Guida alla configurazione delle Access List sul router utente

Introduzione

Nel documento si descrivono le Access Control List (ACL) e si illustra come definire una politica preventiva di sicurezza contro i più comuni attacchi via rete.

Questo documento è una guida per gli amministratori di rete per implementare una politica preventiva di sicurezza sul router di accesso alla rete GARR. Nel documento dapprima si descrivono le Access Control List (ACL) e poi si illustra come definire delle policy di sicurezza mediante ACL, equivalenti ad un un firewall, al fine di proteggere una LAN dai più comuni attacchi via rete.

Capitolo 1 - Access Control List (ACL)

Questo documento è una guida per gli amministratori di rete per implementare una politica preventiva di sicurezza sul router di accesso alla rete GARR. Nel documento dapprima si descrivono le Access Control List (ACL) e poi si illustra come definire delle policy di sicurezza mediante ACL, equivalenti ad un un firewall, al fine di proteggere una LAN dai più comuni attacchi via rete.

1. ACL (Access Control List)

1.1 Modalità di funzionamento

Le ACL servono principalmente a gestire il traffico, agendo con regole prestabilite, sugli indirizzi IP e sui servizi dei pacchetti in transito.
Le ACL vengono elaborate dal router secondo la sequenza con cui le varie clausole compaiono e al primo match si interrompe la valutazione; bisogna, pertanto, inserire prima le entry più selettive e poi quelle più generiche.
Se un'access-list è vuota, il router sottintende *permit any*, se invece, presenta anche una sola entry, il router considera un *deny any* implicito.
Le ACL per il traffico possono essere applicate, alle singole interfacce, sia in input che in output; una ACL in input fa sì che il router applichi prima l' ACL e poi effettui il routing, mentre in output prima il routing e poi l'ACL.

Le ACL possono essere:

Standard (range 1-99 e 1300-1999)

- Identificazione pi facile degli indirizzi (solo source).
- Tipicamente permit o deny di unintera famiglia di protocolli.

Extended (range 100-199 e 2000-2699)

- Identificazione complessa degli indirizzi (source e destination) .
- Tipicamente permit o deny di un protocollo specifico o di un servizio.

1.2 Le regole da ricordare per le ACL

- Evitare luso di ACL quando non necessario (appesantiscono i router).
- Porre le ACL standard in prossimità della destinazione dei pacchetti.
- Porre le ACL estese in prossimità della sorgente dei pacchetti.
- Verificare la possibilità di utilizzare una statica verso NULL0.
- Verificare sempre il funzionamento di una ACL dopo averla attivata.

Non applicare mai una ACL prima di averla definita. Una ACL non vuota presuppone lesistenza di un deny implicito, mentre una ACL vuota presuppone lesistenza di un permit implicito. Inoltre una Wildcard Mask omessa si suppone essere 0.0.0.0 (host).

Il comando per attivare una ACL è **ip access-group *acl-number* in [out]** in modalità enable sulla interface specifica.

Il comando per eliminare una ACL è **no access-list acl-number** (in questo modo si elimina tutta l'acl)

Con i recenti IOS e' possibile eliminare anche le singole entry come descritto di seguito:

```
#sh access-lists 102
Extended IP access list 102
   10 permit ip any 192.168.106.0 0.0.0.255 (23652308 matches)
   20 permit tcp any host 192.168.158.23 eq smtp (1388914 matches)
   30 permit tcp any host 192.168.158.23 established (263155 matches)
   40 permit tcp any host 192.168.158.23 eq www (38624 matches)
   50 deny ip any host 192.168.158.23 (34537 matches)
   55 permit ip any host 192.168.158.63 (726289 matches)
   60 permit tcp any host 192.168.158.239 eq 22 (290766 matches)
    ...
    ...
```

Se ad esempio vogliamo eliminare la prima riga:

```
#conf t
Enter configuration commands, one per line.  End with CNTL/Z.

(config)#ip access-list extended 102
(config-ext-nacl)#no 10
```

1.3 Configurare ed utilizzare le ACL

- Router(config)#
- access-list acl-number {permit | deny } condizione
- Router(config-if)# {protocol} access-group acl-number [in | out]

Le ACL sono numerate e si distinguono in base al numero.

Le ACL sono numerate e si distinguono in base al numero. Per le ACL IP le standard vanno da 1 a 99 e da 1300 a 1999 mentre le extended vanno da 100 a 199 e da 2000 a 2699.

Esempio di sintassi ERRATA!

- access-list 1 permit 192.168.0.0 0.0.255.255
- access-list 1 deny 192.168.1.0 0.0.0.255 (entry piu` specifica di quella precedente quindi non sara` mai valutata)

Esempio di sintassi CORRETTA!

- access-list 1 deny 192.168.1.0 0.0.0.255
- access-list 1 permit 192.168.0.0 0.0.255.255

1.4 Esempio di ACL estesa (forma generale)

Una ACL IP estesa (100-199) è composta come di seguito:

- access-list acl-number {permit | deny}
- {protocol | protocol-keyword}
- {source sorce-wildcard | any}
- {destination destination-wildcard | any}
- [protocol specific options] [log]

Protocol keyword possono essere: icmp, tcp, ed udp per ognuna di queste keyword esiste una sintassi alternativa, come specificato in seguito.

La keyword log consente di effettuare il logging dei pacchetti che matchano la ACL.

1.4.1 Esempio di ACL estesa (forma per TCP)

Una ACL IP estesa per TCP è composta nel modo seguente:

- access-list acl-number {permit | deny}tcp
- {source sorce-wildcard | any}
- {destination destination-wildcard | any}
- [operator destination-port][destination-port][established]

- operator può essere: lt,gt,eq,neq.
- destination-port: è il solito intero a 16 bit senza segno

established seleziona solo i pacchetti che presentano il bit di ACK o di RST ad 1 (tipicamente accade per le connessioni TCP che si sono già avviate). Questo permette di scrivere delle ACL in grado consentire il telnet in una sola direzione.

1.4.2 Esempio di ACL estesa (forma per ICMP)

Una ACL IP, estesa per ICMP, è composta nel modo seguente:

- access-list acl-number {permit | deny} icmp
- {source sorce-wildcard | any}
- {destination destination-wildcard | any}
- [icmp-type [icmp-code] | icmp-message]

I messaggi ICMP possono essere filtrati in base ad *icmp-type, icmp-type + icmp-code* o come *icmp-message* .

L'uso degli icmp-message semplifica notevolmente la configurazione di una ACL di questo tipo.
Alcuni esempi di icmp message:

echo	echo-reply	administratively-prohibited
time-exceeded	ttl-exceeded	unreachable
traceroute	packet-too-big	network-unknown
port-unreachable	unreachable	host-unknown
host-unreachable	net-unknown	net-unreachable

1.4.3 Esempio di ACL estesa (forma per UDP)

Una ACL IP estesa per UDP è composta come di seguito:

- access-list acl-number {permit | deny} udp
- {source sorce-wildcard | any}
- {destination destination-wildcard | any}
- [operator destination-port][destination-port]

- operator può essere: lt,gt,eq,neq .

- destination-port: è il solito intero a 16 bit senza segno la keyword established non esiste in quanto UDP è connectionless

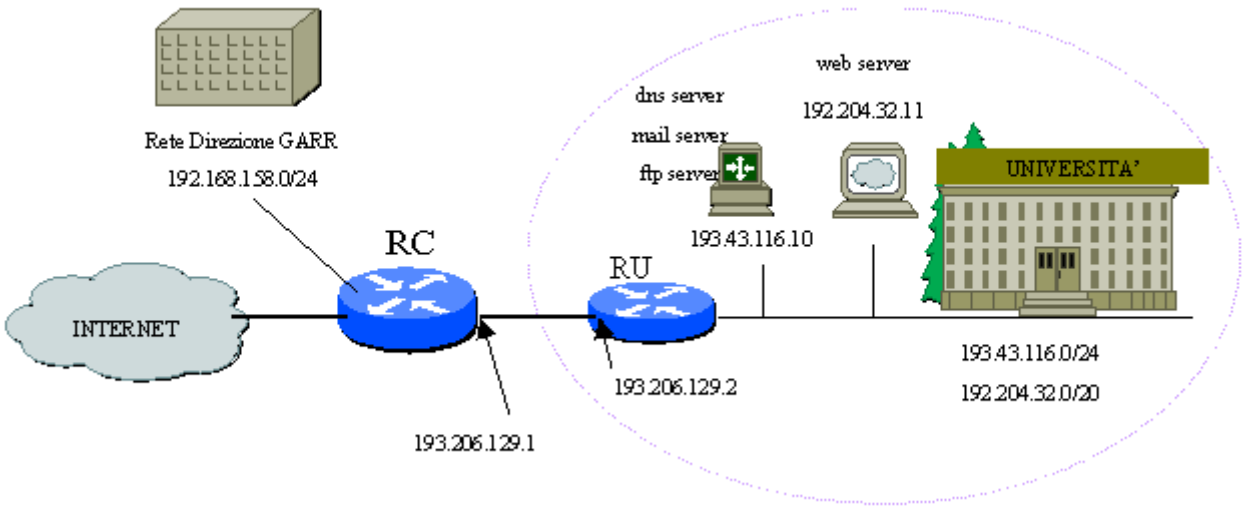
Capitolo 2 - Protezione degli host sulla LAN

E` cura del NOC del GARR comunicare all'utente:

- gli indirizzi IP sul link (lato utente e lato RC)
- indirizzo IP della rete della Direzione
- la password di primo livello per il router-utente

Nella configurazione dei filtri si considera lo schema seguente:

NB gli IP indicati sono solo di esempio



2.1 Filtri per le interfacce verso l'esterno

La protezione degli host di una LAN puo essere effettuata tramite l'utilizzo di ACL applicate in ingresso ed in uscita su tutte le interfacce che vanno verso l'esterno. Nell'esempio che segue i filtri sono denominati ACL-102 (input) e ACL-103 (output) applicati sull'interfaccia di uscita del router di frontiera.
NB: Recentemente e' possibile anche definire le “extended” mediante nome al posto dei numeri, nel nostro caso, ad esempio, le acl 102 e 103 potrebbero diventare ACL-IN e ACL-OUT. Con l'indicazione “remark” è inoltre possibile inserire dei commenti all'inizio di ogni singola entry.

I servizi che vengono filtrati in ognuna delle ACL sono:

ACCESS-LIST 102 (INput)

access-list 102 remark ssh (tcp 22) senza restrizioni
access-list 102 permit tcp any any eq 22
access-list 102 remark icmp permesso solo alle punto-punto e alla rete della Direzione GARR
access-list 102 permit icmp 192.168.158.0 0.0.0.255 any
access-list 102 permit icmp host 193.206.129.1 host 193.206.129.2

TFTP solo verso un eventuale server

access-list 102 permit udp any x.x.x.x eq 69
access-list 102 remark blocca il ping dall'esterno verso gli IP della LAN
access-list 102 deny icmp any any echo

access-list 102 remark telnet permesso solo alle punto-punto
access-list 102 permit tcp host 193.206.129.1 host 193.206.129.2 eq 23
access-list 102 remark http permesso a tutti solo verso i web server interni
access-list 102 permit tcp any host 192.204.32.11 eq 80
access-list 102 permit tcp any host "webserver2" eq 80
access-list 102 remark ftp (20, 21) permesso a tutti solo verso il server interno
access-list 102 permit tcp any host 193.43.116.10 eq 20
access-list 102 permit tcp any host 193.43.116.10 eq 21
access-list 102 remark DNS sulla porta TCP 53 ed UDP permesso a tutti verso il DNS server
access-list 102 permit tcp any host 193.43.116.10 eq 53
access-list 102 permit udp any host 193.43.116.10
access-list 102 remark smtp, pop3 e imap permesso a tutti solo verso i mailserver
access-list 102 permit tcp any host 193.43.116.10 eq 25
access-list 102 permit tcp any host 193.43.116.10 eq 110
access-list 102 permit tcp any host 193.43.116.10 eq 143
access-list 102 remark divieto a tutti per protocollo NFS (udp e tcp porta 2049), openwin (tcp porta 2000) e X11 (tcp porte 6000-6010)
access-list 102 deny udp any any eq 2049
access-list 102 deny tcp any any eq 2049
access-list 102 deny tcp any any eq 2000
access-list 102 deny tcp any any range 6000 6010

access-list 102 remark permesso per porte maggiori di 1023
access-list 102 permit tcp any any gt 1023
access-list 102 permit udp any any gt 1023

access-list 102 remark chiusura per porte minori di 1024
access-list 102 deny udp any any lt 1024
access-list 102 deny tcp any any lt 1024

```
access-list 102 remark permesso per icmp
access-list 102 permit icmp any any
```

ACCESS-LIST 103 (OUTput)

- **blocca il TFTP**

```
access-list 103 deny udp any any eq 69 log
```

- **traffico consentito in uscita alle sole reti interne verso tutti**

(Per evitare che la propria LAN sia origine di un attacco di tipo smurfing è necessario controllare che i pacchetti in uscita abbiano nell' header, come source address, un indirizzo appartenente alle reti della lan)

```
access-list 103 permit ip 193.43.116.0 0.0.0.255 any
access-list 103 permit ip 192.204.32.0 0.0.15.255 any
access-list 103 deny ip any any log
```

2.2 Filtro per account non privilegiato al router

ACCESS-LIST 2

- **telnet deve essere consentito solo ad alcuni host prestabiliti come ad esempio alla rete interna e al router del PoP (punto-punto lato GARR)**

```
access-list 2 permit 193.206.129.1 0.0.0.0
access-list 2 permit 193.43.116.0 0.0.0.255
access-list 2 permit 192.204.32.0 0.0.15.255

line vty 0 4
...
access-class 2 in
password 7
login
```

ACCESS-LIST 102 su Juniper

```
set firewall family inet filter 102 term ssh_ok from protocol tcp
set firewall family inet filter 102 term ssh_ok from destination-port ssh
set firewall family inet filter 102 term ssh_ok then accept
set firewall family inet filter 102 term icmp_ok from source-address
```

```
192.168.158.0/24
set firewall family inet filter 102 term icmp_ok from source-address
193.206.129.1/32
set firewall family inet filter 102 term icmp_ok from destination-address
193.206.129.2/32
set firewall family inet filter 102 term icmp_ok from protocol icmp
set firewall family inet filter 102 term icmp_ok then accept
set firewall family inet filter 102 term icmp_deny from protocol icmp
set firewall family inet filter 102 term icmp_deny then discard
set firewall family inet filter 102 term ssh_ok from destination-port ssh
set firewall family inet filter 102 term telnet_ok from source-address
193.206.129.1/32
set firewall family inet filter 102 term telnet_ok from destination-address
193.206.129.2/32
set firewall family inet filter 102 term telnet_ok from protocol tcp
set firewall family inet filter 102 term telnet_ok from destination-port 23
set firewall family inet filter 102 term telnet_ok then accept
set firewall family inet filter 102 term www_ok from destination-address
192.204.32.11/32
set firewall family inet filter 102 term www_ok from destination-address
“webserver2”
set firewall family inet filter 102 term www_ok from protocol tcp
set firewall family inet filter 102 term www_ok from destination-port 80
set firewall family inet filter 102 term www_ok then accept
set firewall family inet filter 102 term ftp_ok from destination-address
193.43.116.10/32
set firewall family inet filter 102 term ftp_ok from protocol tcp
set firewall family inet filter 102 term ftp_ok from destination-port [20-21]
set firewall family inet filter 102 term ftp_ok then accept
set firewall family inet filter 102 term dns_ok from destination-address
193.43.116.10/32
set firewall family inet filter 102 term dns_ok from protocol tcp
set firewall family inet filter 102 term dns_ok from destination-port 53
set firewall family inet filter 102 term dns_ok from protocol udp
set firewall family inet filter 102 term dns_ok then accept
set firewall family inet filter 102 term smtp-pop3-imap_ok from destination-
address 193.43.116.10/32
set firewall family inet filter 102 term smtp-pop3-imap_ok from protocol tcp
set firewall family inet filter 102 term smtp-pop3-imap_ok from destination-port
[25 110 143]
set firewall family inet filter 102 term smtp-pop3-imap_ok then accept
set firewall family inet filter 102 term nfs_deny from protocol tcp
set firewall family inet filter 102 term nfs_deny from protocol udp
set firewall family inet filter 102 term nfs_deny from destination-port 2049
set firewall family inet filter 102 term nfs_deny then discard
set firewall family inet filter 102 term ow-x11_deny from protocol tcp
set firewall family inet filter 102 term ow-x11_deny from destination-port [2000
```

6000-6010]
set firewall family inet filter 102 term ow-x11_deny then discard
set firewall family inet filter 102 term wellknown_deny from protocol tcp
set firewall family inet filter 102 term wellknown_deny from destination-port [1-1023]
set firewall family inet filter 102 term wellknown_deny then discard
set firewall family inet filter 102 term default then accept

ACCESS-LIST 103 su Juniper

set firewall filter deny spoofing interface specific
set firewall filter deny-spoofing term antispoofing from source-address 193.43.116.0/24
set firewall filter deny-spoofing term antispoofing from source-address 193.204.32.0/20
set firewall filter deny-spoofing term antispoofing then accept
set firewall filter deny-spoofing term antispoofing default then reject

ACCESS-LIST2 su Juniper

set firewall family inet filter NOC-access term telnet_ok from source-address 193.43.116.0/24
set firewall family inet filter NOC-access term telnet_ok from source-address 192.204.32.0/20
set firewall family inet filter NOC-access term telnet_ok from source-address 193.206.129.1/32
set firewall family inet filter NOC-access term telnet_ok from protocol tcp
set firewall family inet filter NOC-access term telnet_ok from destination-port telnet
set firewall family inet filter NOC-access term telnet_ok then accept
set firewall family inet filter NOC-access term telnet_deny from protocol tcp
set firewall family inet filter NOC-access term telnet_deny from destination-port telnet
set firewall family inet filter NOC-access term telnet_deny then discard

Applicata poi alla loopback0

set interfaces lo0 unit 0 family inet filter input NOC-access
set interfaces lo0 unit 0 family inet6 filter input NOC-access-ipv6