Sicurezza dei sistemi e delle reti[1]

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

a.a. 2015/16

Lezione VIII: Complessità del filtering

LPP

Principio del Least Privilege (LPP):

> *"ogni attore dispone del minimo dei privilegi necessari per raggiungere gli obiettivi assegnatigli dalle specifiche del sistema"*

È molto difficile da applicare: c'è una costante tensione fra flessibilità e sicurezza.

Protocolli firewall-friendly

Protocolli come Telnet, SSH, rlogin, etc. sono semplici da gestire:

- per loro natura implicano ruoli ben definiti del client e server
- il pattern di scambio di messaggi è un semplice request/reply

In generale invece esistono protocolli molto piú elaborati che richiedono politiche assai piú sofisticate per applicare il LPP.

SMTP

Politica: Nella rete aziendale un solo server SMTP è autorizzato a gestire la posta elettronica con l'esterno.

- SMTP: protocollo firewall-friendly
- Client interni alla rete non passano per il firewall

SMTP

Primo tentativo: In analogia con quanto fatto per SSH
smtpSrv := 159.149.70.23
External := not(159.149.70.0/24)

verso	IP src	IP dst	prot.	port src	port dst	flag	azione
IN	External	smtpSrv	TCP	> 1023	25	1/0	Permit
OUT	smtpSrv	External	TCP	25	> 1023	1	Permit
Any	Any	Any	Any	Any	Any	*	Deny

È corretto? No: le connessioni SYN vengono bloccate

SMTP

Le regole devono essere necessariamente piú sofisticate perché vogliamo:

- Scambiare posta: un Mail Server (MS) riceve e invia posta "da" e "verso" altri MS.
- Ricevere posta: MS si connettono al MS aziendale agendo da client.
- Inviare posta: il MS aziendale si connette ad altri MS agendo da client.

Il tipo di connessioni da gestire non è uno solo!

SMTP

Secondo tentativo:

verso	IP src	IP dst	prot.	port src	port dst	flag	azione
IN	External	smtpSrv	TCP	Any	Any	1/0	Permit
OUT	smtpSrv	External	TCP	Any	Any	1/0	Permit
Any	Any	Any	Any	Any	Any	*	Deny

SMTP

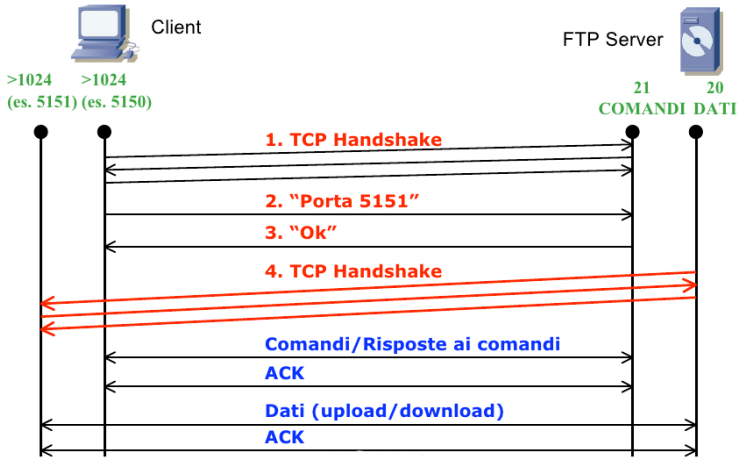
verso	IP src	IP dst	prot.	port src	port dst	flag	azione
IN	External	smtpSrv	TCP	> 1023	25	1/0	Permit
OUT	smtpSrv	External	TCP	25	> 1023	1	Permit
OUT	smtpSrv	External	TCP	> 1023	25	1/0	Permit
IN	External	smtpSrv	TCP	25	> 1023	1	Permit
Any	Any	Any	Any	Any	Any	*	Deny

Riassumendo

- Un principio che dovrebbe ispirare la scrittura delle policy è il Least Privilege
- Non è banale l'applicazione in situazioni realistiche

FTP

FTP non è un protocollo "firewall-friendly"...



FTP

verso	IP src	IP dst	prot.	port src	port dst	flag	azione
OUT	Internal	External	TCP	> 1023	21	1/0	Permit
IN	External	Internal	TCP	21	> 1023	1	Permit
IN	External	Internal	TCP	20	> 1023	**1/0**	Permit
OUT	Internal	External	TCP	> 1023	20	1	Permit
Any	Any	Any	Any	Any	Any	*	Deny

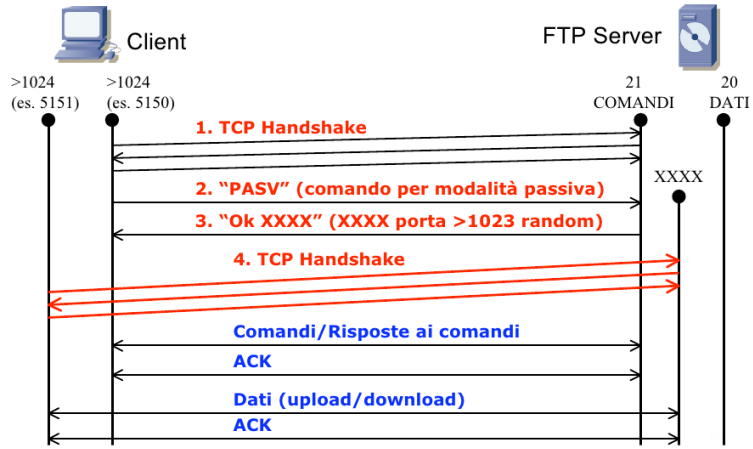
FTP

Il canale dati, dal server verso il client:
ftpserver:20 → ftpclient:XXXX
La politica di gestione "solo connessioni da interno a esterno" non è applicabile al caso in oggetto:

- connessione da esterno a interno
- porta di destinazione della connessione non determinata a priori

FTP in "passive mode"

Una nuova versione del protocollo firewall-friendly...



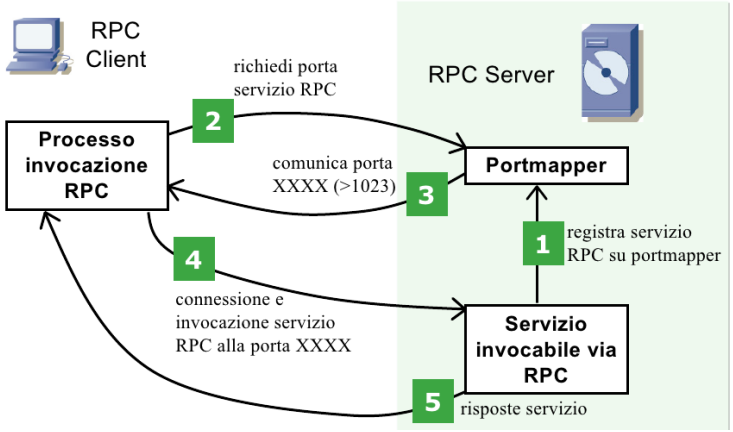
La seconda connessione, relativa al canale dati, viene aperta dal client verso il server:
ftpclient:YYYY → ftpserver:XXXX
La politica di gestione "solo connessioni solo da interno a

FTP in modo passivo

verso	IP src	IP dst	prot.	port src	port dst	flag	azione
OUT	Internal	External	TCP	> 1023	21	1/0	Permit
IN	External	Internal	TCP	21	> 1023	1	Permit
OUT	Internal	External	TCP	> 1023	> 1023	1/0	Permit
IN	External	Internal	TCP	> 1023	> 1023	1	Permit
Any	Any	Any	Any	Any	Any	*	Deny

RPC

Un protocollo complesso



RPC

Il server RPC (attraverso il servizio Portmapper, nel caso UNIX), determina dinamicamente la porta (> 1023) da assegnare al servizio RPC e quindi non si conosce a priori la porta che il server RPC assegnerà al servizio.
(Versione TCP, Unix)

verso	IP src	IP dst	prot.	port src	port dst	flag	azione
IN	External	rpcSrv	TCP	> 1023	111	1/0	Permit
OUT	rpcSrv	External	TCP	111	> 1023	1	Permit
IN	External	rpcSrv	TCP	> 1023	**Any**	**1/0**	Permit
OUT	rpcSrv	External	TCP	Any	> 1023	1	Permit
Any	Any	Any	Any	Any	Any	*	Deny

Riassumendo

Alcuni protocolli risultano piú difficili da gestire

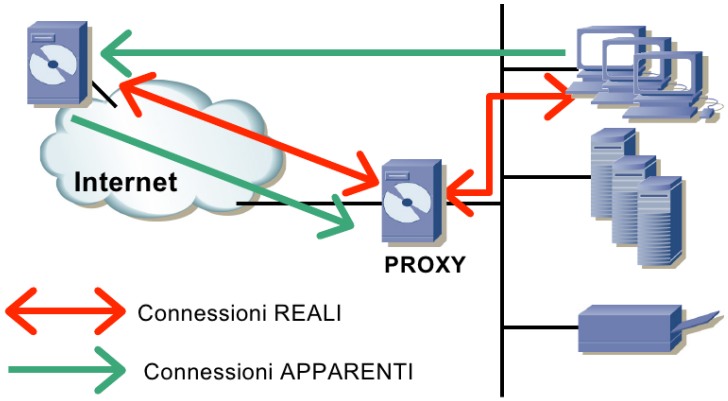
- FTP "attivo"
- RPC

Proxy

Un proxy è un componente che media le comunicazioni tra altri due componenti che rimangono inconsapevoli della sua presenza.

- Un proxy disaccoppia la comunicazione tra due componenti rendendola indiretta
- Un proxy agisce sia da client (rispetto al server originale) che da server (rispetto al client originale)

Proxy



Proxy

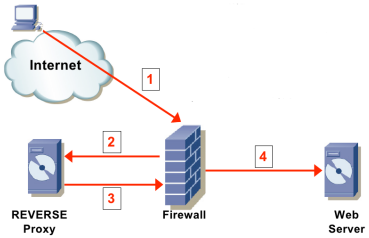
Web Proxy Cache di pagine web.

Anonymizing Proxy Anonimizzazione di connessioni web.

Reverse Proxy Gestiscono l'accesso da utenti esterni a risorse interne.

Proxy Firewall

Reverse proxy

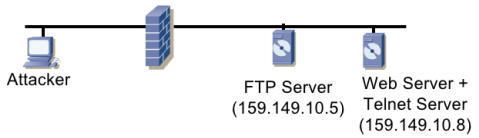


1. Connessione da utente esterno verso il Web Server
2. Redirezione della connessione verso il Reverse Proxy
3. Autenticazione, verifica, filtraggio, ecc...
4. Inoltro verso il Web Server

Proxy firewall

- Può essere usato per analizzare i dati delle applicazioni perché opera a livello applicativo
- Performance potenzialmente molto critiche
- Analogo ad un firewall stateful, ma lavora a livello del protocollo applicativo
- A volte plug-in dei firewall: *protocol decoding*

Firewall proxy per prevenire FTP bounce



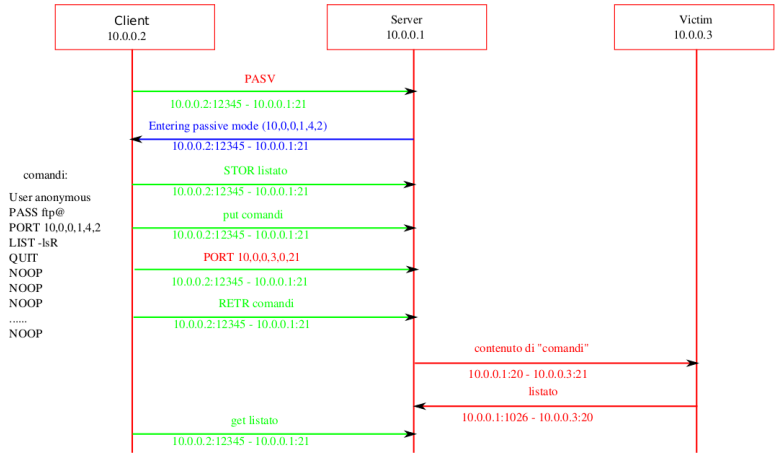
- Il comando PORT di FTP: PORT h1, h2, h3, h4, p1, p2
  - (h1, h2, h3, h4) gli ottetti dell'IP del server
  - $(256 \cdot p1 + p2)$ la porta per la connessione dal server
- PORT 159, 149, 10, 8, 0, 23 (159.149.10.8, porta 23/tcp)
- RETR: apertura di una connessione proveniente dall'FTP server

FTP bounce

Nel caso piú semplice può servire per fare una scansione:

- PORT 159, 149, 10, 8, 0, 23 (159.149.10.8, porta 23/tcp)
- L'attaccante riesce a capire se la porta 23 di 159.149.10.8 accetta connessioni

FTP bounce evoluto



Riassumendo

Un proxy

- firewall stateful che lavorano a livello applicativo
- potenzialmente molto onerosi, ma possono risultare utili quando è possibile prevedere quali applicazioni useranno gli utenti della rete

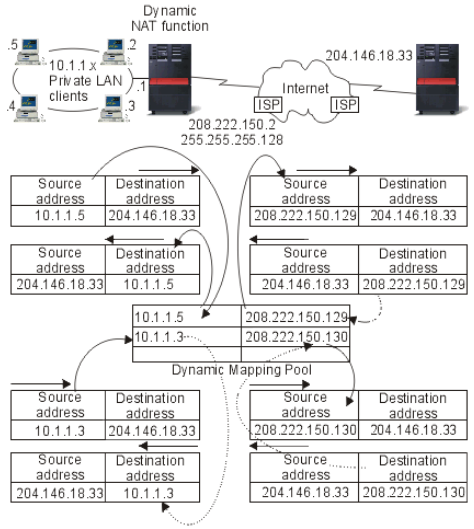
Network Address Translation (NAT) e IP Masquerading

- Consente di manipolare gli indirizzi IP nel passaggio tra le due interfacce di un firewall/router
- Tipicamente viene usato sfruttando le classi di indirizzi IP riservate e non istradabili (10., 172.16-31, e 192.168)
- Maschera gli indirizzi effettivamente utilizzati all'interno della rete

NAT

- Il router modifica gli indirizzi dei pacchetti prima instradarli
- Statico: IP interni mappati staticamente in IP pubblici.
- Dinamico: L'associazione tra IP interno e IP pubblico avviene a run-time

Dynamic NAT

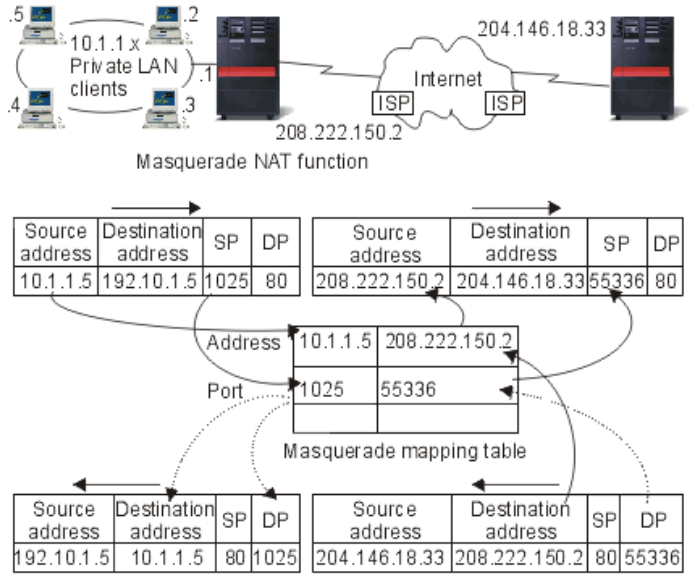


Masquerading

È una forma di NAT capace di lavorare anche ai livelli applicativi.

- Si usa la Port Address Translation: l'associazione avviene modificando la porta sorgente (p.es. ≥ 32536)
- Il masquerading proxy conosce alcuni protocolli e adatta la conversazione alle nuove condizioni (p.es. FTP)

Esempio Masquerading



Riassumendo

NAT e masquerading

- nate piú come tecniche di gestione della rete, che misure di sicurezza
- nascondo la rete interna, garantendo l'irrangiugibilità diretta

Intrusion detection system

IDS

Un sistema di monitoraggio (generalmente del tutto passivo) che genera **allarmi**

Tre fasi:

1. Raccolta dati
2. Analisi dei dati
3. Generazione degli allarmi

Perché usare un IDS?

In generale i sistemi di monitoraggio sono utili perché:

- le tecnologie di prevenzione degli eventi indesiderati o pericolosi possono fallire
- è utile avere un un meccanismo di segnalazione che permetta di attivare procedure di correzione o di emergenza
- l'uso di strumenti che permettano di monitorare lo stato corrente di un sistema, sia esso un componente che una rete, per accumulare conoscenza statistica sulle modalità d'uso

HIDS e NIDS

In base al punto in cui avviene la raccolta dati

HIDS (Host-based Intrusion Detection System) sistemi che analizzano informazioni relative all'attività locale di un singolo host (log di sistema, accesso a file critici, ...)

NIDS (Network Intrusion Detection System) sistemi che utilizzano le informazioni raccolte da analizzatori di traffico di rete.

Rilevazione di eventi critici

L'analisi dei dati raccolti per **rilevare** una situazione d'allarme:

Misuse detection Si caratterizza l'**abuso**: si rilevano le situazioni che ricadono nella descrizione di un attacco (sono detti anche signature based)

Anomaly detection Si caratterizza l'**uso normale**: si rilevano le situazioni che si scostano dal "normale" funzionamento in modo da poter rilevare anche attacchi ancora sconosciuti

Misuse detection

Elencare le situazioni illecite

Signature Detection

L'amministratore definisce pattern (signature) predefiniti di usi non conformi e il sistema analizza gli eventi monitorati(di rete, di sistema, nei log) rispetto all'elenco di pattern

È la tecnica piú affermata e diffusa

Problemi

I misuse detection rilevano solo attacchi che corrispondono a schemi noti

- Regole rigide non rilevano attacchi non noti e varianti (a volte l'IDS è cosí rigido che basta cambiare un bit per evadere la rilevazione)
- Piú le regole sono flessibili e piú aumenta la complessità di gestione/configurazione
- l'elenco di firme deve essere adattato alle specificità della rete monitorata

Anomaly detection

Anomalie rispetto

- ad eventi singoli: esempio azioni "anomale" di un utente rispetto un profilo d'uso predefinito
  es: `http://example.com/##&<>623??%`
- a dati aggregati: tipicamente, deviazioni rispetto a parametri statistici
  es: il traffico con sorgente 123.45.67.88 è di 5GB al minuto

Anomaly detection

- \+ Non dipende dalla conoscenza puntuale di tutte le modalità di intrusione
- – Molto complesso da realizzare (come fare il modello dell'uso "normale"?) e oneroso da gestire
- Non forniscono informazioni su quale vulnerabilità l'attaccante intende colpire

Modelli di uso normale

Il sistema monitorato inizialmente durante gli usi normali.

- L'ipotesi di assenza di compromissione e normalità non è facile da verificare: in fase di test si rischia l'anomalia
- Impiego di tecniche come data mining, analisi bayesiana, ecc...
- È molto complesso tenere il passo con l'evoluzione del sistema
- il monitoraggio introduce inefficienze maggiori rispetto ai misuse

Usi consolidati dell'anomaly detection

Ci sono casi in cui l'anomaly detection funziona bene e il loro uso è ormai standard

- hashing dei file (HIDS): l'integrità dei file del sistema controllata con hash da una distribuzione originale (es. Tripwire)
- Protocol Anomaly Detection (NIDS): analizzato il traffico di rete rispetto alle specifiche del protocollo applicativo

Riassumendo

- HIDS e NIDS
- Misuse e anomaly detection