Sicurezza delle reti[1]

Mattia Monga

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

a.a. 2010/11

Lezione VIII: Sicurezza perimetrale

Stateless filtering TCP

In generale, le ACL per il filtraggio stateless di pacchetti dovranno specificare:

verso	IP src	IP dst	prot.	port src	port dst	flag	azione

verso IN/OUT o l'indicazione delle zone sorgente e destinazione (es. DMZ-¿Internet), o delle interfacce (es. eth0-¿eth1)

IP sorgente/destinatario indirizzi (es. 159.149.10.1, 159.149.10.0/24) o *variabili* definite precedentemente

protocollo TCP, UDP, ICMP, IP

porta sorgente/destinazione valore o range (es. > 1023)

flag se è attivo ACK (solo TCP)

azione permit, deny

Uso di variabili

È molto utile l'uso di variabili per denotare valori di parametri (tipicamente indirizzi IP e sottoreti)
Ciò permette di

- scrivere politiche di tipo generale, che possono essere *istanziate* sulla specifica topologia di rete
- modificare indipendentemente politiche e topologia

Example

DMZ := 159.149.70.0/24
Internal := 192.168.20.0/24
Private := 10.0.0.0/8
External := not(Internal or DMZ or Private)
WebServer := 159.149.70.11 and 159.149.70.12

SSH con stateless filtering

La politica da implementare autorizza solo connessioni SSH dall'interno della rete aziendale verso l'esterno.
semplificazione: identifichiamo SSH con i pacchetti TCP con porta destinazione 22 (si noti che talvolta si cambia la porta proprio per ragioni di sicurezza!)

verso	IP src	IP dst	prot.	port src	port dst	flag	azione
OUT	Internal	Any	TCP	> 1023	22	*	Permit
IN	Any	Internal	TCP	22	> 1023	*	Permit
Any	Any	Any	Any	Any	Any	*	Deny

In realtà però possiamo notare che i pacchetti provenienti dall'esterno della rete dovrebbero essere solo risposte del server: quindi ACK deve essere settato.
Inoltre solo alcuni server ssh potrebbero essere autorizzati.

SSH

sshSrvs := 159.149.70.13 and 159.149.70.42

verso	IP src	IP dst	prot.	port src	port dst	flag	azione
OUT	Internal	sshSrvs	TCP	> 1023	22	1/0	Permit
IN	sshSrvs	Internal	TCP	22	> 1023	1	Permit
Any	Any	Any	Any	Any	Any	*	Deny

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

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

Il tipo di connessioni da gestire non è uno solo!

SMTP

Secondo tentativo:

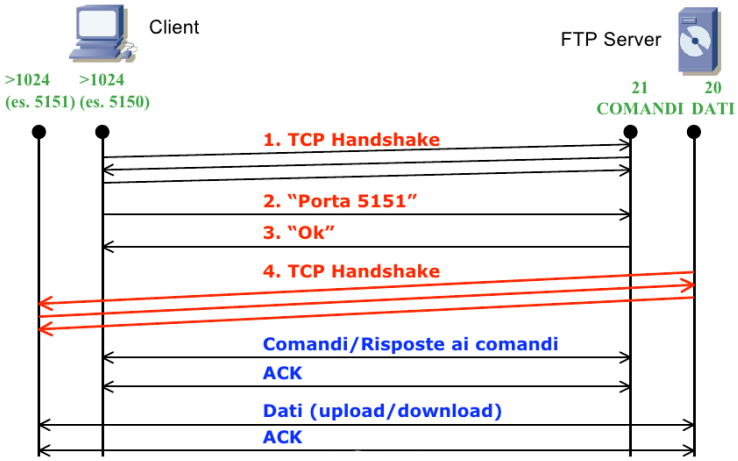
verso	IP src	IP dst	prot.	port src	port dst	flag	azione
IN	External	smtpSrv	TCP	Any	Any	1/0	Permit
OUT	smtpSrv	External	TCP	Any	Any	1/0	Permit
Any	Any	Any	Any	Any	Any	*	Deny

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

FTP

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



FTP

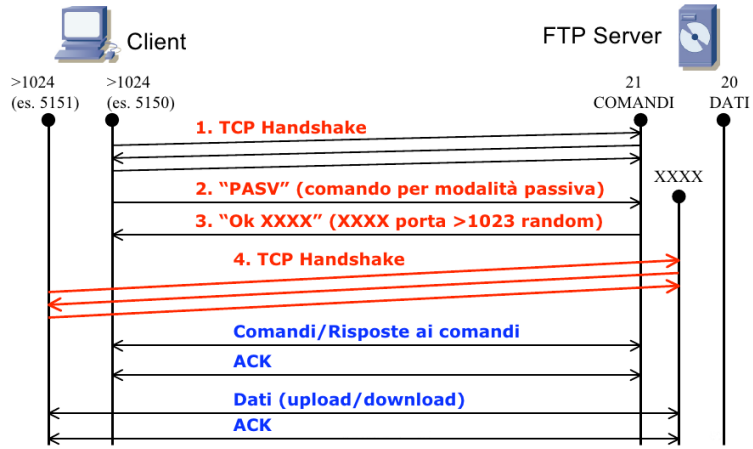
verso	IP src	IP dst	prot.	port src	port dst	flag	azione
OUT	Internal	External	TCP	> 1023	21	1/0	Permit
IN	External	Internal	TCP	21	> 1023	1	Permit
IN	External	Internal	TCP	20	> 1023	1/0	Permit
OUT	Internal	External	TCP	> 1023	20	1	Permit
Any	Any	Any	Any	Any	Any	*	Deny

La seconda connessione, relativa al canale dati, viene aperta dal server verso il client: ftpserver:20 $\rightarrow$ 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...



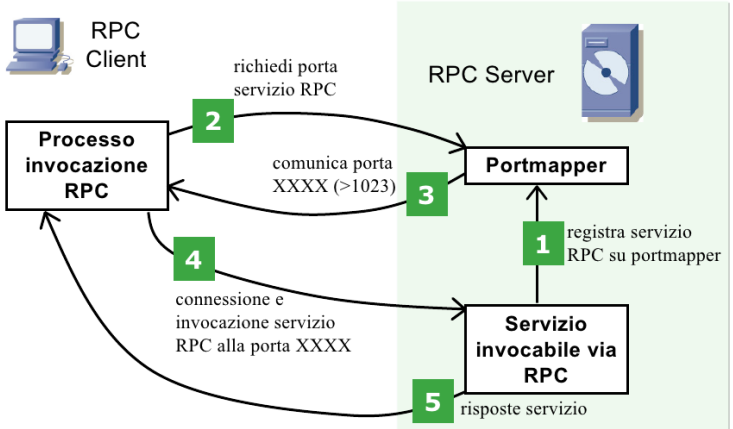
La seconda connessione, relativa al canale dati, viene aperta dal client verso il server:
ftpclient:YYYY $\rightarrow$ ftpserver:XXXX
La politica di gestione "solo connessioni solo da interno a

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