

sicurezza di rete a livello applicativo

proxy applicativi

- due sessioni di trasporto
 - $S \leftrightarrow \text{Proxy} \leftrightarrow D$
- 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, apache**

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

web proxy e SSL/TLS

- nelle grandi organizzazioni è necessario avere supportare sia SSL/TLS sia una verifica a livello applicativo
- soluzione: le macchine degli utenti hanno un certificato dell'organizzazione installato
 - gli utenti non hanno il controllo amministrativo delle proprie macchine
- il web proxy può fare MitM su session SSL/TLS perché ha la corrispondente chiave privata.

Network Intrusion Detection Systems (NIDS)

Network IDS (NIDS)

- verifica la presenza sulla rete di traffico riconducibile ad attività sospette
- elementi
 - sniffer
 - particolari precauzioni per reti switched
 - database di regole/anomalie
 - rule-based: db delle regole
 - anomaly-based: conoscenza rappresentata e aggiornata con tecniche di machine learning
 - detection engine
 - rule-based: ricerca efficiente nel db delle regole
 - anomaly-based: tecniche di machine learning

NIDS: problemi

presentano le difficoltà tipiche del rilevamento automatico dei problemi

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

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 deve ispezionare il traffico (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»