Riassunto Network Security and Management

Tommaso Puccetti
Studente presso Universita degli studi di Firenze

November 26, 2018

Contents

List of Tables

List of Figures

1 Security Basics

Propriet della **Security**:

- **Confidentiality:** assicurare che persone non autorizzate accedano alle informazioni.
- **Integrity:** assicurare che le informazioni non vengano alterate da individui non autorizzati, in un modo che non sia individuabile dagli utenti autorizzati
- **Authentication**: Assicurarsi che gli utenti siano chi dicono di essere.

La security non deve essere confusa con la sicurezza. **Security:**

- La qualit o lo stato di essere sicuri (liberi da pericoli, da paura o ansi, libero dalla prospettiva di essere licenziato);
- Qualcosa di dato, depositato o impegnato con lo scopo di rendere un impegno un obbligo;
- Uno strumento di investimento nella forma di un contratto, che fornisce l'evidenza della sua propriet;
- Qualcosa che protegge (misure messe in atto contro lo spionaggio o sabotaggio, crimini o attacchi.)

Per quanto riguarda la safety:

- La condizione di essere sicuri rispetto al subire o causare danno, infortuni, o perdite.
- Un dispositivo progettato per prevenire operazione involontarie o pericolose.

La sicurezza ha un **costo**: un sistema sicuro **pi complesso da realizzare e da manutenere**, in definitiva **pi complesso**. BLABLABLA

2 NAT

Problema: *gli indirizzi IP sono pochi e costosi, per di pi non sempre vogliamo esporre la struttura interna di una Intranet (rete locale).*
Pe questo motivo vengono utilizzate classi di indirizzi IP (IPv4) **non-routable** come definito in **RFC 1918**, riservati alle reti locali con lo **scopo di ridurre le richieste su indirizzi pubblici**. I pacchetti con tali indirizzi per l'instradamento e l'indirizzamento tramite protocollo IP da router internet.
Si utilizza il **NAT** e **NAPT** mascherano un indirizzo tramite proxy a livello IP:

- Si trasforma un indirizzo sorgente (IP Number e port) in un altro indirizzo.
- Il server NAT viene visto all'esterno come la sorgente della comunicazione
- Il NAT **trasparente** per l'utente interno

Classi di indirizzi privati:1

Class	Private Address Range
A	10.0.0.0 - 10.255.255.255
B	172.16.0.0 - 172.31.255.255
C	192.168.0.0 - 192.168.255.255

Figure 1: Security concepts and relationships

2.1 NAT statico

- Si ha un mapping uno a uno tra indirizzi esterni ed interni.
- Pu essere utilizzato in congiunzione con un firewall.
- Non risolve il problema della scarsit di indirizzi.
- Risulta molto facile da implementare

NAT Statico:2

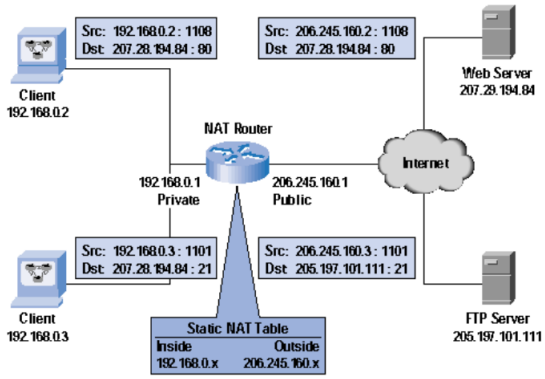

Figure 2: Security concepts and relationships

2.2 NAT Dinamico

- Mapping dinamico tra indirizzi esterni ed indirizzi esterni
- Risolve il problema della scarsit degli indirizzi
- Richiede Server stateful (mantiene informazioni di stato dell'utente durante una sessione).

NAT Dinamico:3

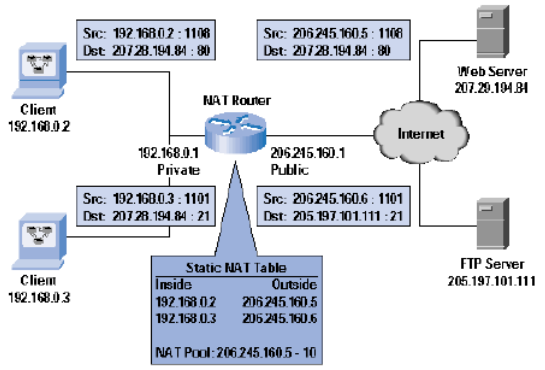

Figure 3: Security concepts and relationships

2.3 NAPT: Network Address and Port Translation

- Mapping dinamico tra indirizzi interni e d esterni **con porte dinamiche**.
- Risolve il problema della scarsit di indirizzi
- Richied un serve stateful pi complesso rispetto al NAT.

NAPT:4

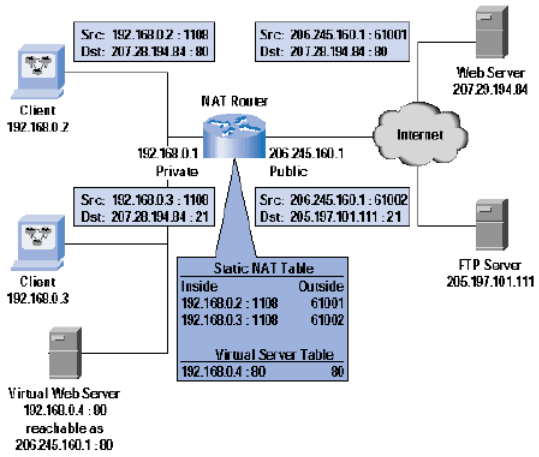

Figure 4: Security concepts and relationships

Tuttavia il NAT ha una controindicazione: implica un **ricalcolo dei checksum** IP e TCP come avviene in IPSec (standard per reti a pacchetto per la sicurezza su reti IP). Le due cose possono **interferire portando ad un completo blocco delle comunicazioni**. Si hanno problemi sia con la funzione **AH** (Authentication Header, protocollo per controllo integrit di pacchetto, garantisce authentication pacchetto per pacchetto tramite checksum a chiave simmetrica) sia con la funzione **ESP** (Encapsulating Security Payload utilizzato per autenitcit confidenzialit e integrit) di IPSEC

INSERIRE natipsec

La soluzione pu essere quella di applicare il NAT e poi IPSEC, o in alternativa eseguirli insieme. Da tenere in considerazione il fatto che un Host dietro a un NAT non pu cominciare una comunicazione IPSec (???????).

Inoltre la **co-locazione di NAT e IPSEC un potenziale pericolo per la sicurezza**. La terza opzione quella di utilizzare un tunnel IP-over-IP ma deprecabile.

2.4 Basi

I pacchetti non routable non sono trasportati in internet (ovvero i router li scartano) poich il loro indirizzo IP non univoco. Serve pertanto un traduzione da indirizzi non routable a indirizzi routable: il NAT si occupa di questo.

INSERIRE basi

Un NAT svolge le seguenti **operazioni** sia quando arriva un pacchetto sull'interfaccia **interna** che su quella **esterna**:

- Si cerca un **binding** se c' si trasla il pacchetto e si esegue un **forward** di quest'ultimo, altrimenti si **scarta** il pacchetto
- Allo scadere di un **timer** specifico si **cancella il binding**

2.5 NAT RFC 1631 e RFC 2776

Per quanto riguarda **RFC 1631** si varia **solo gli indirizzi IP**, in questo modo tuttavia non risolviamo il problema della scarsit di indirizzi. Infatti il numero di indirizzi necessari pari al numero di PC che vogliono utilizzare *contemporaneamente* lo stesso protocollo.

INSERIRE 1631

In RFC 2776 invece, si varia **sia indirizzi IP che le porte**. In tal modo il numero delle sessione contemporanee (ovvero il numero di bindings contemporanei) pari circa a 64000 (escludendo le porte ben conosciute.)

2.6 NAT binding

Per binding intendiamo una relazione:

$$IP, proto, port(int) <=> IP, Proto, Port(ext)$$

In realt quello che inteso come bindig composto da **Binding + Filter**.

- Il primo associa indirizzo porta interna a un indirizzo porta esterna (realizza la funzione $interno <=> esterno$)
- Il secondo decide se e quali pacchetti dall'esterno vanno ritradotti. ***Attenzione***: il comportamento del filter genera differenti comportamenti del NAT, alcuni voluti altri no.

Il binding varia a seconda dei protocolli che utilizziamo, nello specifico parliamo delle differenze che si riscontrano tra **TCP** e **UDP** a livello di NAT:

- TCP **stateful**, dunque il binding aggiornato in base ad un timer che varia a seconda dello stato della connessione e della dimensione della CWIN. Per questo protocollo il NAT ha un comportamento **symmetric** ossia **binding e filter sono basati sulla stessa quintupla**

  $$(protocollo, IP, portesorgente - destinazione)$$

  (Quintupla ?????)
  Per questo motivo **le comunicazioni devono partire dall'interno** e non possbili effettuare una callback, quindi PASSIVE FTP (????????). Inoltre il **demultiplexing definito a livello TCP**
- UDP **stateless**, il binding basato solo su un timer e sulla conoscenza del comportamento dell'applicazione (informazioni sulle porte utilizzate ad esempio). Il **demultiplexing** fatto a **livello applicazione**, in questo modo una sola applicazioni pu utilizzare una sola socket in uscita per due stream diversi con destinatari diversi (a differenza di TCP).

Abbiamo bisogno di un comportamento diverso del NAT per UDP. Esistono diversi modi di implementare NAT per UDP, queste diverse implementazioni dipendono dalle modalit di esecuzione del Filter. In base a come si comporta NAT alcuni applicativi possono funzionare o meno, in parte o del tutto.

2.6.1 Symmetric NAT

Funziona esattamente come il symmetric per TCP, non funzioneranno i programmi che hanno bisogno di referral e handover (?????)

INSERIRE sym

2.6.2 Full Cone NAT

Il filter non fa niente. Tutto e tutti potranno raggiungere il sorgente (compresi malintenzionati, permetto perfino di eseguire un **port scanning**)

INSERIRE cone

2.6.3 Restricted Cone NAT

Il **Filter basato sull'IP del destinatario.** Significa che accettiamo comunicazioni da porte diverse purch abbiano lo stesso IP (provengano dallo stesso Host). **Non c' controllo sul numero di porta**. Questa politica del Filter restrittiva poich non permette a programmi come MSN e mulo di funzionare

INSERIRE rest

2.6.4 Port Restricted Cone NAT

Il filter basato sulla porta del destinatario. Funzionano tutti i programmi UDP anche se con delle limitazioni.

INSERIRE port

2.6.5 NAT - STUN

Come pu un'applicazione conoscere il tipo di NAT ?
Si utilizza un protocollo chiamato **STUN**, un protocollo **request-reply**. Esso permette alle applicazioni in esecuzione su un computer di scoprire la presenza ed i tipi di NAT e firewall che si interpongono tra il computer e la rete pubblica. Permette inoltre a questi computer di conoscere gli indirizzi IP e le porte con cui il dispositivo NAT li sta rendendo visibili sulla rete pubblica. **Ha a disposizione due porte sul client e due porte e due indirizzi ip sul server**
STUN non grarantisce una conoscenza accurata, infatti il **NAT pu essere non deterministico**, ossia cambiare il comportamento a seconda della

disponibilit delle risorse. Un altro problema si riscontra nella possibilit che ci siano pi NAT nel percorso sorgente-destinazione, in questo caso la classificazione non rigorosa e il comportamento imprevedibile (Il secondo livello di NAT potrebbe non avere lo stesso comportamento del primo)

2.7 NAT: ulteriori classificazioni

I NAT possono essere classificati in base a tre parametri:

- Come viene fatto il **binding**.
- Come vengono **aggiornati i filters.**
- Quando si riavviano i **timers.**

2.7.1 In base al Binding

- **Endpoint**: il NAT riusa il binding per tutte le sessioni provenienti da stesso IP/PORTA, IP/PORTA esterni non vengono valutati (**come full cone NAT**)
- **Endpoint**:Il NAT riusa il binding per tutte le sessioni provenienti dalla stesso IP/porta verso lo stesso IP esterno (la porta non si considera). **E come un Restricted Cone NAT.**
- **Endpoint address and port dependent**: come symmetric NAT.

2.7.2 In base al Port Binding

- **Port preservation**: Il NAT tenta di mantenere la porta di origine. Se due Host interni utilizzano la stessa porta di origine uno l'avr cambiata l'altro no.
- **Port overloading**: Il NAT fa port preservation in modo aggressivo, un secondo tentativo di binding fa scadere il binding esistente
- **Port**: ??????

2.7.3 In base al Timer Refresh

- **Bidirectional**: il timer aggiornato dai pacchetti in entrambe le direzioni.
- **Outbound**: Solo pacchetti interno verso l'esterno rinfrescano i timer. Risulta necessario usare un **keep alive**. Inoltre il timer potrebbe essere per session o per binding (nel caso di riuso del binding per piu sessioni)
- **Inbound**: solo i pacchetti dall'esterno verso l'interno rinfrescano il timer, anche in questo caso c' bisogno di un keep- alive
- **Transport protocol state**: come in TCP ma si possono usare altre informazioni (da la possibilit di fare attacchi DOS).

2.7.4 In base all'External Filtering

- **Endpoint independent**: non filtra o scarta pacchetti (full cone)
- **Endpoint address dependent**: Filtra i pacchetti che non provengono dall'IP originario del binding (restricted cone).
- **Endpoint address and port dependent**: Filtra i pacchetti che non provengono dall'IP/porta originario del binding (port restricted cone o symmetric).

2.8 Considerazioni

Per quanto riguarda le **applicazioni p2p** esse tendono ad aggirare il NAT ma cos facendo creano spesso problemi di sicurezza. Per quanto riguarda ICMP rischia di fallire per lo stesso motivo di IPSEC (nel payload sono spesso contenute info su IP e porta originante). Rispetto all'**IP fragmentation** il problema quello di ricostruire i pacchetti (o almeno mantenute informazioni sul primo pacchetto), perch nei frammenti successivi **manca header UDP/TCP**, ma **potrebbe essere un attacco a frammentazione**. Inoltre il primo pacchetto pu arrivare fuori sequenza. Una soluzione quella di provare a configurare il nat in modo che esso stesso modifichi il contenuto del payload.

2.8.1 NAT: UPnP e IGD

Universal Plug n Play: Set di protocolli per la definizione e l'annuncio di device e servizi. Un dispositivo compatibile UPnP pu unirsi dinamicamente ad una rete, ottenendo un indirizzo IP, annunciare il suo nome, trasmettere le proprie capacit su richiesta e venire a conoscenza della presenza e delle capacit degli altri device della rete.
L'**Internet Gateway Device (IGD)** permete ad un device UPnP di scoprire l'indirizzo esterno di un NAT e di creare filters e bindings per i suoi servizi in modo automatico. In questo modo le porte sono aperte in modo incontrollato e potrebbero sovrascrivere i binding esistenti... come per la porta 80 (implementato in Windows).