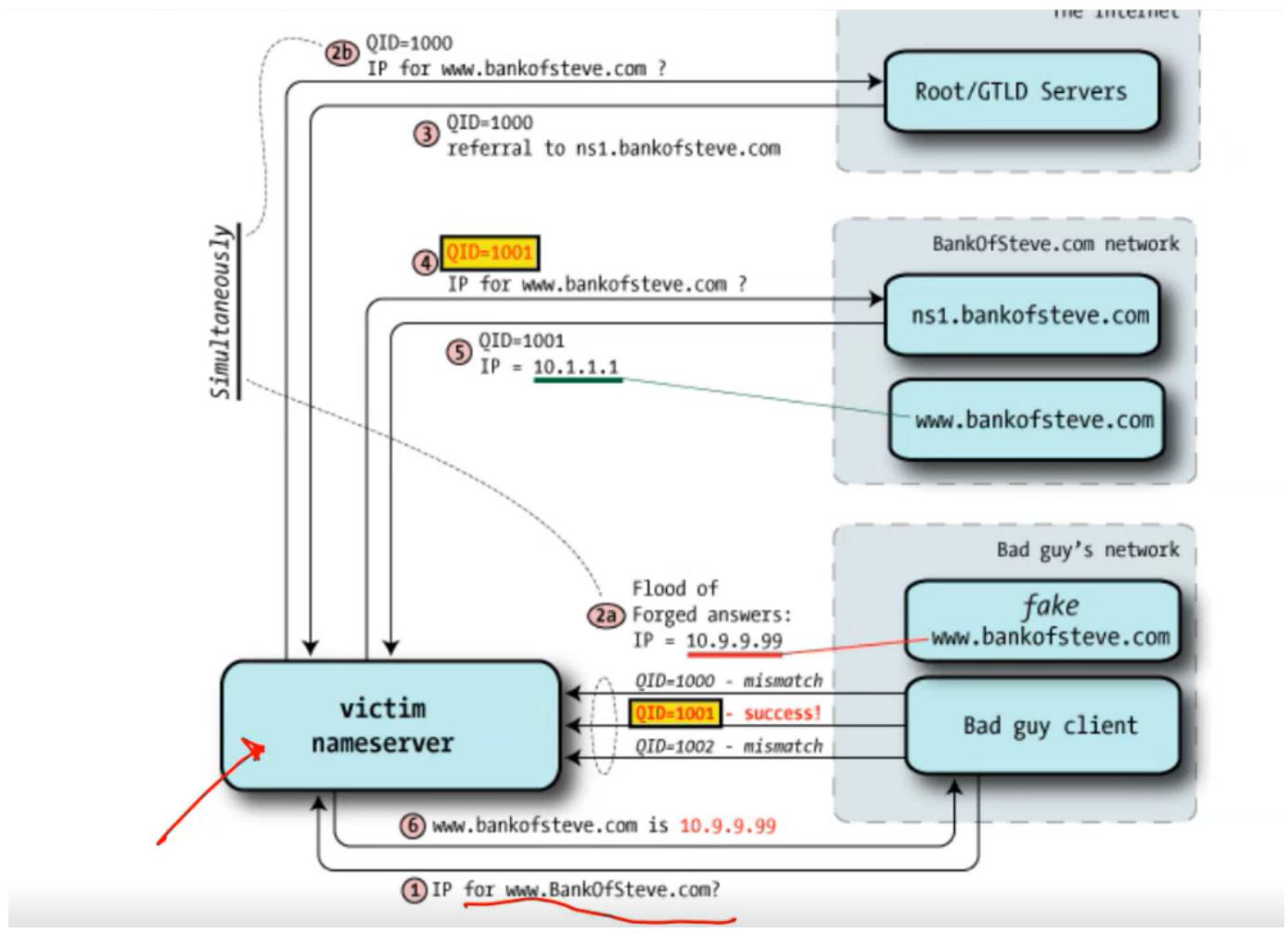


Per poterlo effettuare è necessario conoscere la porta sorgente dalla quale il client ha effettuato la richiesta DNS e la Query ID usata.

Supponiamo che gli attaccanti abbiano un nome di dominio di loro controllo (badguy.com) al cui interno è configurato un nameserver autorevole per il loro dominio (ns.badguy.com).

Possiamo chiedere al DNS locale, che è la nostra vittima in questo esempio, di risolvere un indirizzo IP della rete in modo da leggere la porta sorgente sniffando i pacchetti che vengono scambiati sulla rete.

Nel passo successivo supponiamo di voler reindirizzare le richieste legittime per il sito www.bankofsteve.com ad un sito fake controllato dagli attaccanti.



La vittima effettua una query DNS per il dominio partendo dai roots o dai GTLD servers. Simultaneamente a queste query, badguy effettua delle query con QUERY ID appena superiore a quello visto durante lo sniffing del traffico lecito, in cui risponde che l'indirizzo IP del dominio richiesto è invece un IP da lui controllato. A condizione che il pacchetto inviato dall'attaccante arrivi prima a destinazione rispetto alla risposta lecita, questa informazione viene presa come valida dal DNS locale.

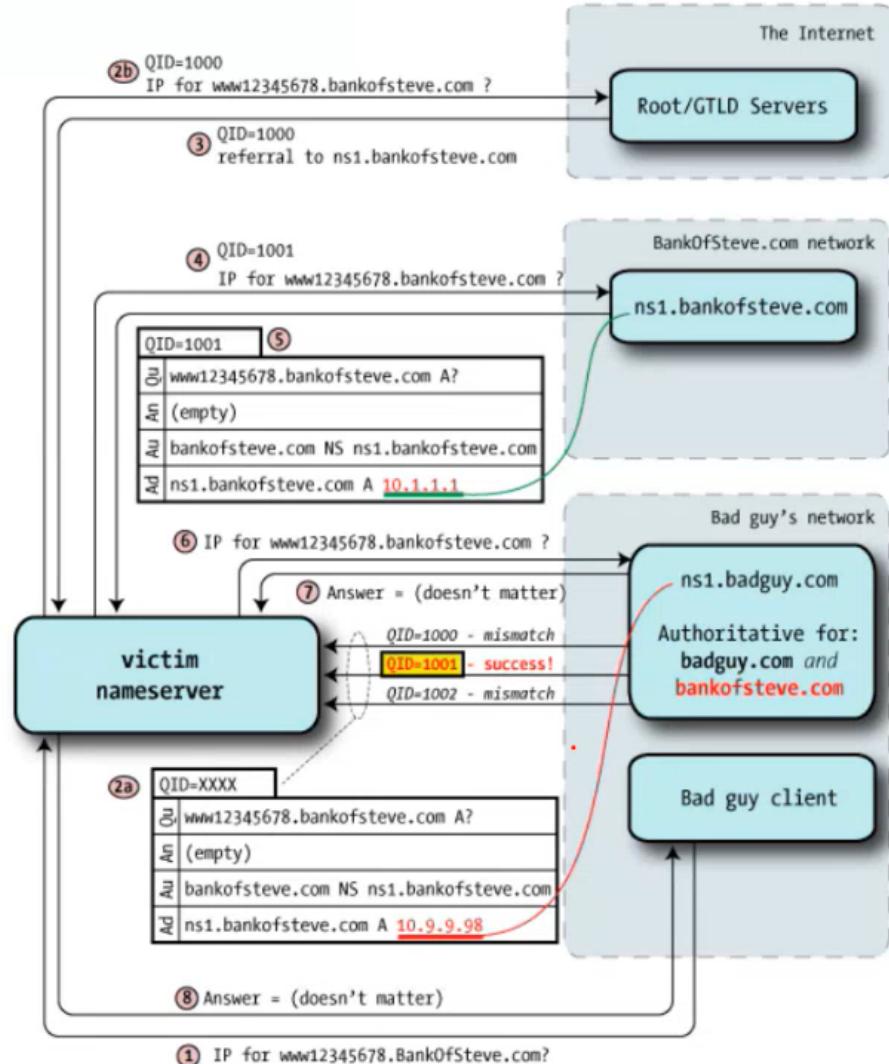
• Note that the attack only works if:

- name isn't already in the cache
- bad guy guesses the query ID
- bad guy is faster than the real nameserver

Le contromisure da adottare consistono nell'usare QUERY ID random ma anche qui bisogna usare un generatore randomico sicuro e davvero randomico per evitare che vengano predetti i QUERY ID futuri.

L'attacco così mostrato al giorno d'oggi è difficile da replicare in quanto in un sito web esistono riferimenti ad altri sitiweb di cui dunque le entry DNS dovrebbero essere iniettate all'interno della cache della vittima.

DNS cache poisoning - versione di Kaminsky



Anzichè avvelenare la cache relativa ai singoli siti web, molto meglio se riesco a inserire un server DNS fake all'interno del domini così che tutte le richieste DNS degli utenti verranno risolte dal DNS fake controllato dall'attaccante.

L'attacco è simile a quello precedente, dove però l'attaccante forza il DNS server locale a risolvere un sito web del dominio target sicuramente non esistente, dopo di chè analizzando il possibile QUERY ID da usare, andrà ad inviare una risposta per la richiesta effettuata inserendo anche un record per il nameserver del dominio targer e il corrispondente indirizzo IP che però sarà differente rispetto a quello reale, in quanto sarà l'IP controllato dall'attaccante.

08/07/2023

L'idle scan è una tecnica con cui viene trovata nella rete una macchina che mi permetta di fare delle scansioni in modo tale da non essere facilmente rilevabile da un firewall o IDS ma nel contempo di ottenere i risultati:

```

→ ~ nmap
Nmap 7.94 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan →
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports sequentially - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)

```

Usiamo un IP della rete (192.168.104.104) per scansionare la rete 192.168.104.253:

```

→ ~ nmap -sI 192.168.104.104 192.168.104.253
WARNING: Many people use -Pn w/Idlescan to prevent pings from their true IP. On the other hand, timing info Nmap gains from pings can allow for faster, more reliable
scans.
You requested a scan type which requires root privileges.
QUITTING!

```

Il primo warning ci dice che se ad nmap non specifico l'opzione -Pn, prima di fare la scansione delle porte nmap effettuerà un ping scan, che nel caso in cui non vogliamo apparire agli occhi della vittima sarebbe da evitare.

Il ping deve essere effettuato per forza con l'IP dell'attaccante dunque evitiamo il ping scan iniziale:

```

Nmap scan report for 192.168.104.253
Host is up (0.17s latency).
Nmap done: 256 IP addresses (47 hosts up) scanned in 35.74 seconds
→ ~ nmap -sI 192.168.104.104 192.168.104.253
WARNING: Many people use -Pn w/Idlescan to prevent pings from their true IP. On the other hand, timing info Nmap gains from pings can allow for faster, more reliable
scans.
You requested a scan type which requires root privileges.
QUITTING!
→ ~ sudo nmap -Pn -sI 192.168.104.104 192.168.104.253 → VITTIMA
Password: → ZOMBIE
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-08 11:21 CEST
Nmap done: 1 IP address (0 hosts up) scanned in 1.47 seconds
→ ~

```

```

+ ~ sudo nmap -Pn 192.168.104.244
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-08 11:22 CEST
Nmap scan report for 192.168.104.244
Host is up (0.024s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
443/tcp    open  https
MAC Address: 98:22:6E:F2:A5:A8 (Amazon Technologies)

Nmap done: 1 IP address (1 host up) scanned in 0.69 seconds
+ ~ sudo nmap -sI 192.168.104.104 192.168.104.244
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-08 11:22 CEST
Idle scan zombie 192.168.104.104 (192.168.104.104) port 80 cannot be used because IP ID sequence class is: All zeros. Try another proxy.
QUITTING!
+ ~ sudo nmap -v 192.168.104.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-08 11:23 CEST
Initiating ARP Ping Scan at 11:23
Scanning 255 hosts [1 port/host]

```

In verde uno scan delle porte classiche mentre in giallo uno scan effettuato usando uno zombie della rete.

Il suggerimento dato da nmap ci dice che viene usato un campo che è l'ID dell'header IP.

Nell'header IP è presente una riga destinata alla gestione della frammentazione con i flag, l'offset e poi c'è un ID usato per riassemblare i frammenti andando a capire che i frammenti fanno parte dello stesso pacchetto originale.

Quel campo ID è particolare in quanto al di là dell'uso classico, lo standard ci dà delle libertà su come ciascun sistema operativo può incrementarlo, usarlo etc...



Idle scan

- The idle scan is a TCP port scan method that through utility software tools such as Nmap and Hping allows sending spoofed packets to a computer.
- First of all it is necessary to identify a zombie (by means of a ping sweep)
- The zombie must be inactive in the Internet**
- nmap -sI <zombie IP> <victim IP>**

Poniamoci nel caso di far fare allo zombie la scansione di una porta aperta. Prima cosa da fare è individuare uno zombie, ovvero una macchina inattiva in rete che non produce traffico e implementa un OS datato oppure estremamente limitato (un device IoT).

Bisogna a questo punto mandare un pacchetto allo zombie per capire quale valore utilizza nel campo ID dell'header IP. Sfruttiamo TCP mandando un pacchetto di SYN/ACK allo zombie così da poter leggere il campo ID senza però instaurare una sessione con lo zombie stesso. Inoltre mi sincero che lo zombie mi risponda con un pacchetto di RST il chè non è scontato in quanto se ci fosse in mezzo alla comunicazione un firewall, lo zombie poteva anche non rispondere al pacchetto di SYN/ACK.



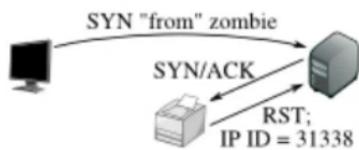
Idle Scan - Open port

Step 1: Probe the zombie's IP ID.



The attacker sends a SYN/ACK to the zombie. The zombie, not expecting the SYN/ACK, sends back a RST, disclosing its IP ID.

Step 2: Forge a SYN packet from the zombie.



The target sends a SYN/ACK in response to the SYN that appears to come from the zombie. The zombie, not expecting it, sends back a RST, incrementing its IP ID in the process.

Step 3: Probe the zombie's IP ID again.



The zombie's IP ID has increased by 2 since step 1, so the port is open!

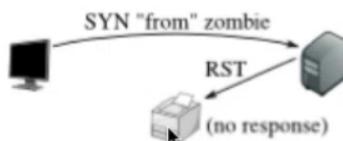
A questo punto viene mandato un pacchetto di SYN per il port scan usando come indirizzo IP sorgente non quello della macchina attaccante ma dello zombie (spoofing). Se la porta è aperta, il target genererà un pacchetto di SYN/ACK che verrà inviato allo zombie. Lo zombie risponderà con un pacchetto di RST alla vittima. Se nel frattempo lo zombie è stato silente e non ha generato altro traffico, l'ID del pacchetto sarà esattamente l'ID del pacchetto precedente + 1. A questo punto si attende un po' di tempo e si ripete lo step 1 andando a leggere l'ID corrente dello zombie. Se trovo che il valore è pari a quello iniziale + 2 questo mi dice che nel mezzo lo zombie ha mandato un pacchetto al target e **se la porta fosse stata chiusa:**

Step 1: Probe the zombie's IP ID.



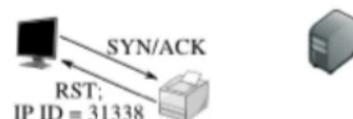
The attacker sends a SYN/ACK to the zombie. The zombie, not expecting the SYN/ACK, sends back a RST, disclosing its IP ID. This step is always the same.

Step 2: Forge a SYN packet from the zombie.



The target sends a RST (the port is closed) in response to the SYN that appears to come from the zombie. The zombie ignores the unsolicited RST, leaving its IP ID unchanged.

Step 3: Probe the zombie's IP ID again.



The zombie's IP ID has increased by only 1 since step 1, so the port is not open.

il target avrebbe mandato un RST allo zombie e lo zombie non avrebbe risposto lasciando l'ID intatto pari a quello originale + 1, ovvero incrementato di un solo valore.

Le 2 condizioni sono:

- lo zombie deve essere inattivo sulla rete, altrimenti non troverò mai l'ID incrementato come mi aspetto e diventa difficile tracciare il percorso dei pacchetti

- che il campo ID dell'header IP venga incrementato nel modo in cui ce lo aspettiamo. L'RFC da piena libertà sull'utilizzo di questo campo. Basterebbe gestire in modo random questo ID per far crollare questa condizione oppure disabilitare l'utilizzo di questo ID non incrementandolo dato che la frammentazione al giorno d'oggi viene usata raramente.

Facendo uno scan sulla rete per capire quali host sono sfruttabili come zombie tramite il comando:

```
nmap -Pn --open -p T:80,443 -o 192.168.0.0/24
```

```
→ ~ sudo nmap -Pn -vv --open -p T:80,443 -o 192.168.104.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-08 11:30 CEST
Initiating ARP Ping Scan at 11:30
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 11:30, 27.50s elapsed (255 total hosts)
Initiating Parallel DNS resolution of 54 hosts. at 11:30
Completed Parallel DNS resolution of 54 hosts. at 11:30, 0.09s elapsed
Initiating Parallel DNS resolution of 1 host. at 11:30
Completed Parallel DNS resolution of 1 host. at 11:30, 0.01s elapsed
Initiating SYN Stealth Scan at 11:30
Scanning 54 hosts [2 ports/host]
Discovered open port 80/tcp on 192.168.104.143
Discovered open port 80/tcp on 192.168.104.188
Discovered open port 80/tcp on 192.168.104.175
Discovered open port 80/tcp on 192.168.104.1
Discovered open port 80/tcp on 192.168.104.239
Discovered open port 80/tcp on 192.168.104.242
Discovered open port 443/tcp on 192.168.104.143
Discovered open port 443/tcp on 192.168.104.1
Discovered open port 443/tcp on 192.168.104.244
Completed SYN Stealth Scan at 11:31, 20.36s elapsed (108 total ports)
Initiating OS detection (try #1) against 54 hosts
adjust_timeouts2: packet supposedly had rtt of -216475 microseconds. Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -216475 microseconds. Ignoring time.
Completed os scan against 192.168.104.52 in 37.201s (53 hosts left)
TCP/IP fingerprint:
SCAN(V=7.94E-4%D=7/8%O=80%CT=%CU=36562%PV=Y%DS=1%DC=D%G=N%W=20B001%TM=64A92D10%P=arm-apple-darwin22.4.0)
SEQ(SP=106%CD=1%SR=10B%TI=ZKTS=A)
SEQ(SP=106%CD=1%SR=10B%TI=ZKII=IXTS=A)
OPSC01=MSB4ST11NW6%02=MSB4ST11NW6%03=M5B4NNT11NW6%04=M5B4ST11NW6%05=M5B4ST11NW6%06=M5B4ST11
WINC(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)
ECNC(R=Y%DF=Y%T=40%W=7210%O=MSB4NNSNW6%CC=N%Q=)
T1(R=Y%DF=Y%T=40%S=0%A=S+F=AS%RD=0%Q=)
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%T=40%W=0%S=A%KA=Z%F=R%O=%RD=0%Q=)
T5(R=Y%DF=Y%T=40%W=0%S=Z%KA=S+F=ARX%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=40%W=0%S=A%KA=Z%F=R%O=%RD=0%Q=)
U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(R=Y%DFI=N%T=40%CD=S)

Uptime guess: 8.008 days (since Fri Jun 30 11:20:37 2023)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: All zeros 
Nmap scan report for 192.168.104.143
Host is up, received arp-response (0.011s latency).
Scanned at 2023-07-08 11:30:46 CEST for 64s

PORT      STATE SERVICE REASON
80/tcp    open  http   syn-ack ttl 64
443/tcp   open  https  syn-ack ttl 64
MAC Address: C2:39:6F:94:A1:37 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
Aggressive OS guesses: Android 4.0 (97%), Linux 3.10 - 4.11 (97%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (97%), DD-WRT v0.0 (Linux 4.4.2) (97%), Android 4.1.2 (96%), Linux 2.6.32 (96%), Linux 3.2 - 3.10 (96%), Linux 3.2 - 3.16 (96%), Linux 2.6.32 - 3.10 (96%), Linux 3.2 - 4.9 (96%) No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SCAN(V=7.94E-4%D=7/8%O=80%CT=%CU=31778%PV=Y%DS=1%DC=D%G=N%W=C2396F%TM=64A92D10%P=arm-apple-darwin22.4.0)
SEQ(SP=103%CD=1%SR=10%TI=ZKII=IXTS=8)
SEQ(SP=103%CD=1%SR=10%TI=ZKCI=IXII=IXTS=8)
OPSC01=MSB4ST11NW3%02=MSB4ST11NW3%03=M5B4NNT11NW3%04=M5B4ST11NW3%05=M5B4ST11NW3%06=M5B4ST11
WINC(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)
ECNC(R=Y%DF=Y%T=40%W=7210%O=MSB4NNSNW3%CC=Y%Q=)
T1(R=Y%DF=Y%T=40%S=0%A=S+F=AS%RD=0%Q=)
```

in giallo vediamo che questo host ad esempio non incrementa l'ID Sequence dunque non è sfruttabile.

Mentre questo IP dato che riporta IP ID Sequence Generation: Incremental significa che è sfruttabile:

```
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: All zeros

Nmap scan report for 192.168.104.239
Host is up, received arp-response (0.015s latency).
Scanned at 2023-07-08 11:30:46 CEST for 74s
Not shown: 1 closed tcp port (reset)
PORT      STATE SERVICE REASON
80/tcp    open  http   syn-ack ttl 255
MAC Address: E8:9F:6D:52:4E:78 (Espressif)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

TCP/IP fingerprint:

```
OS:SCAN(V=7.94%E=4%D=7/8%OT=80%CT=443%CU=32965%PV=Y%DS=1%DC=D%G=Y%M=E89F6D%
OS:TM=64A92D10%P=arm-apple-darwin22.4.0)SEQ(SP=0%GCD=2DFB0%ISR=A6%TI=I%CI=I
OS:%II=RIKSS=S%TS=U)SEQ(SP=0%GCD=2DFBD%ISR=A6%TI=I%CI=I%II=RIKSS=0%TS=U)SEQ
OS:(SP=78%GCD=1%ISR=A6%TI=I%CI=I%II=RIKSS=S%TS=U)SEQ(SP=7B%GCD=1%ISR=A6%TI=
OS:I%CI=I%TS=U)SEQ(SP=B4%GCD=1%ISR=BCXTI=RDXC1=I%II=RIKTS=U)OPS(O1=M59C%02=
OS:M59C%03=M59C%04=M59C%05=M59C%06=M59C)WIN(W1=1670%W2=1670%W3=1670%W4=1670
OS:%W5=1670%W6=1670)ECN(R=Y%DF=N%T=FF%W=1670%O=M59C%CC=N%Q=)T1(R=Y%DF=N%T=F
OS:F%S=0%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=N%T=FF%W=1670%S=0%A=S+%F=AS%O=M
OS:59C%RD=0%Q=)T4(R=Y%DF=N%T=FF%W=1670%S=A%A=S+%F=AR%O=%RD=0%Q=)T5(R=Y%DF=N%
OS:T=FF%W=1670%S=A%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=N%T=FF%W=1670%S=A%A=S+%F=A
OS:R%O=%RD=0%Q=)T7(R=Y%DF=N%T=FF%W=1670%S=A%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=
OS:N%T=FF%IPL=38%UN=0%RIPL=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=S%T=FF%C
OS:D=S)
```

Network Distance: 1 hop

TCP Sequence Prediction: Difficulty=120 (Good luck!)

IP ID Sequence Generation: Incremental

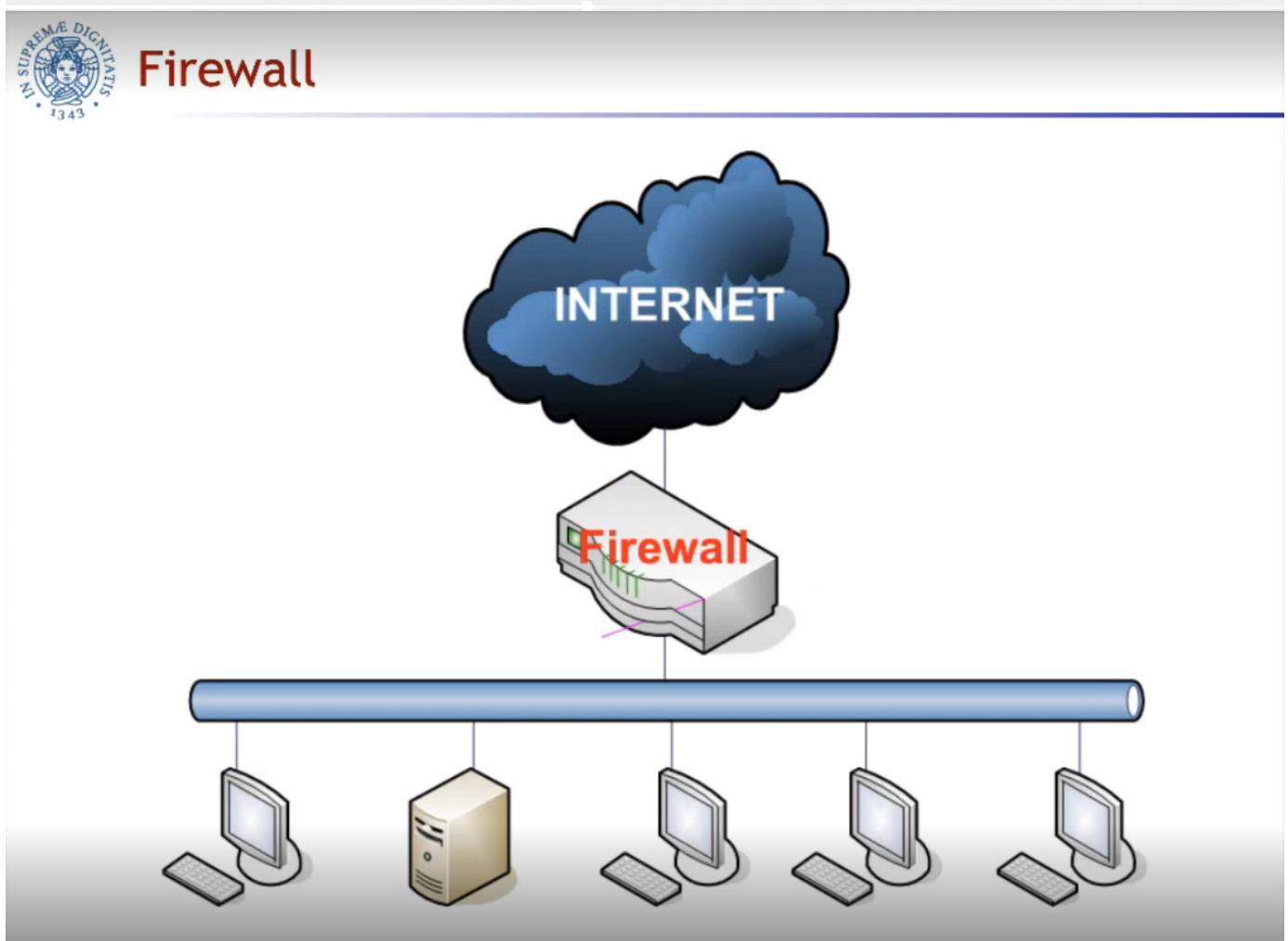
```
Nmap scan report for 192.168.104.242
Host is up, received arp-response (0.015s latency).
Scanned at 2023-07-08 11:30:46 CEST for 74s
Not shown: 1 closed tcp port (reset)
PORT      STATE SERVICE REASON
80/tcp    open  http   syn-ack ttl 255
MAC Address: 94:3C:C6:4C:8F:2C (Espressif)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

Firewall

Firewall

**Dr. Christian Callegari, PhD
Dept. of Information Engineering
University of Pisa - ITALY**

**Master in CyberSecurity
Pisa**



Si tratta di una macchina che lavora nella rete e interposta tra la rete interna e la rete Internet.
Il firewall deve essere un dispositivo sicuro e posto per vedere tutte le connessioni di rete.



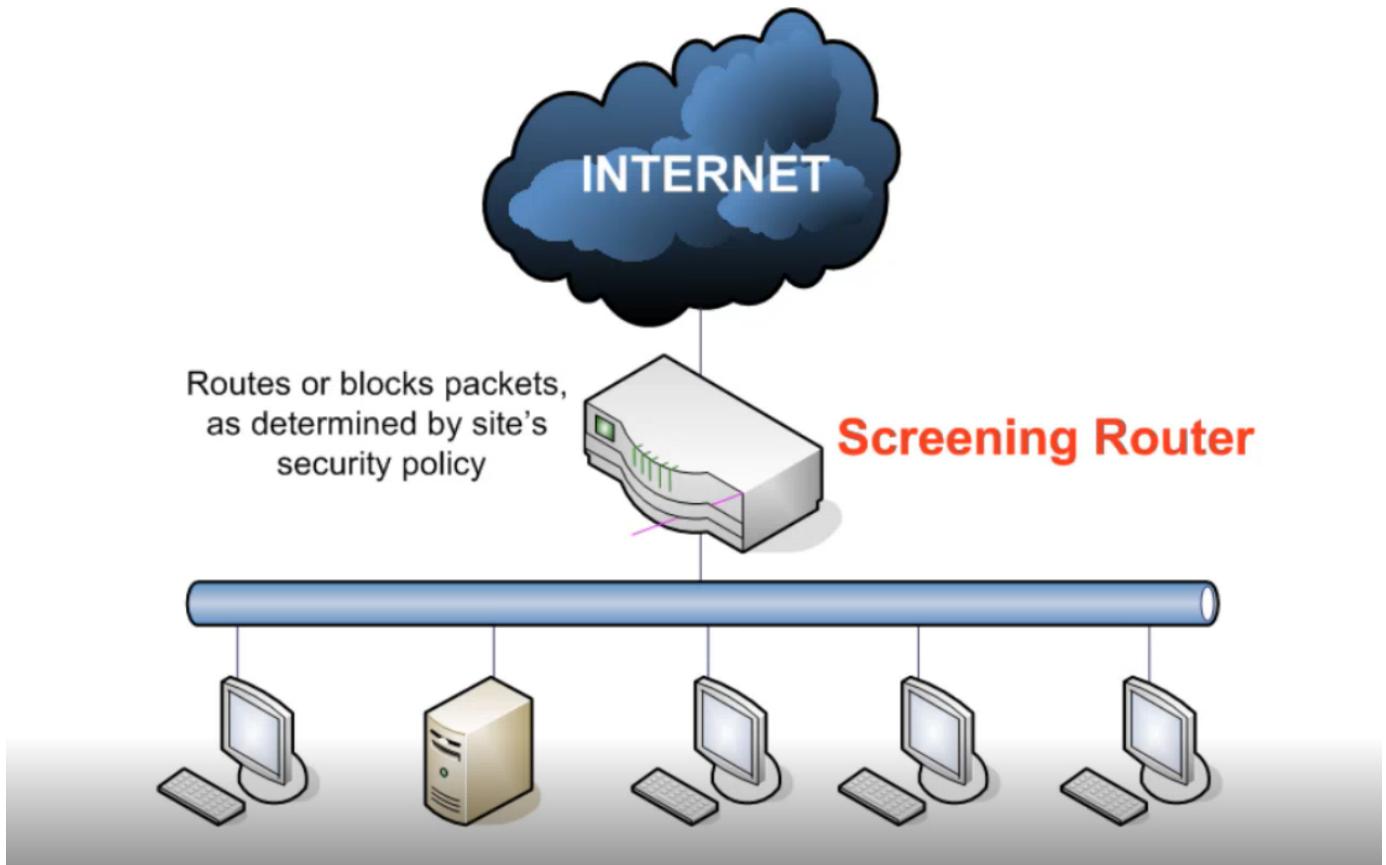
What is a Firewall?

- A **choke point** of control and monitoring
- Interconnects networks with differing trust
- Imposes restrictions on network services
 - only authorized traffic is allowed
- Auditing and controlling access
 - can implement alarms for abnormal behavior
- **Itself immune to penetration**
- Provides perimeter defence
- **Fail Safe Strategy!**
- Characterized by protocol level it controls in
 - Packet filtering
 - Circuit gateways
 - Application gateways

»

Il firewall può essere compromesso dunque ci vuole una strategia Fail Safe!

Packet Filtering



A packet filtering FW selectively forwards packets from/to LAN

- Mainly uses transport-layer information
 - IP Source Address, Destination Address
 - Protocol/Next Header (TCP, UDP, ICMP, etc)
 - TCP or UDP source & destination ports
 - TCP Flags (SYN, ACK, FIN, RST, PSH, etc)
 - ICMP message type
- Examples
 - DNS uses port 53
 - No incoming port 53 packets except known trusted servers
- In some cases the FW also checks the payload content



Packet filtering

- Can Filter with incoming or outgoing interfaces
 - E.g., Ingress filtering of spoofed IP addresses
- Permits or denies certain services
 - knowledge of TCP and UDP port utilization on a number of O.S.
- Once analysed the packet, the FW can:
 - Forward the packet
 - Drop the packet
 - Drop packet and notify the source
 - ICMP “destination unreachable”
 - ICMP “destination administratively unreachable”
 - TCP reset
 - Create an event in the log file
 - Send an alarm
- The FW can optionally
 - Modify the packet (e.g., NAT)
 - Send the packet to a different destination

Ci sono diverse strategie per configurare un firewall ma di base il Default Deny è la strategia Fail Safe in quanto sbagliando a configurare una regola comunque c'è una regola di fallback in cui piuttosto che

far passare il traffico, lo nega:



Configuration

- Start with a security policy (**Default Deny** or Default Permit)
- Specify allowable packets in terms of logical expressions on packet fields
- General rules - least privilege
 - All that is not expressly permitted is prohibited
 - If you do not need it, eliminate it
- Rules order is important!

deny 192.168.0.10
accept 192.168.0.0/24

is different from

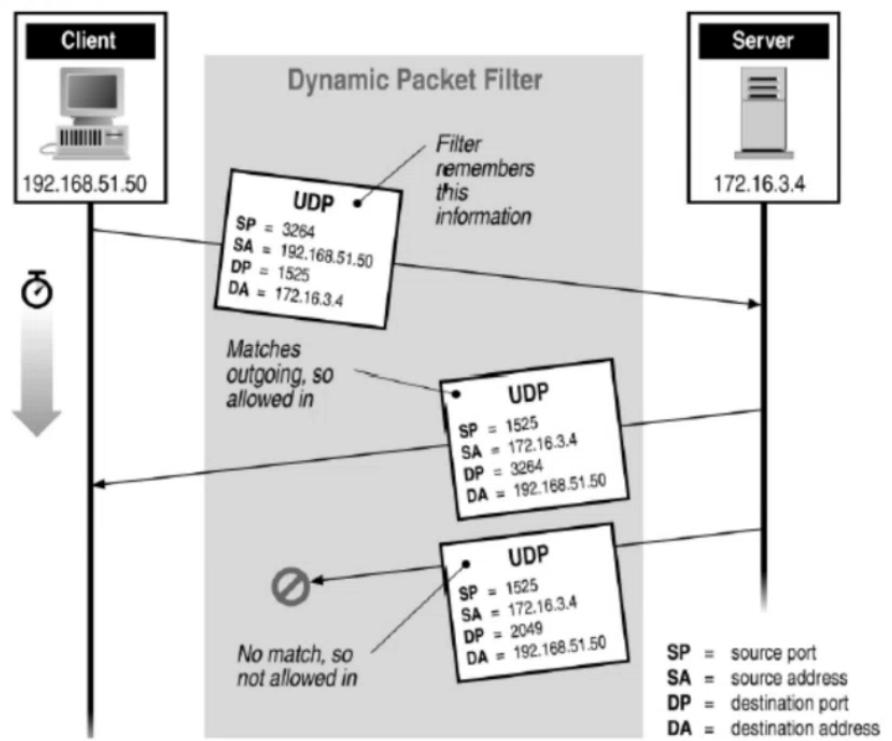
accept 192.168.0.0/24
deny 192.168.0.10

Un firewall stateful (o dinamico) ci consente di evitare di dover scrivere un quantitativo di regole innumerevole in quanto se una regola consente un traffico verso l'esterno sarà automaticamente consentito anche tutto il traffico di ritorno:

Stateful (Dynamic) Packet filtering

A FW can filter packets on the basis of previous “behavior”

- First packet of that type?
- Response packet?



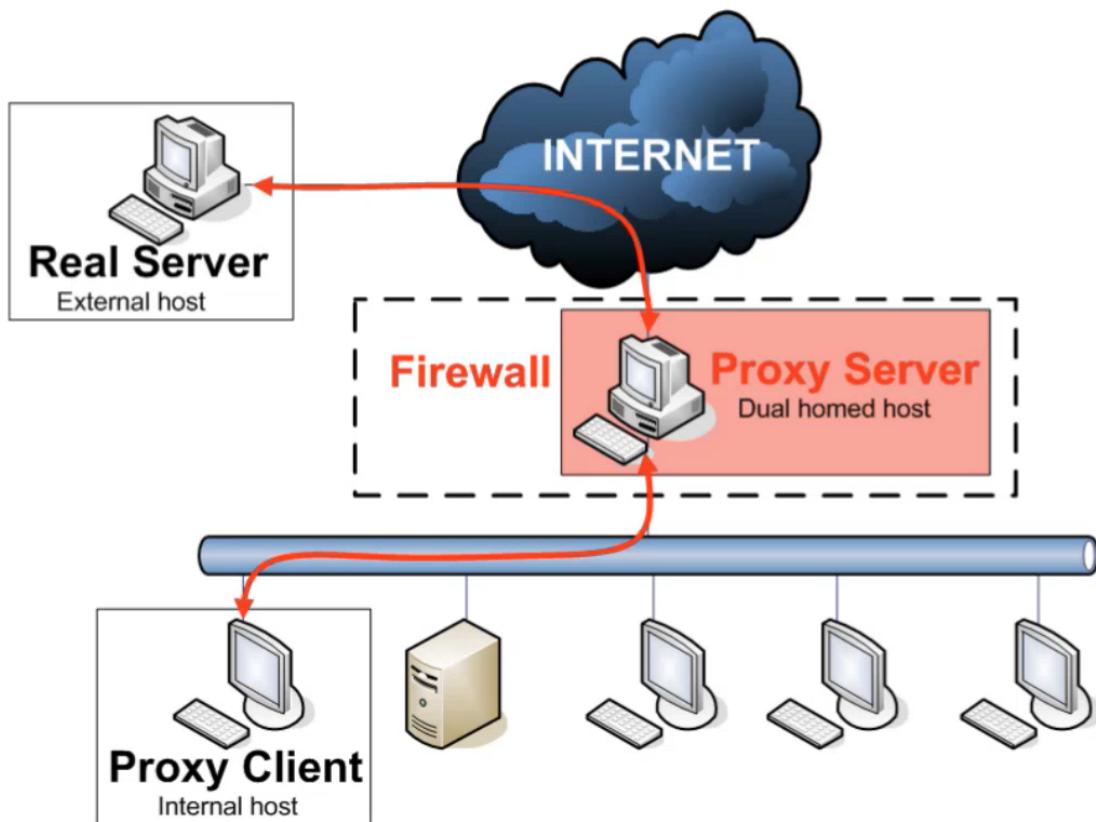
questo presenta un problema di memoria in quanto mandando un numero esagerato di pacchetti al firewall potrebbe bloccarsi e non consentire più alcuna connessione.

Questo tipo di firewall packet filtering non gestisce regole specifiche per le utenze, ovvero se siamo in una situazione con DHCP non possiamo sapere quale utente ha ricevuto quale IP dunque non può applicare delle regole specifiche per ciascuna utenza. Per farlo è necessario un

Proxy



Proxy



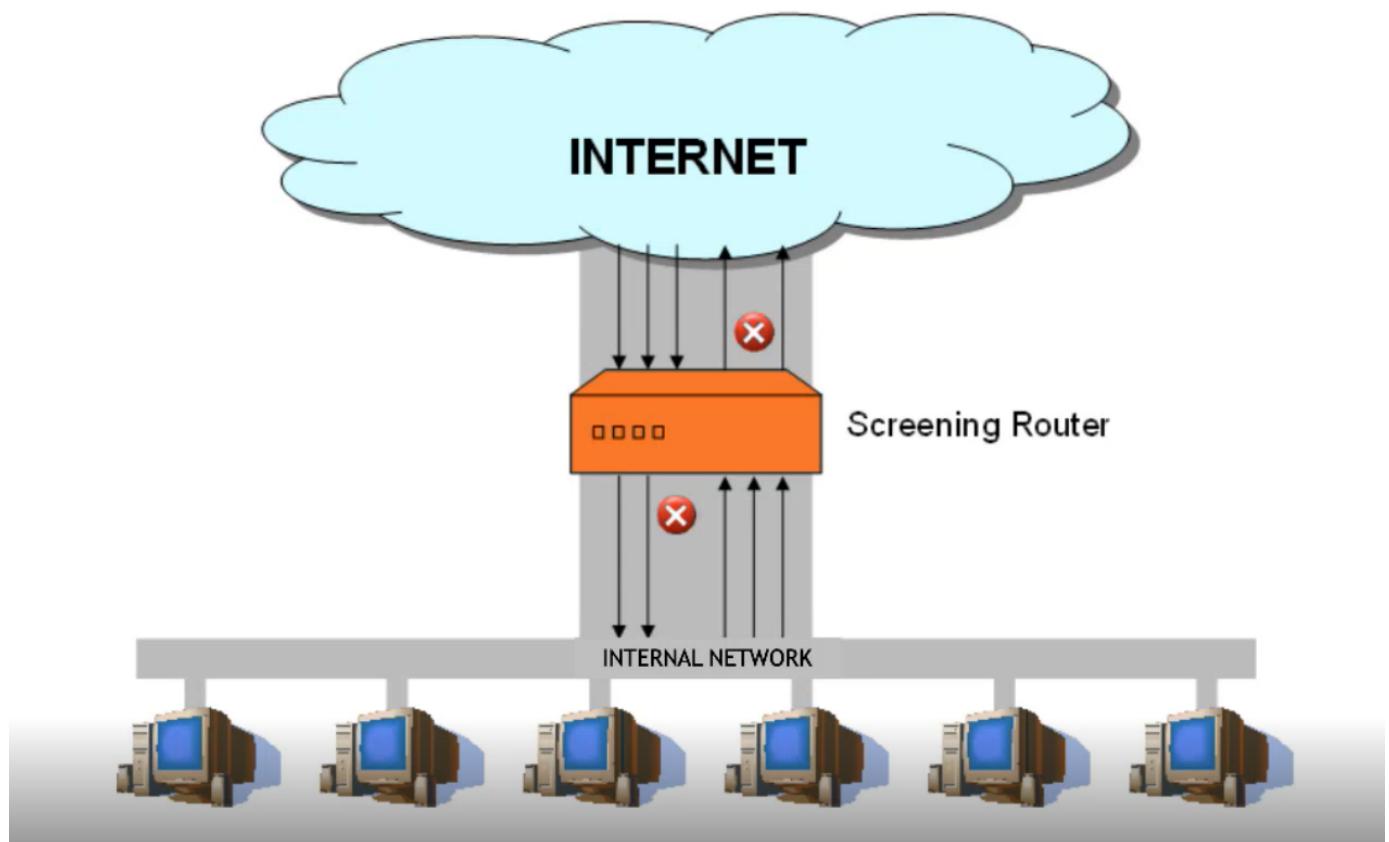
Christian Callegari - Firewall

12

Ci riferiamo dunque quando parliamo di firewall di un tipo di appliance fisica interposta tra la rete interna ed esterna:



Firewall - Architectures

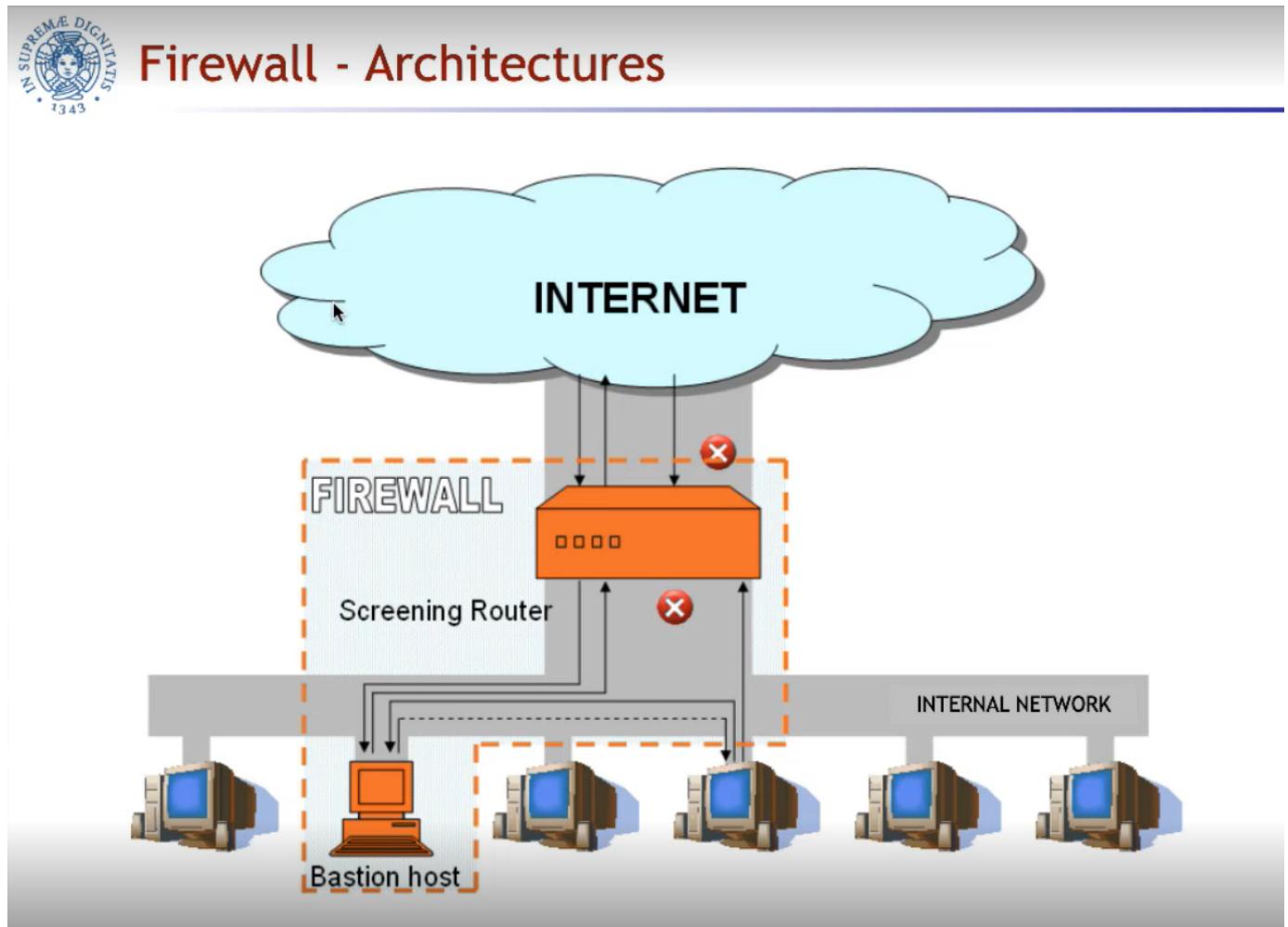


Il limite grosso di un firewall in questa posizione è che non viene fatta una analisi del traffico interno dunque un insider threat non viene rilevato.

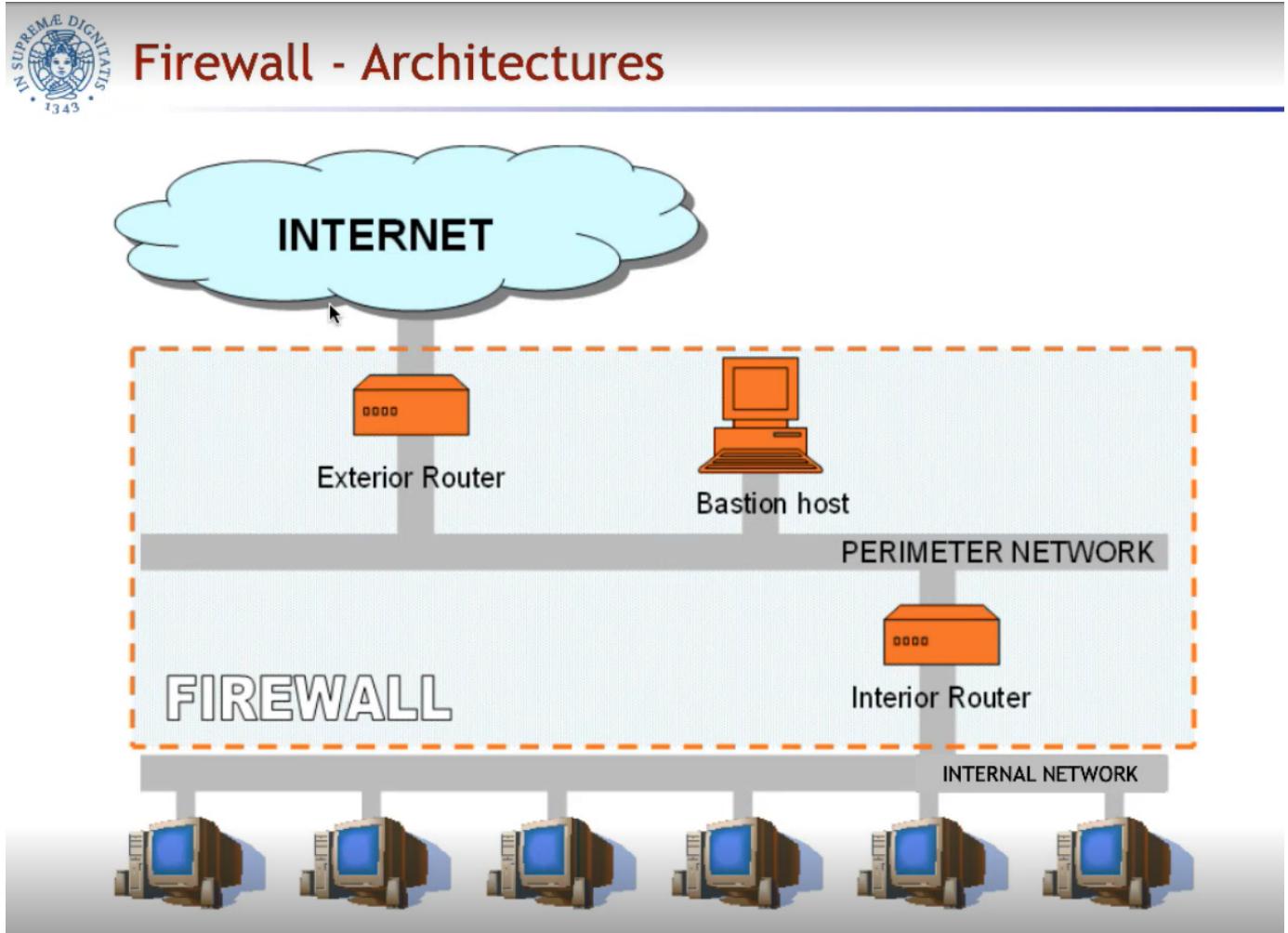
Il firewall visto prima ha come regola di default che le connessioni entranti sono vietate tutte, dunque consento agli utenti di iniziare una connessione verso l'esterno ma non permetto connessioni iniziate dall'esterno.

Potrei però avere un server web aziendale (bastion host), ovvero un host della rete da proteggere in

quanto è esposto su internet ma è posizionato all'interno della rete:



Una soluzione alternativa è quella di interporre il bastion host tra due firewall:



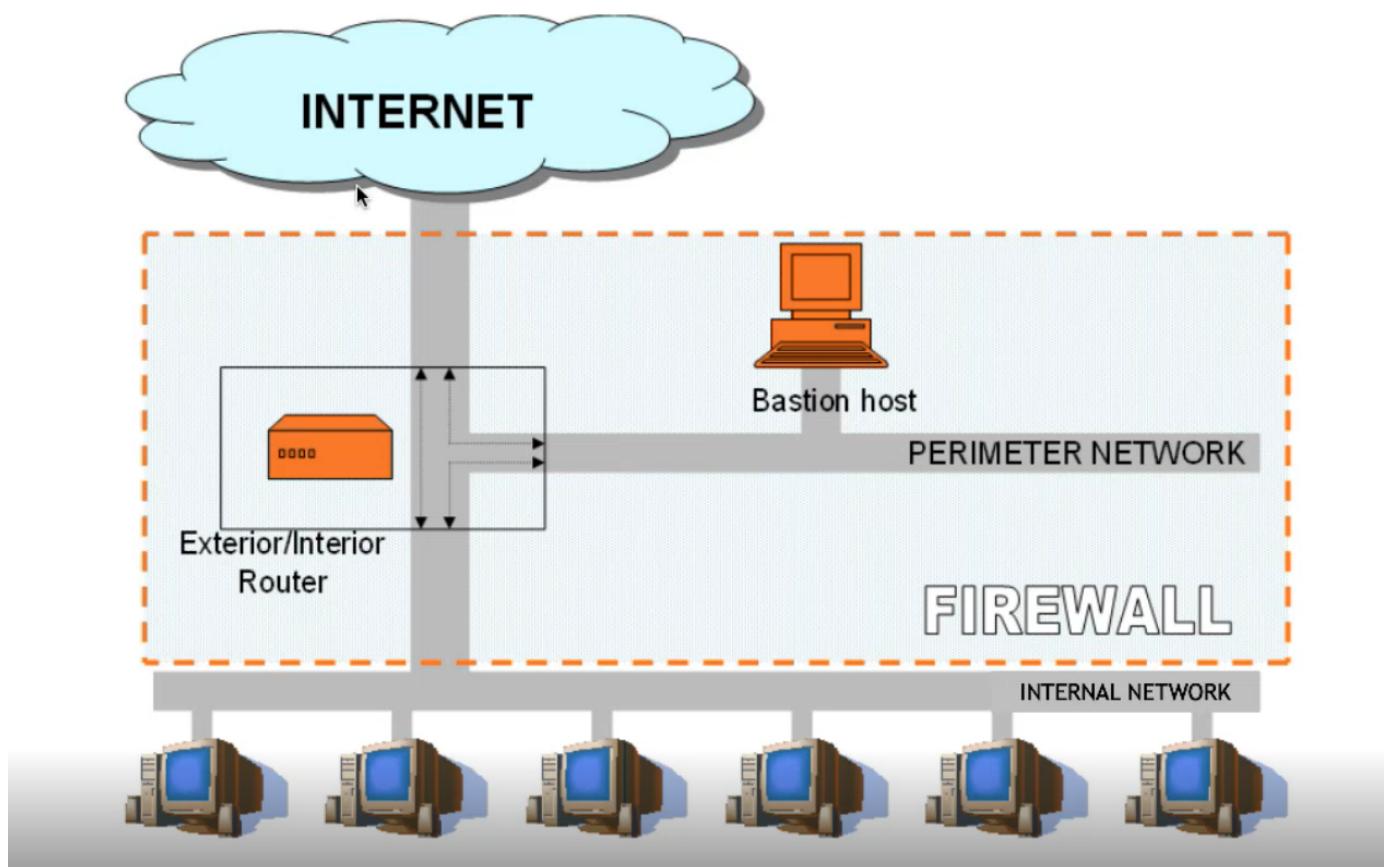
La perimeter network è una DMZ.

Questa soluzione è ottima ma è costosa in quanto il firewall costa, dunque nella maggior parte dei casi

si usano semplicemente porte fisiche del firewall diverse ma usando un unico dispositivo di firewalling:



Firewall - Architectures



IPTables

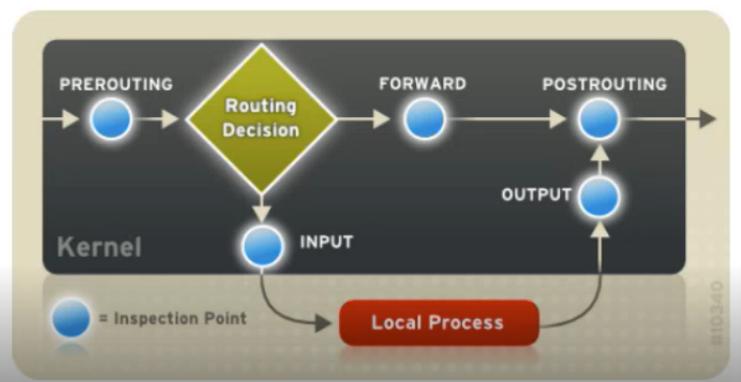


- Linux Firewall kernel module

- From kernel 2.4

- Three main concepts:

- Tables
 - FILTER
 - NAT
 - MANGLE
- Chain
 - PREROUTING
 - INPUT
 - FORWARD
 - OUTPUT
 - POSTROUTING
- Rules



E' un modulo del kernel che deve essere abilitato ed è costituito da tabelle e chain:

- chain: inspection mode ovvero i punti in cui posso andare ad effettuare le varie operazioni
- tabelle: le regole da applicare

Il firewall anche se è un filtro non si limita solo a forwardare il traffico o dropparlo ma deve anche gestire del traffico che è destinato a se stesso (ad esempio il traffico dell'interfaccia di gestione).

Un pacchetto che arriva passa per il:

1. **PREROUTING**: serve per capire a chi è destinato il pacchetto. Qui viene effettuato il NAT del traffico. Il traffico che arriva dall'esterno verso l'interno della mia rete è indirizzato all'indirizzo IP pubblico del firewall e non a un IP specifico della rete interna, dunque la prima cosa da fare è fare il DENAT ovvero associare l'IP privato a cui è destinato il traffico.
2. **ROUTING DECISION**: è destinato al firewall stesso? Allora va mandato all'input dove verifico se è lecito oppure no dunque faccio filtraggio. Se è destinato a una rete interna, vado al blocco di forward.
3. **FORWARD**: in questo blocco scrivo le regole di filtraggio, vedo se è consentito oppure no
4. **POSTROUTING**: serve per il traffico outbound. Infatti se ricevo del traffico che proviene dall'interno verso l'esterno bypasso il PREROUTING, supponendo che non sia destinato al firewall bypasso anche l'INPUT e passo al FORWARD in cui verifico se quello specifico IP sorgente privato può mandare il traffico oppure no, a quel punto giro il pacchetto al blocco di POSTROUTING che si occupa di fare NAT e inviarlo in rete.

Le regole in IPTables sono molto facili ovvero sono quasi human readable e specificate in questo modo:



IPTables - Rule Targets & Syntax

- ACCEPT - let the packet through
- REJECT - sends ICMP error message
- DROP - reject, but don't send ICMP message
- MASQ - masquerade
- RETURN - end of chain; stop traversing this chain and resume the calling chain
- QUEUE - pass the packet to the user space
- (none) - rule's counters incremented and packet passed on (used for accounting)

`iptables -t TABLE -A CHAIN -[ilo] IFACE -s w.x.y.z -d a.b.c.d -p PROT -m state --state STATE -j ACTION`

- TABLE = nat | filter | mangle
- CHAIN = INPUT | OUTPUT | FORWARD | PREROUTING | POSTROUTING
- IFACE = eth0 | eth1 | ppp0 | ...
- PROT = tcp | icmp | udp | ...
- STATE = NEW | ESTABLISHED | RELATED | ...
- ACTION = DROP | ACCEPT | REJECT | DNAT | SNAT | ...

con `-t` viene specificata la tabella, noi ci concentreremo solo sul filter.

Noi faremo delle regole per la nostra macchina specifica dunque saranno o in input o in output, pertanto nella chain specificata con `-A` avremo INPUT o OUTPUT.

`--state` si riferisce al filtraggio dinamico ovvero se voglio dire ad esempio che il traffico di risposta è consentito devo specificare come state l'ESTABLISHED e RELATED



IPTables - Examples

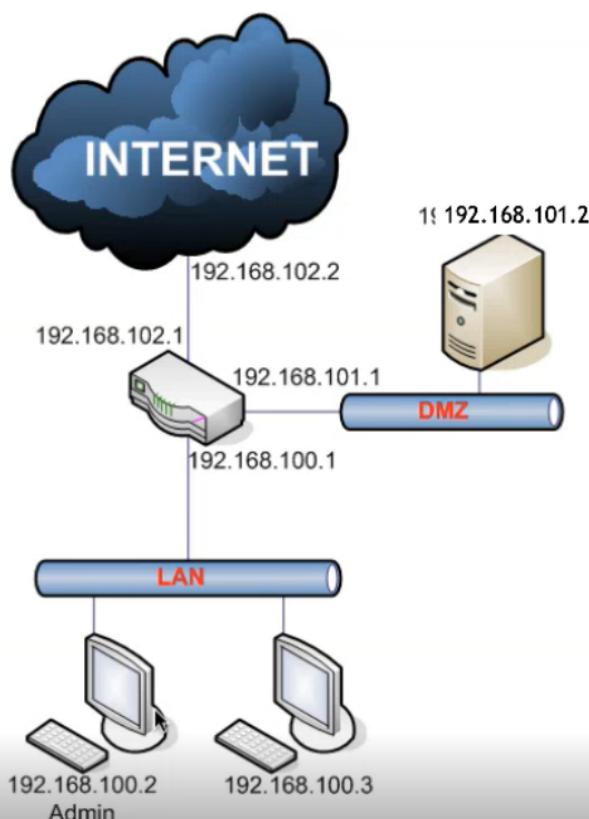
1. `iptables --flush`
» Delete all rules
2. `iptables -A INPUT -i lo -j ACCEPT`
» Accept all packets arriving on lo for local processes
3. `iptables -A OUTPUT -o lo -j ACCEPT`
4. `iptables --policy INPUT DROP`
» Unless other rules apply, drop all INPUT packets
5. `iptables --policy OUTPUT DROP`
6. `iptables --policy FORWARD DROP`
7. `iptables -L -v -n`
» List all rules, verbosely, using numeric IP addresses etc.
8. `iptables -A INPUT -i eth1 -p tcp -s 192.168.17.1 --sport 1024:65535 -d 192.168.17.2 --dport 22 -j ACCEPT`
 - Accept all TCP packets arriving on eth1 for local processes from 192.168.17.1 with any source port higher than 1023 to 192.168.17.2 and destination port 22
9. `iptables -t nat -A PREROUTING -p TCP -i eth0 -d 128.168.60.12 --dport 80 -j DNAT --to-destination 192.168.10.2`
 - Change the destination address of all TCP packets arriving on eth0 aimed at 128.168.60.12 port 80 to 192.168.10.2 port 80.

1. `iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 0:1023 -j REJECT`
 - Reject all incoming TCP traffic destined for ports 0 to 1023

Supponiamo di avere una rete come in figura:



Securing a single network



- **FIREWALL**
 - Linux (Iptables)
 - Commercial firewalls (e.g., checkpoint)
- **INTRUSION DETECTION SYSTEM**
 - Snort
 - Bro
 - Commercial IDSs
- Internal traffic protection
 - IPSec



- **Default Deny**
- **From LAN to FW**
 - SSH - only from Administrator
- **From LAN to DMZ**
 - SSH, Web, E-Mail, DNS, ICMP
- **From LAN to Ext**
 - SSH, Web, E-Mail, DNS, ICMP
- **From DMZ to LAN**
 - Only established connections
(related?)
- **From DMZ to FW**
 - Ping (?)
- **From DMZ to Ext**
 - SSH, Web, E-Mail, DNS
- **From Ext to FW**
 - Nothing
- **From Ext to LAN**
 - Only established connections
(related?)
- **From Ext to DMZ**
 - Web (Port Forwarding)

Le regole IPTables si cancellano ad ogni reboot, dunque in genere si crea un file in modo tale che al boot le regole vengano applicate grazie alla configurazione di un servizio di sistema che applica le regole presenti all'interno del file.

Definiamo alcune macro o variabili globali per rendere maggiormente leggibili le regole:



Iptables

```
#!/bin/sh
# Define interfaces
INTIF="eth0"
DMZIF="eth1"
EXTIF="eth2"
# Define interfaces addresses
INTIP="192.168.100.1"
DMZIP="192.168.101.1"
EXTIP="192.168.102.1"
# Define networks
INTNET="192.168.100.0/255.255.255.0"
DMZNET="192.168.101.0/255.255.255.0"
EXTNET="192.168.102.0/255.255.255.0"
# Note: in case of ADSL connection with dynamic IP address
# EXTIF="ppp0"
# EXTNET="0/0"
```



Iptables

```
## Set Kernel parameters and load kernel module
# Enable IP forwarding (Essential!)
echo "1" > /proc/sys/net/ipv4/ip_forward
# Do not reply broadcast ping
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Enale protection from bogus error messages
echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
# Do not accept icmp redirect
echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects
# Log “strange” packets
echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
# Load kernel modules
modprobe ip_tables
modprobe iptable_nat
modprobe ip_conntrack
modprobe ipt_MASQUERADE
```



Iptables

```
# Reset rules and counters
iptables -t filter -F
iptables -t filter -X
iptables -t filter -Z
iptables -t mangle -F
iptables -t mangle -X
iptables -t mangle -Z
iptables -t nat -F
iptables -t nat -X
iptables -t nat -Z
```

La tabella `mangle` viene usata nel caso in cui si vogliano manipolare i pacchetti.

iptables da la possibilità di modificare alcuni campi dei pacchetti:



Iptables

```
# Set default Policies
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t mangle -P OUTPUT ACCEPT
iptables -t mangle -P INPUT ACCEPT
iptables -t mangle -P FORWARD ACCEPT
iptables -t mangle -P POSTROUTING ACCEPT
iptables -t mangle -P PREROUTING ACCEPT
```

```
## Enable loopback traffic
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```



Iptables

```
## Traffic to/from firewall
# Address from which it is possible to access the firewall (ssh)
ADMIN="192.168.100.2"
## INPUT
# From internal network (ssh)
iptables -A INPUT -s $ADMIN -i $INTIF -p tcp --dport 22 -j ACCEPT
# Accept traffic related to already established connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# Enable ICMP (e.g., ping) from INTNET and DMZ
iptables -A INPUT -p icmp -i $DMZIF -j ACCEPT
iptables -A INPUT -p icmp -i $INTIF -j ACCEPT
## OUTPUT
## (sanity check)
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
## LOCAL LAN NATTING (MASQUERADING)
iptables -t nat -A POSTROUTING -o $EXTIF -s $INTNET -j SNAT --to-source $EXTIP
```



Iptables

```
## Traffic from LAN to DMZ and EXT,
iptables -A FORWARD -s $INTNET -i $INTIF -p tcp --dport 22 -j ACCEPT #(SSH)
iptables -A FORWARD -s $INTNET -i $INTIF -p tcp --dport 25 -j ACCEPT #(SMTP)
iptables -A FORWARD -s $INTNET -i $INTIF -p tcp --dport 110 -j ACCEPT #(POP3)
iptables -A FORWARD -s $INTNET -i $INTIF -p tcp --dport 143 -j ACCEPT #(IMAP2)
iptables -A FORWARD -s $INTNET -i $INTIF -p tcp --dport 995 -j ACCEPT #(POP3 over SSL)
iptables -A FORWARD -s $INTNET -i $INTIF -p tcp --dport 993 -j ACCEPT #(IMAP2 over SSL)
iptables -A FORWARD -s $INTNET -i $INTIF -p udp --dport 53 -j ACCEPT #(DNS)
iptables -A FORWARD -s $INTNET -i $INTIF -p tcp --dport 80 -j ACCEPT #(HTTP)
iptables -A FORWARD -s $INTNET -i $INTIF -p tcp --dport 443 -j ACCEPT #(HTTPS)
iptables -A FORWARD -s $INTNET -i $INTIF -p icmp --icmp-type echo-request -j ACCEPT #(outgoing Ping)
iptables -A FORWARD -s ! $INTNET -i ! $INTIF -p icmp --icmp-type echo-reply -j ACCEPT #(Ping Reply)
iptables -A FORWARD -s ! $INTNET -i ! $INTIF -p icmp --icmp-type time-exceeded -j ACCEPT #(Traceroute reply)
iptables -A FORWARD -s ! $INTNET -i ! $INTIF -p icmp --icmp-type destination-unreachable -j ACCEPT
```

Per il portforwarding evidenziato nella figura:



Iptables

```
## Complete access from administrator to DMZ and EXT
iptables -A FORWARD -s $ADMIN -j ACCEPT
## Traffic from DMZ to EXT,
## (e-mail, web, dns and ssh)
iptables -A FORWARD -s $DMZNET -i $DMZIF -d $EXTNET -o ! $INTIF -p tcp --dport 25 -j ACCEPT
#(SMTP)
iptables -A FORWARD -s $DMZNET -i $DMZIF -d $EXTNET -o ! $INTIF -p udp --dport 53 -j ACCEPT
#(DNS)
iptables -A FORWARD -s $DMZNET -i $DMZIF -d $EXTNET -o ! $INTIF -p tcp --dport 80 -j ACCEPT
#(HTTP)
iptables -A FORWARD -s $DMZNET -i $DMZIF -d $EXTNET -o ! $INTIF -p tcp --dport 443 -j ACCEPT
#(HTTPS)
## Traffic from EXT to DMZ (Web only)
## Nat of web server, with public access
# private webserver IP (dmz)
webserver="192.168.101.2"
# NAT 1-*
iptables -t nat -A PREROUTING -d $EXTIP -p tcp --dport 80 -j DNAT --to-destination $webserver:80
## Access to services in DMZ
iptables -A FORWARD -d $webserver -p tcp --dport 80 -j ACCEPT
```

il traffico dall'esterno verso la DMZ lo accetto ma devo anche specificare al NAT di fare port forwarding verso lo specifico server che ho nella rete interna.



Iptables

```
# Accept traffic of established/related connections
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
## SNAT of DMZ
iptables -t nat -A POSTROUTING -s $DMZNET -j SNAT --to-source $EXTIP
# NAT 1-*
## LOG dropped packets
## unicast only (no broadcast/multicast)

iptables -A INPUT -m pkttype --pkt-type unicast -j LOG --log-prefix "INPUT DROP: "
iptables -A OUTPUT -m pkttype --pkt-type unicast -j LOG --log-prefix "OUTPUT DROP: "
iptables -A FORWARD -m pkttype --pkt-type unicast -j LOG --log-prefix "FORWARD DROP: "
```