

sicurezza delle reti

(metodi crittografici esclusi)

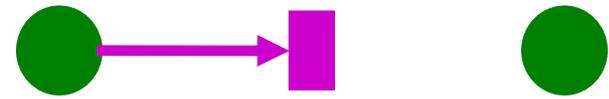
classificazione degli attacchi



sorgente
di informazioni

destinatario

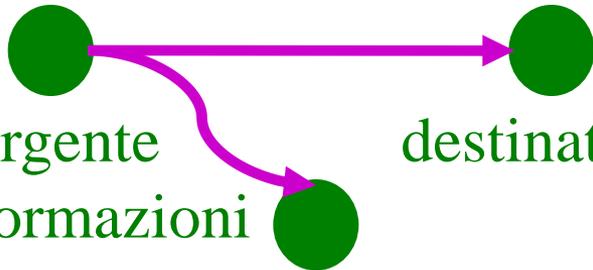
flusso normale



sorgente
di informazioni

destinatario

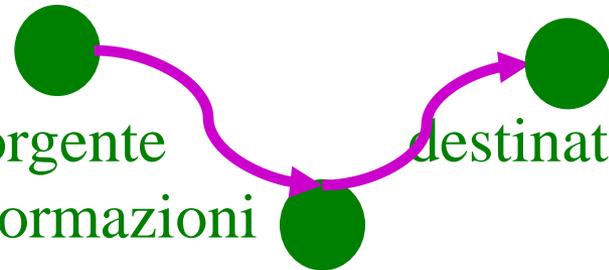
interruzione (DoS)



sorgente
di informazioni

destinatario

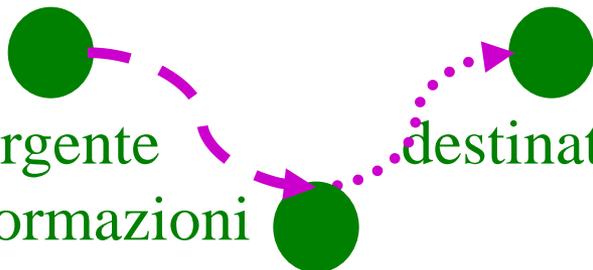
intercettazione (sniffing)



sorgente
di informazioni

destinatario

intercettazione (MiM passivo)



sorgente
di informazioni

destinatario

modifica (MiM attivo)



sorgente
di informazioni

destinatario

creazione (spoofing)

confinamento nelle reti

- il traffico è (o dovrebbe essere) ammesso solo tra sistemi e/o utenti con caratteristiche di sicurezza analoghe (principio di isolamento)
- esempi di classi di utenti
 - amministrazione
 - integrità e disponibilità: critiche per il business
 - confidenzialità: critica per legge
 - insieme di applicazioni ben definito
 - sistemi sotto controllo diretto
 - docenti
 - integrità e disponibilità: critiche per il business
 - confidenzialità: critica per certi aspetti particolari
 - richiesta alta flessibilità nelle applicazioni
 - il controllo dei sistemi è delegato ai gruppi di ricerca
 - studenti
 - best effort
 - sistemi non controllabili
 - utenti da internet di ricerca
 - servizi selezionati: web, email, siti di ricerca interni, login alle macchine di ricerca e a utenti da internet (altri)
 - servizi selezionati: web, email, siti di ricerca interni

la sicurezza nelle reti

esempi di vulnerabilità

stack protocollare

esempi di contromisure

DoS
Sniffing
MiM passivo
MiM attivo
spoofing

applicazione	applicazione
presentazione	
sessione	TCP
trasporto	
rete	IP
link	link
fisico	fisico

application gateway, autenticazione
metodi crittografici

stateful firewall, ids,
metodi crittografici

screening router, ids
nat, metodi crittografici

vlan, conf. switch,
autenticazione, metodi crittografici

isolamento del mezzo
metodi crittografici

sicurezza del mezzo trasmissivo

- cavi rame
 - vulnerabilità
 - wiretapping
 - taglio
 - emissione elettromagnetica
 - contromisure
 - protezione fisica
 - wiretapping detection
- fibra
 - vulnerabilità
 - wiretapping
 - taglio
 - contromisure
 - protezione fisica
 - wiretapping detection

sicurezza del mezzo trasmissivo

- wireless
 - vulnerabilità
 - copertura “ad area” non “a presa”
 - impossibile stabilire chi riceve il traffico
 - area di copertura varia con la sensibilità delle stazioni
 - antenne fortemente direzionali possono ricevere hot spot molto lontani
 - facile avere utenti “parassiti”
 - DoS elettromagnetico
 - non adatto a sistemi la cui disponibilità è critica
 - banda molto limitata
 - contromisure
 - confidenzialità e integrità: meccanismi crittografici a livello data link (più o meno efficaci)

livello data link

switch

- vulnerabilità
 - mac spoofing/flooding
 - impatto su arp, spanning tree, e livelli superiori
 - e altri problemi mitigabili: bugs del firmware, errori di configurazione
- contromisure
 - switch sofisticati permettono di rilevare e contrastare attacchi basati su mac spoofing
 - autenticazione
 - mac lock
 - web based (captive portal)
 - al primo accesso funziona solo dhcp e dns
 - lo switch fa da web server e tramite http chiede username e password
 - 802.1X
 - isolamento
 - VLAN
 - vulnerabilità: GVRP è abilitato di default per standard e permette la configurazione automatica di vlan, oramai obsoleto e deletereo.

virtual lan vs. real lan

- le “forze” che guidano il progetto della rete fisica e delle vlan sono diverse
- rete fisica, cioè switch e cablaggio
 - ha come obiettivo la rete economicamente più conveniente che soddisfi i seguenti vincoli
 - piena connettività tra le prese
 - prestazioni richieste
 - vincoli ambientali (es. posizione spazi adibiti agli impianti)
 - vincoli tecnologici (es. lunghezza dei cavi)
 - normative (es. materiali non infiammabili)
 - affidabilità (se richiesta, es. resiste al fault di uno switch)
- vlan, cioè configurazione ideale “desiderata”
 - ha come obiettivo la migliore configurazione che soddisfi...
 - le specifiche funzionali
 - chi deve essere connesso con chi? vedi strutture organizzative che occupano l’edificio (es. dipartimenti, reparti, ecc)
 - le specifiche di sicurezza basate sulle classi di utenti (vedi isolamento)
 - facilità di gestione (es. ciascuna vlan un amministratore)

vlan

- su un singolo switch
 - ciascuna porta è associata ad una vlan
 - uno “switch virtuale” per ciascuna vlan
- su più switch: 802.1Q
 - tra uno switch e l’altro i pacchetti viaggiano taggati con l’identificatore della vlan (802.1p)
- comportamenti caratteristici
 - confinamento di broadcast e unicast
 - switch di scarsa qualità (non standard) confinavano solo il broadcast (vulnerabile)
 - la comunicazione tra vlan deve avvenire attraverso un router proprio come tra lan distinte
- argomenti correlati
 - vlan asimmetriche, vlan per protocollo, GVRP, multiple spanning tree

Wi-Fi

- autenticazione
 - web based (captive portal)
 - al primo accesso funziona solo dhcp e dns
 - lo switch fa da web server e tramite http chiede username e password
 - 802.1X
 - in ogni caso basati su metodi crittografici
- confidenzialità, integrità e isolamento solo con metodi crittografici

livelli rete, trasporto e applicazione

vulnerabilità e minacce

- veicolo per attacchi ai sistemi
- elemento umano
 - email, web, ecc. l'utente permette l'entrata di virus, trojan, ecc.
- molte vulnerabilità già viste all'inizio del corso
 - protocolli in chiaro e non autenticati: necessari metodi crittografici
 - DoS, DDoS

firewalls

- apparecchiature che confinano (filtrano) selettivamente il traffico di rete
- network (layer3+4) vs. application (layer7)
 - network: stateful vs. stateless
- hardware vs. software
- personal fw vs. network fw

- possono avere altre funzionalità
 - nat, virtual private network, autenticazione utenti

Unified Threat Management (UTM)

- evoluzione del concetto di firewall
- unisce molte delle funzionalità di sicurezza relative alle reti
 - firewall
 - network intrusion detection/prevention
 - sicurezza delle email
 - antivirus, anti-spam
 - VPN termination
 - applicative content inspection
 - load balancing
- semplicità di gestione e riduzione dei costi
- vedremo molte delle funzionalità come se fossero apparati separati

soggetti, oggetti e diritti in un fw

- soggetto: un pacchetto/messaggio in ingresso
- oggetto: solo il firewall
- accesso: richiesta al firewall di essere instradato verso la destinazione
- diritti: per ciascun tipo di traffico in ingresso le destinazioni ammesse
- è un modo di vedere il fw molto concreto ma poco utile

il firewall come reference monitor

un firewall può essere visto come un reference monitor di rete

- soggetto: l'host sorgente che ha inviato il pacchetto/messaggio
- oggetto: l'host destinazione che riceverà il pacchetto/messaggio
- accesso: richiesta all'host destinazione di processare il pacchetto/messaggio inviato dall'host sorgente
 - caratterizzato dal valore di campi del pacchetto stesso
 - la semantica vera (cioè l'effetto sull'host destinazione) dell'accesso è data dal protocollo di rete e dal contesto in cui viene usato: il fw non ha bisogno di conoscerla
- diritti: categorie di traffico ammesso per la coppia di host

stateless firewall

- regole di “packet filtering”
 - basate sulla quadrupla
<saddr,sport,daddr,dport>
 - in pratica: router + access control list
 - detti anche “screening routers”
- non mantengono alcuno stato
- tipicamente quando si parla di firewall si fa riferimento a **firewall “stateful”**

stateful firewall

- si ricordano le “connessioni”
 - stato della connessione: new, established, ecc.
- ciascun pacchetto è assegnato ad una connessione
 - filtrano in base allo stato della connessione
 - es. ammesso da esterno verso interno solo se connessione è established
 - verificano la correttezza del protocollo
 - es. syn ammesso solo per connessioni nuove
 - permettono regole in base allo stato della connessione
- possono modificare il traffico
 - es. per implementare schemi anti-SYNflood

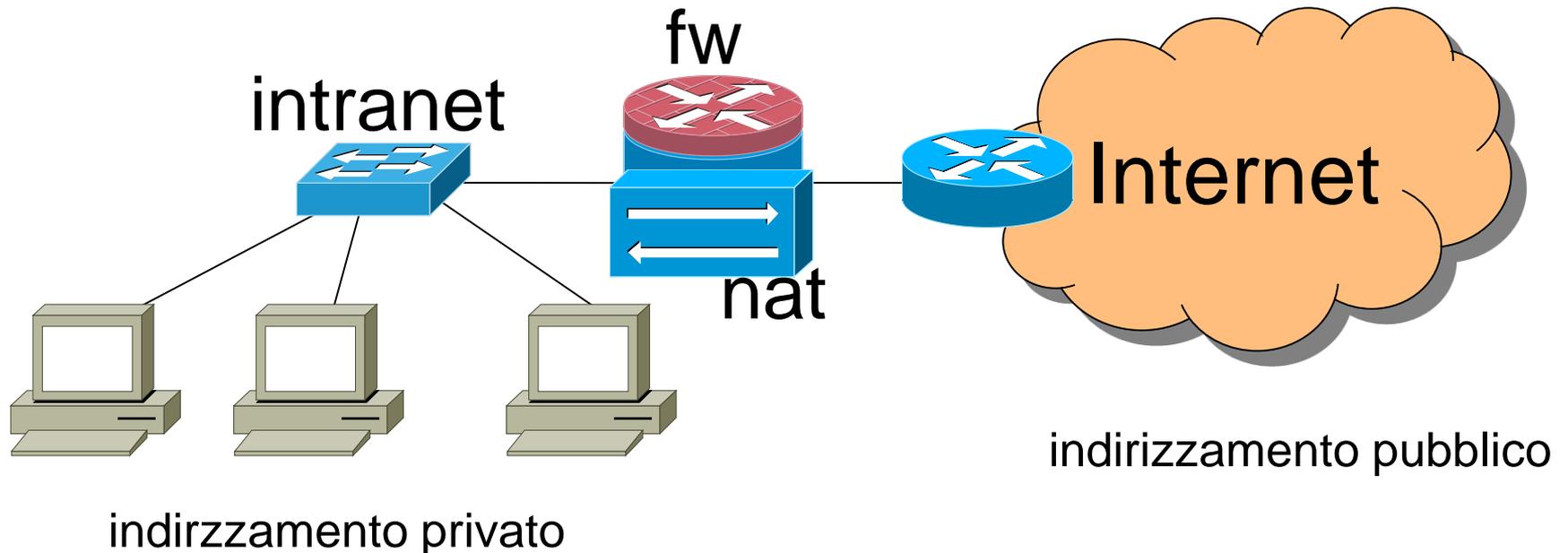
connessioni

connessione: un insieme di pacchetti che “hanno lo stesso scopo”

- connessione per i protocolli, vedi standard:
 - TCP connesso, UDP non connesso
- connessione per il sistema operativo
 - il sistema operativo mantiene uno stato per i protocolli connessi
 - per il sistema operativo esiste una versione connessa di UDP
- connessione per il firewall: definita appositamente
 - **il primo pacchetto della conversazione crea la connessione**
identificata con la quadrupla <saddr,sport,daddr,dport>
 - **gli altri pacchetti con tale quadrupla sono relativi alla connessione già creata**
 - es. per un firewall ping e le richieste DNS (su udp) creano “connessioni”

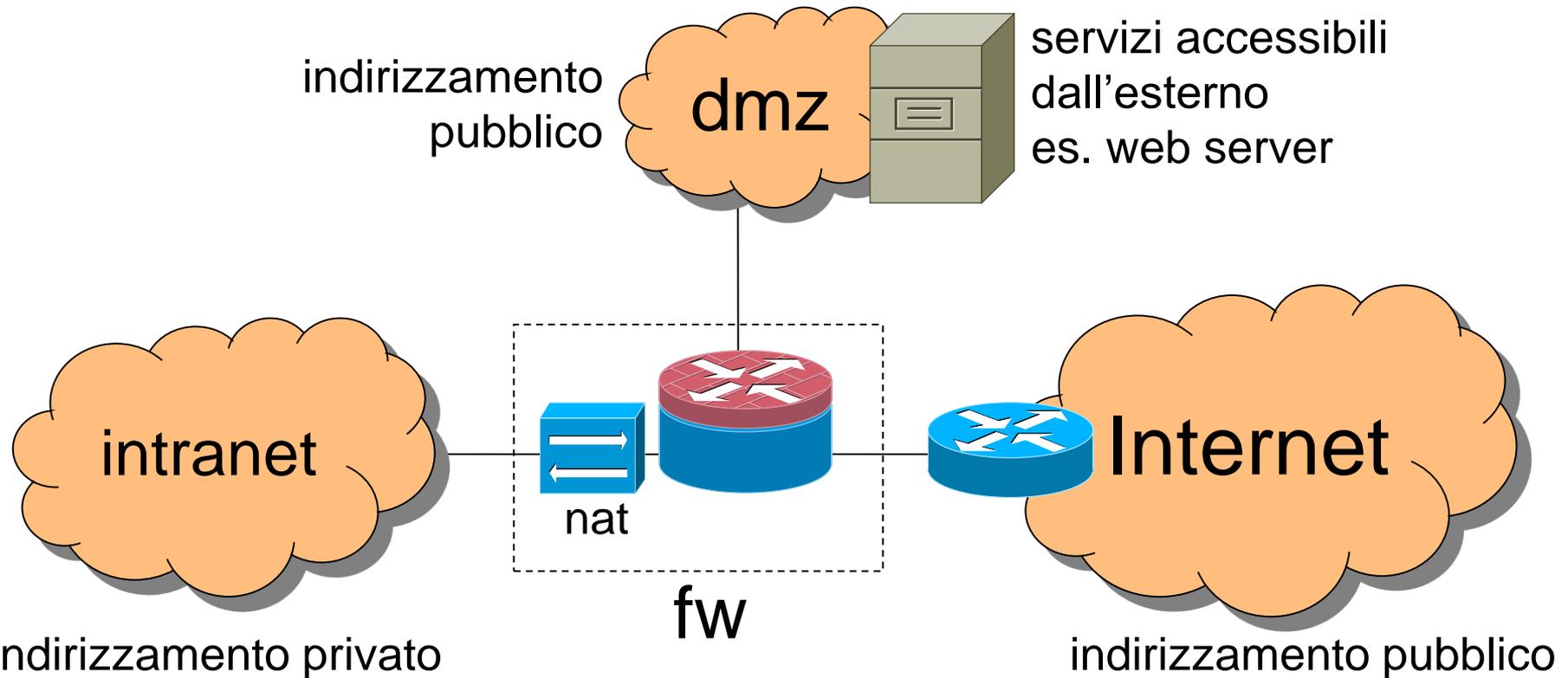
firewall, nat e intranet

- un semplice caso d'uso
 - isolare la intranet da Internet
 - in questa configurazione tipicamente il fw fa anche nat ma non è strettamente necessario



dmz

- demilitarized zone (zona smilitarizzata) o perimeter network (rete perimetrale)
 - rete distinta sia da Internet che dalla intranet
 - gli host in tale rete sono in una classe di sicurezza distinta da quelli della intranet e di Internet



dmz: esempio di policy

da \ a	host intranet	host dmz	host Internet
host intranet	<p>all</p> <p>non si passa per il fw</p>	<p>richiesta</p> <p>dalla intranet si accede ai servizi della dmz</p>	<p>-</p> <p>dalla intranet non si accede a Internet</p>
host dmz	<p>risposta</p> <p>dalla dmz si risponde alle richieste della intranet</p>	<p>all</p> <p>non si passa per il fw</p>	<p>risposta</p> <p>dalla dmz si risponde alle richieste di Internet</p>
host Internet	<p>-</p> <p>da Internet non si accede alla intranet</p>	<p>richiesta</p> <p>da Internet si accede ai servizi della dmz</p>	<p>all</p> <p>non si passa per il fw</p>

interpretazione per protocolli basati su tcp

- “richiesta”
 - un syn per una connessione NEW
- “risposta”
 - tutto ciò che è relativo ad una connessione ESTABLISHED (dopo il syn)
- facile da implementare in un firewall stateful

varianti

- intranet potrebbe dover accedere ad Internet
 - questo potrebbe rappresentare un grave problema di sicurezza perché Internet è una fonte di input non fidato
 - consigliato l'uso di application level gateway o application level packet inspection
- dmz potrebbe dover accedere a...
 - dns, per i log
 - fonte non fidata
 - servizi di altri fornitori
 - il grado di fiducia può variare

problemi

- la configurazione può essere molto complessa
 - facile fare errori
 - risoluzione DNS:
 - non fidata
 - quando farla?
 - inefficiente
 - configurazione deve essere fatta con indirizzi IP
 - o con nomi configurati staticamente

firewall e DDoS

- DDoS che saturano la banda
 - il firewall non aiuta
 - perché il DDoS è di fatto sull'infrastruttura di rete
 - anycast e CDN sono approcci migliori ma costosi
 - esistono specifici servizi in cloud che fanno economia di scala
- DDoS sulle risorse del server ma senza spoofing
 - rilevamento in base all'attività delle sessioni (aging)
 - contrasto mediante white/black-list gestite in modo automatico

syn-flood e syn-proxy

- **syn-flood**: DDoS che inviano tanti syn da indirizzi spoofati
- **syn-proxy**: un firewall che protegge il server da syn malevoli
- come funziona
 - quando arriva un syn: il fw risponde syn-ack e non inoltra il syn al server
 - quando arriva l'ack, il fw riconosce la sessione come buona e apre una sessione tcp con il server
 - da ora in poi il fw fa da proxy tra le due sessioni
 - il fw è progettato per scalare sulle connessioni “mezze aperte” (ricevuto solo il syn)

syn cookies

- **syn cookies:** contromisura per attacchi syn-flood
 - implementata su server o in fw con funzione di syn-proxy
 - una delle poche soluzioni in grado di discriminare il traffico lecito da quello non lecito in maniera automatica
 - nessuno stato da mantenere per rispondere al syn!
 - ... e quini scalabile all'infinito rispetto al numero delle connessioni mezze aperte
-
- implementato in Linux
 - `echo 1 > /proc/sys/net/ipv4/tcp_syncookies`

syn cookies

- lo stato è codificato nel sequence number di tcp
- per le connessioni «buone» il server ottiene nella risposta lo stato che avrebbe dovuto mantenere
 - il syn+ack contiene il sequence number +1
- **contrasto a ack malevoli**
 - la codifica è firmata in modo da poter riconoscere i ack genuini
 - lo schema sarebbe ovviamente facilmente attaccabile senza questa contromisura
- il campo sequence number è di 32 bit
 - lo spazio è poco: deve contenere una firma + altro
 - la firma è quindi piuttosto debole
 - comunque un attacco (es. brute force) riesce a far passare solo una frazione minima di pacchetti
- nota che se l'attaccante risponde con un ack corretto, non può fare spoofing e si può applicare la contromisura delle white/black-list

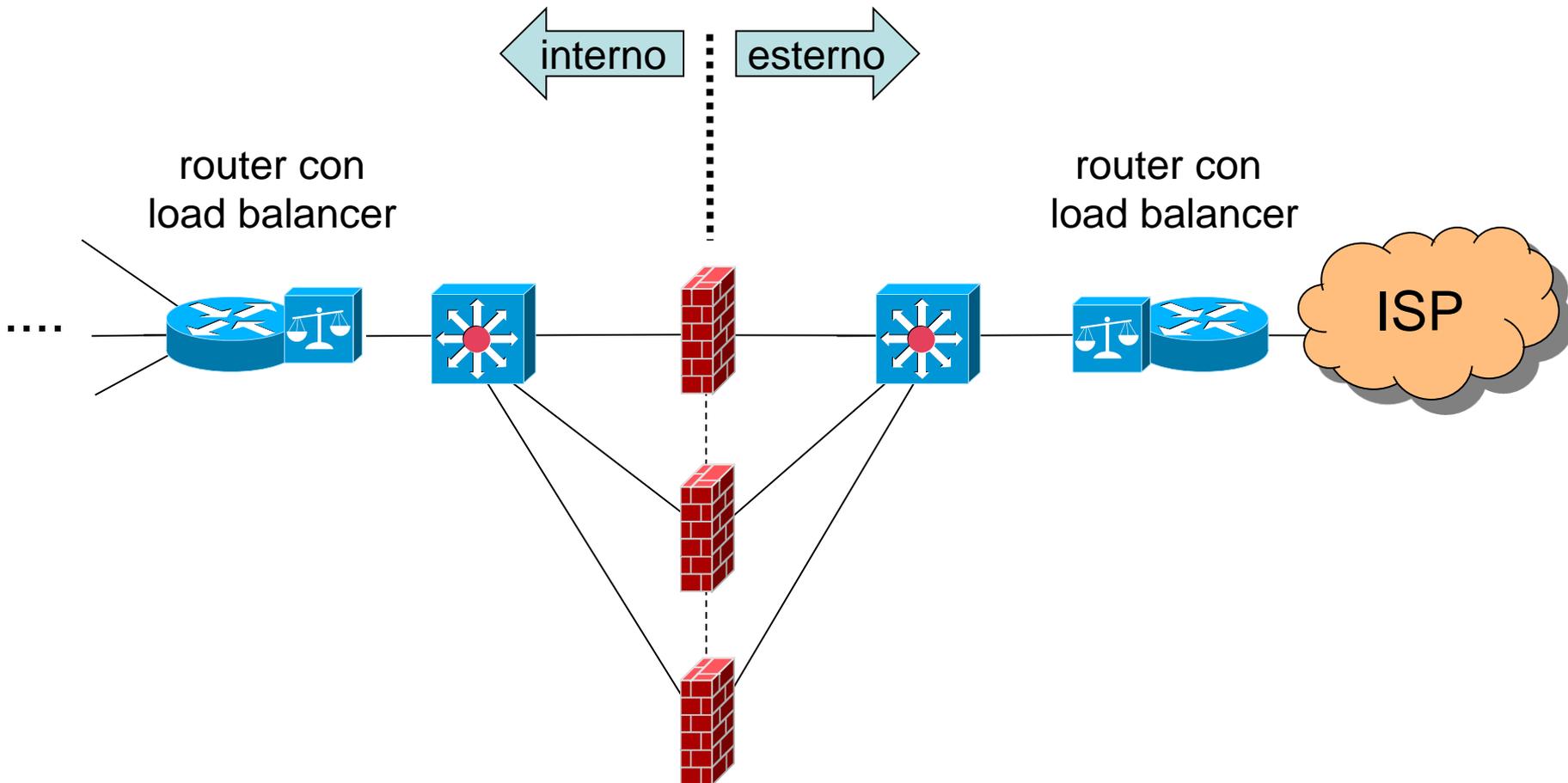
syn cookies

- limitato supporto a tcp options
- limitato supporto per maximum “segment size”

firewalls: prestazioni e affidabilità

- i fw spesso sono
 - un **collo di bottiglia** per le prestazioni
 - un **single point of failure**
- spesso utilizzati in configurazione “cluster” con tutti gli elementi attivi
 - **bilanciamento del carico**
 - **ridondanza**
- sono necessari load balancer e switch aggiuntivi

una architettura d'esempio



firewall e condivisione dello stato...

- se i load balancer inviano i pacchetti ad un firewall a caso...
- ... tutti i firewall devono vedere lo stesso stato delle connessioni!
- necessità di...
 - un collegamento tra i firewall
 - protocollo ad-hoc per la notifica delle nuove connessioni agli altri firewall
- si tratta di soluzioni proprietarie

architetture senza condivisione di stato

- strategia alternativa: i pacchetti di una stessa connessione sono sempre trattati da uno stesso firewall
- il load balancer non può fare scelte casuali
- ciascun fw deve vedere il traffico di una connessione in entrambe le direzioni
- in qualche modo i load balancer devono fare scelte coordinate

architetture senza condivisione di stato

coordinamento tra i load balancer

prima possibilità

- ciascun load balancer si ricorda il firewall assegnato alla connessione
 - ... e usa tale fw per i prossimi pacchetti
- richiede memoria lineare con il numero di connessioni
 - poco male la ram costa poco
- richiede di accedere alla ram at wire speed!
 - ...e la cache costa tanto!

architetture senza condivisione di stato

coordinamento tra i load balancer

seconda possibilità

- i load balancer adottano lo stesso algoritmo di scelta, tale che...
 - dipende solo dal pacchetto (es. da src, dest)
 - deterministico
- ...ma usato in maniera «speculare»
 - LB esterno: $fw = \text{hash}(\text{src_ip}, \text{dest_ip})$
 - LB interno: $fw = \text{hash}(\text{dest_ip}, \text{src_ip})$
 - infatti i pacchetti di ritorno hanno i src e dest invertiti
- no ram, no cache, solo cpu!
 - asic

architetture senza condivisione di stato

coordinamento tra i load balancer

caveat

- il load balancer assegna l'intera connessione ad un fw, ma...
- ...alcuni protocolli coinvolgono più sessioni tcp
 - es. ftp

configurazioni high-availability (HA)

- quanto down-time siamo disposti a sopportare?
 - dipende dal business (service level agreement, penali, ecc.)
- le configurazioni tolleranti ai guasti per mezzo di elementi ridondati si dicono in “fail over” o “high-availability” (HA)
- i single point of failure vanno evitati
 - altrimenti esiste un fault per cui si ha down immediato del sistema
 - vogliamo invece avere del tempo per permettere ad un tecnico di ripristinare lo stato iniziale senza che ci sia un down
 - un limitato degrado del servizio potrebbe essere ammesso nel frattempo (dipende dallo SLA)

HA e analisi del rischio

l'high-availability è necessaria quando...

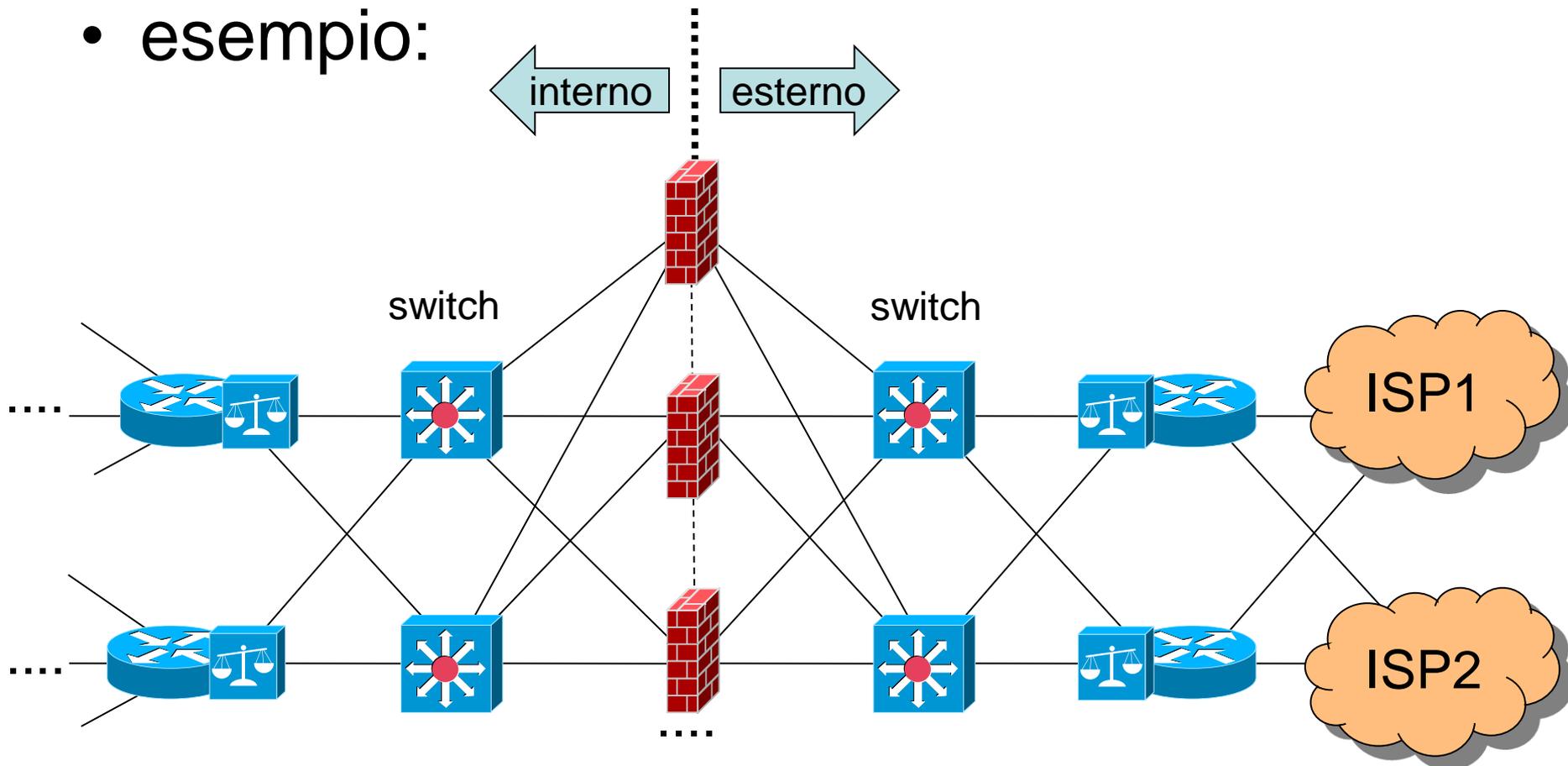
- gli SLA non prevedono il tempo per la sostituzione di uno degli elementi prima che scatti la penale
 - cioè esiste un rischio di pagare penali
- la penale (impatto) è sufficientemente alta da essere un rischio da mitigare
 - è possibile calcolare la spesa attesa per penali se riusciamo a stimare la probabilità di fault
 - i clienti che richiedono la penale potrebbero essere molti (es. hosting)
- il costo della configurazione HA dovrebbe essere inferiore al costo atteso derivante dal rischio di pagare penali

HA: componenti di contorno

- in un cluster di fw, load balancer e switch sono dei «single point of failure»
 - per una full high-availability bisogna ridondare tutto!
- le configurazioni HA sono più complesse e più costose
 - capex (capital expenses)
 - apparati ridondati
 - deployment più complesso
 - sistema di alert sui fault
 - opex (operating expense)
 - personale (reperibilità, skill specifici)
 - più apparati da mantenere
 - più spazio da affittare nel datacenter

full high-availability

- ciascun elemento dell'architettura deve essere ridondato
- esempio:



full high-availability

- in caso di guasto di un singolo elemento la rete deve riconfigurare automaticamente
- OSPF ha tempi di convergenza molto rapidi
 - deve girare sui load balancer e sui firewall
 - convergenza in ~200ms (fonte cisco)
- soluzioni di livello 2 (802.1w rapid spanning tree) sono più lente
 - ~1s (fonte cisco)

(intermezzo: gw ridondato)

- principio: no single point of failure in lan
- soluzioni tipiche per il default gateway di una lan
 - quando si rompe corri ad accendere il fail-over (!)
 - fai partecipare ciascuna macchina della lan al protocollo di routing (!!!!)
 - Virtual Routing Redoundancy Protocol, rfc 5798
- VRRP
 - un indirizzo ip *A* per un «router virtuale»
 - il router virtuale è composto da, almeno, due router fisici
 - solo uno dei due risponde all'arp request per *A* : il *master*
 - il master manda un heartbeat periodico agli altri router
 - quando il master muore, uno degli altri prende il suo posto
 - c'è un meccanismo di elezione come per il root bridge nello spanning tree degli switch
- HSRP: analogo, proprietario cisco

linux netfilter

- sistema di moduli Linux che realizza un fw
 - packet filter
 - stateful
 - nat
 - sistema di access list (chains) organizzate per funzionalità (tables)
 - tables
 - filter
 - chains: INPUT, OUTPUT, FORWARD
 - nat
 - chains: PREROUTING, POSTROUTING, OUTPUT
 - si possono definire delle “user-defined chain”
 - ciascuna tabella viene gestita da un modulo del kernel

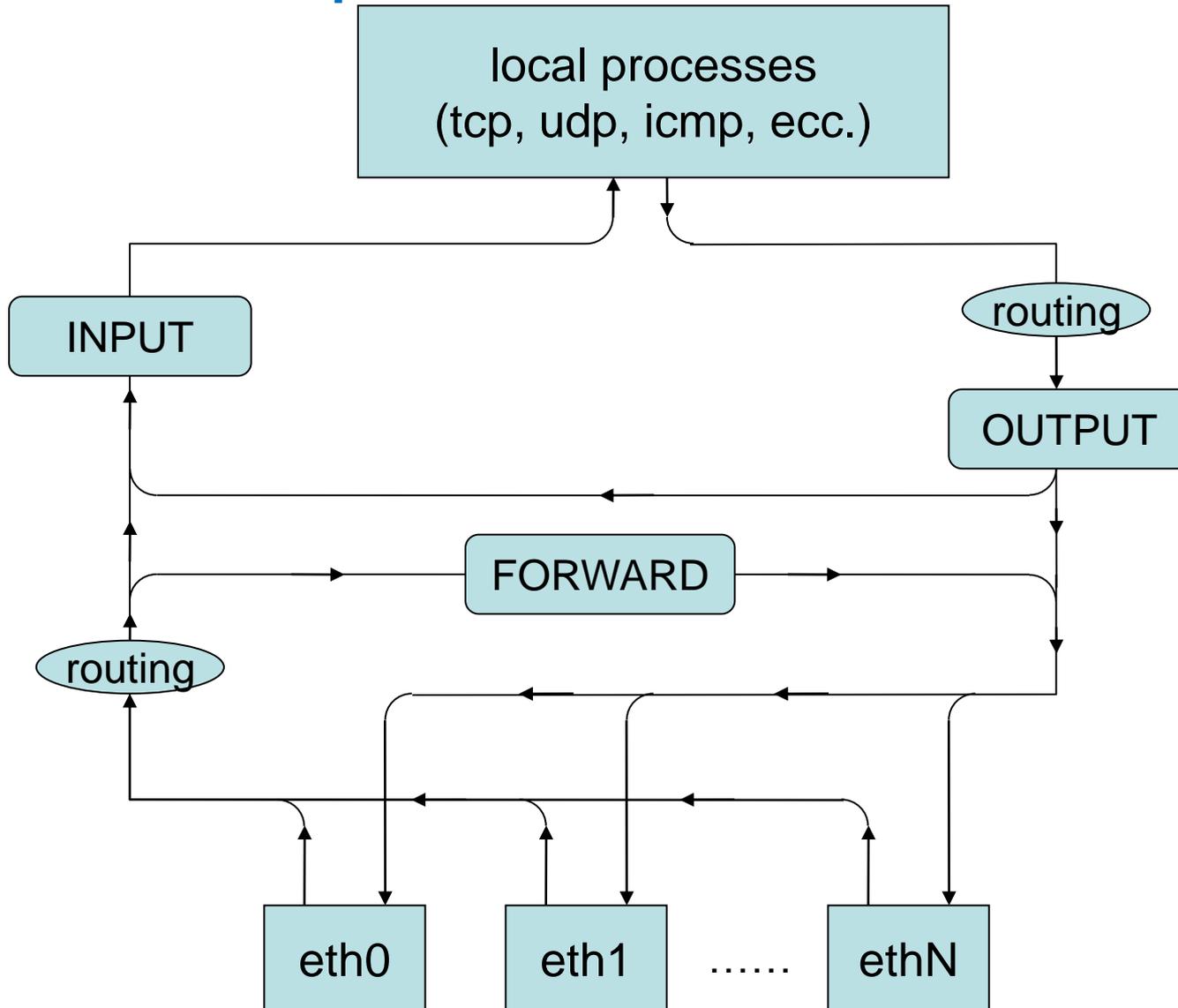
linux netfilter: chains (acl)

- ciascuna chain ha una sequenza di regole e un target
 - il target specifica cosa fare del pacchetto se questo non ha attivato alcuna regola
- target possibili
 - ACCEPT (pacchetto ok, lascia passare)
 - DROP (pacchetto droppato)
 - REJECT (come drop ma invia un icmp di errore al mittente)
 - <chain name> (salta alla chain specificata, come una chiamata a procedura, utile con “user-defined chain”)
 - RETURN (ritorna alla chain chiamante)
 - ...

linux netfilter: firewall rules

- una regola ha dei parametri e un target
 - i parametri specificano per quali pacchetti la regola viene attivata
 - il target specifica cosa fare
- parametri
 - protocollo (-p ip/tcp/udp/)
 - source/destination address (-s/-d [!]x.x.x.x/x)
 - source/destination port (--sport/--dport port)
 - input/output interface (-i/-o ethN)
 - tcp flags syn=1 e ack=0 (--syn)
 - e altri

chains e flusso pacchetti per la «filter table»



linux netfilter: connection tracking

- netfilter tiene traccia delle “connessioni”
 - per tcp e udp: coppia (non ordinata) di coppie {<addr, port>, <addr, port>}
 - per icmp echo: coppia di indirizzi {addr, addr}
- a ciascun pacchetto è associata una connessione (eventualmente ne viene creata una nuova)
 - questo viene fatto appena un pacchetto entra in netfilter da una interfaccia o dai processi locali
 - le connessioni muoiono per timeout o per protocollo (fin/ack, rst)
- stato del pacchetto
 - NEW: primo pacchetto della connessione
 - ESTABLISHED: pacchetto di una connessione già esistente
 - RELATED: nuova connessione associata ad una precedente (ftp, icmp errors)
 - INVALID: impossibile associare una connessione (es. icmp error non associati ad alcuna connessione)

linux netfilter: extended matches

- l'opzione `-m <nome>` permette di attivare molti altri tipi di regole
 - **state**: filtro in base allo stato (NEW, ESTABLISHED, RELATED, INVALID)
 - **conntrack** è una versione estesa
 - **mac**: filtro in base agli indirizzi mac di livello 2
 - **limit**: filtro in base alla frequenza di arrivo
 - anti DoS, anti scanning ecc.
 - **iprange**: es. 192.168.1.13-192.168.2.19
 - e molti altri

netfilter: comandi

- iptables
 - modifica e mostra la configurazione
 - -L mostra la configurazione
 - --flush cancella la configurazione
 - -A <chain> aggiunge in coda a <chain> un regola
 - ecc.
- iptables-save
 - dump della configurazione attuale
- iptables-restore
 - carica la configurazione da un dump fatto con iptables-save (più efficiente che molte chiamate a iptables)

esempio: nessun filtro

- `iptables --flush`
- `iptables-save`

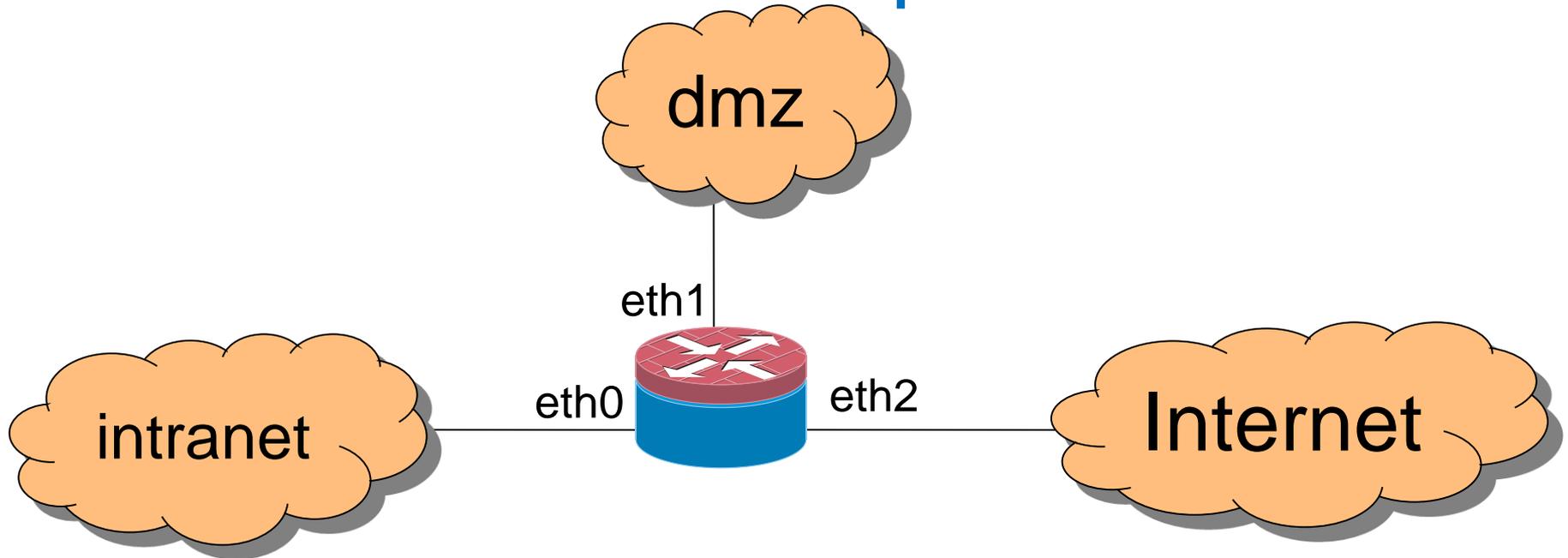
```
# Generated by iptables-save v1.3.3 on Fri Dec  8 17:58:19 2006
*filter
:INPUT ACCEPT [37:4620]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [19:1352]
COMMIT
# Completed on Fri Dec  8 17:58:19 2006
```

esempio: un semplice «personal firewall»

- `iptables -P INPUT DROP`
- `iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT`

```
# Generated by iptables-save v1.3.3 on Fri Dec 8 18:26:51 2006
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [14735:2397896]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Fri Dec 8 18:26:51 2006
```

implementazione della politica dmz di esempio



```
*filter
:INPUT ACCEPT [64:3448]
:FORWARD DROP [13:858]
:OUTPUT ACCEPT [409:37987]
-A FORWARD -i eth0 -o eth1 -m state --state NEW -j ACCEPT
-A FORWARD -i eth2 -o eth1 -m state --state NEW -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

proxy applicativi

- due sessioni di trasporto
 - S ↔ Proxy ↔ D
- permettono verifiche di sicurezza a livello applicativo
 - devono conoscere il protocollo applicativo o almeno parte di esso
- **spesso usati per prevenire comportamenti insicuri degli utenti della intranet**
- protocolli
 - http, https
 - smtp, pop3, imap
 - ftp, dns, sql
- circuit level proxies (SOCKS)
 - proxy generico, indipendente dal protocollo applicativo
 - utile per forzare una autenticazione a livello tcp

web proxy: obiettivi

- **autenticazione, controllo di accesso al servizio**
- **test per codice malevolo**
- **anonimità**
- **prestazioni: caching**
- **modifica contenuto**
 - censura
 - riformattazione per schermi piccoli (cellulari, PDA)
- **reverse proxy: un proxy in prossimità di un web server che riceve le richieste da Internet**
 - sicurezza (bastion host), load balancing, encryption acceleration, cache del contenuto statico
- **open source: squid, privoxy, spikeproxy**

transparent web proxy

- un web proxy regolare richiede la configurazione del browser
 - scomodo
 - la presenza del proxy è palese
- un proxy “transparent” intercetta richieste http qualsiasi
 - non richiede configurazione del browser
 - il proxy c'è ma non si vede
- esercizio: configurare linux netfilter per un transparent proxy

web proxy: vulnerabilità

- il protocollo http prevede il metodo CONNECT
- tale metodo viene interpretato dai proxy come un comando per lasciare passare connessioni arbitrarie
 - usato per connessioni http criptate
 - https (http su secure socket layer)
 - se connessione è criptata il proxy non può conoscerne il contenuto
- disabilitare CONNECT comporta l'impossibilità di usare https

network based IDS (NIDS)

- verifica la presenza sulla rete di traffico riconducibile ad attività sospette
- elementi
 - sniffer
 - particolari precauzioni per reti switched
 - se rule-based
 - db delle regole
 - detection engine
 - se anomaly-based
 - rappresentazione della conoscenza
 - machine learning
 - detection engine

NIDS: problemi

i NIDS hanno tutti i problemi visti quando abbiamo parlato di rilevamento automatico dei problemi

- falsi positivi e negativi
- difficili da comparare
- rule-based vs. anomaly-based
- tuning difficile

un NIDS molto famoso

snort

- <http://www.snort.org/>
- rule-based
 - regole pubbliche aggiornate regolarmente
- open source
- supportato sia sotto Windows che sotto Linux

NIDS e reti switched

- il nids è innanzi tutto uno sniffer
- sniffare reti switched è complesso
- supporto offerto dallo switch: mirroring verso una «destination port»
 - port mirroring
 - multi-port mirroring
 - vlan mirroring
- caveat
 - la banda della destination port può essere insufficiente anche se lo switch è full speed

NIDS: azioni

- può generare un log delle attività sospette
- può riconfigurare automaticamente un firewall
 - attività di contrasto automatica
 - detti anche intrusion prevention systems (IPS)
 - un NIDS–IPS è tipicamente in configurazione in-line con firewall incorporato
 - in sostanza un firewall con due interfacce che analizza il traffico da cui è attraversato

NIDS e connessioni tcp

- un NIDS non si può limitare a verificare ciascun pacchetto ip
 - cioè deve essere stateful!
- gran parte degli attacchi sono a livello applicativo e incapsulati in tcp
- è necessario seguire la sessione tcp e applicare le regole allo stream di bytes risultante
 - riordino dei segmenti tcp (e dei frammenti ipv4)
 - molto oneroso! richiede allocazione di buffer per ciascuna connessione

NIDS: scalabilità

- un NIDS dovrebbe essere in grado di elaborare tutto il traffico della rete
 - ogni pacchetto un lookup nel db
 - db in memoria
 - **perdita di pacchetti → perdita di accuratezza**
 - **falsi negativi**
 - le risorse necessarie dipendono da
 - numero di pacchetti
 - quantità di flussi tcp
 - quantità di segmenti fuori sequenza e uso della frammentazione

NIDS: load balancing

- è possibile fare cluster di NIDS con bilanciamento di carico
- **flusso**: pacchetti con la stessa quadrupla <saddr, sport, daddr, dport>
 - altra definizione di flusso possibile: stessa coppia <saddr, daddr>
- è necessario che ciascuna connessione venga analizzata dallo stesso NIDS
 - **una connessione è fatta di due flussi**
- i pacchetti di un flusso devono essere ispezionati dallo stesso NIDS nel cluster

switches e link aggregation (lag)

- gli switch permettono di aggregare più link in un solo link logico (lag)
 - utile per utilizzare due link paralleli senza spanning tree
 - autoconfigurato con il Link Aggregation Control Protocol, ieee 802.3ad
- bilanciamento del traffico sui vari link
 - ciascun flusso sullo stesso link
 - fondamentale per non desequenziare i pacchetti di un flusso – tcp inefficiente al risequenziamento massivo
 - tecnica dell'hash: $\text{link} = \text{hash}(\text{src}, \text{dst})$

NIDS load balancing = $\frac{1}{2}$ lag

- il lag è spesso configurabile anche staticamente
 - noi lo useremo con un solo switch!
- dietro a ciascuna porta del lag un NIDS
- mirroring di VLAN su un lag
- i lag mandano flussi identici su porte identiche e quindi sugli stessi NIDS.

HIDS vs. NIDS

- controllano aspetti diversi
 - NIDS:
 - legittimità del traffico nella rete
 - host IDS (HIDS)
 - legittimità del comportamento del software (integrità del sistema)
 - legittimità del traffico di rete da/per un host specifico
- approcci complementari
 - vedi principio «defence in depth»