Intrusion Detection Systems

Intrusione, controllo e contenimento nel digital warfare

Stefano Zanero – Information Security Analyst – Webbit'02 Padova, 06/07/02

I tre obiettivi della sicurezza

- *Confidenzialità:* solo le persone autorizzate possono accedere al sistema informativo
- *Integrità:* solo persone autorizzate possono modificare componenti del sistema, e solo nelle modalità per cui sono state autorizzate a procedere
- *Disponibilità:* il sistema deve fornire i servizi richiesti in un tempo "ragionevole" secondo i requisiti
- Obiettivi addizionali (casi specifici dei tre sopra):
  - Non-repudiation: un messaggio spedito deve poter essere "provato" e non poter essere negato
  - Privacy: forma particolare di confidenzialità che mira a proteggere uno specifico utente.
  - Identificabilità degli utenti
  - Accountability (logging)

Implementazione Classica

- *Confidenzialità – Integrità – Disponibilità* sono definite da una relazione molti-a-molti tra:
  - UTENTI opportunamente identificati
  - RISORSE cui possono accedere
- Meccanismi di *identificazione all'accesso*, utilizzo di *permessi*, dispositivi di *auditing* per verificare il buon funzionamento del tutto
- Meccanismi pervasivi di controllo dei permessi a runtime (esempio: processi in ambiente UNIX)

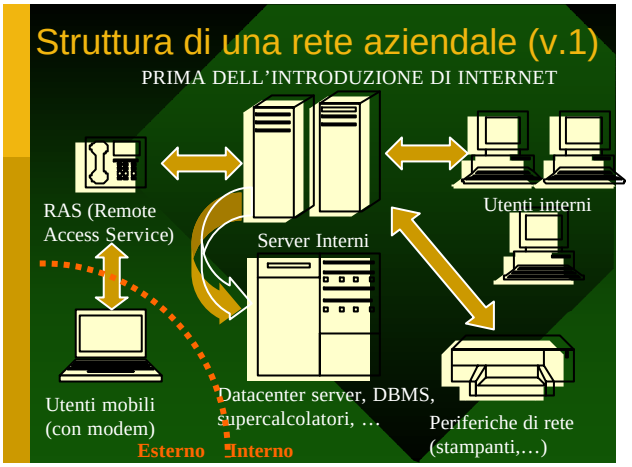
Un problema difficile

- Purtroppo il parallelo con la logica regge fino a un limite, ovvero fino al limite in cui i programmi si comportano conformemente alla specifica
- Per il teorema di Rice non è possibile "provare" in modo automatico la correttezza di un programma (in particolare rispetto ad ingressi eccezionali)
- La verifica manuale dei programmi e la loro scrittura in modo corretto (mediante corretti procedimenti di software engineering) teoricamente funziona ma nella pratica ha dimostrato dei limiti
- Esistono numerosi tipi di "exploit" per sfruttare vulnerabilità del software.

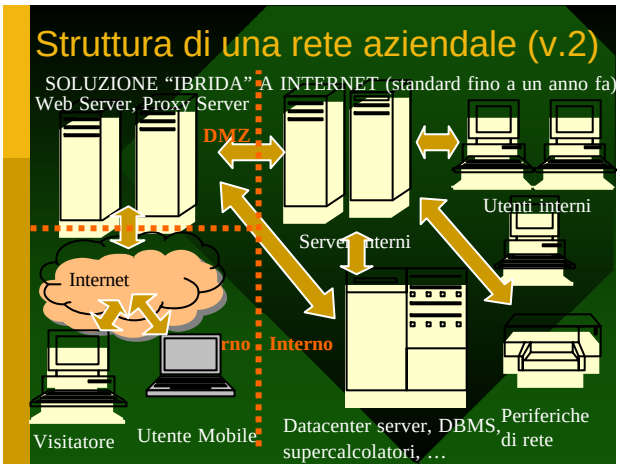
Tipico meccanismo di “exploit”

- Non è possibile “accedere” al sistema se non fornendo login e password. Ma è vero ?
- Ricordiamoci che i “demoni” offrono servizi di rete, a volte anche ad utenti non autenticati.
- Ricordiamoci che i “demoni” usano per le restrizioni il SUID, che a volte è “root”. Quindi un demone può avere un accesso non ristretto al sistema.
- Se si riesce a “imbrogliare” un demone e a fargli eseguire comandi al posto nostro, possiamo sfruttare il suo SUID e violare le restrizioni d’accesso.
  - Esempi: crash sulle boundary condition, buffer overflow, fallimento nell’handling di input eccezionali, …
- Questo è un tipico meccanismo di EXPLOIT

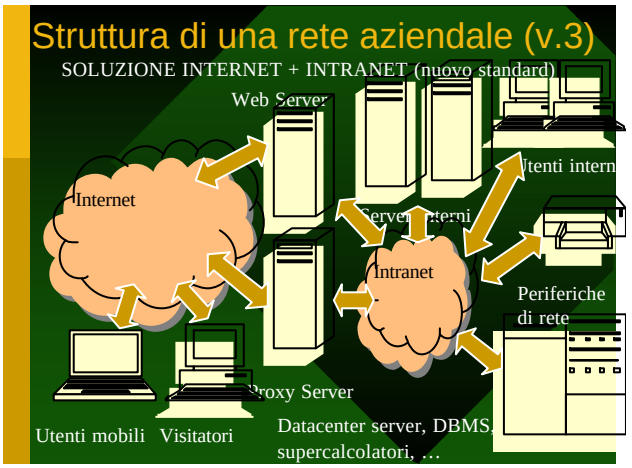
Struttura di una rete aziendale (v.1)



Struttura di una rete aziendale (v.2)



Struttura di una rete aziendale (v.3)



Evoluzione o involuzione ?

- Schema di sicurezza basato sul solito paradigma: identificazione degli utenti, e dei loro privilegi
- Una volta bastava un RAS; oggi punti di accesso multipli, e sostanziale mancanza di distinzione tra interno ed esterno. Soluzioni: Firewall, che sono comunque limitati.
- Gestione dei privilegi d'accesso: molto semplice "sistema per sistema", difficile attraverso la rete. Soluzioni: directory, Kerberos, …
- È necessario garantire l'accesso di utenti non autenticati ad alcuni servizi (es. servizio web e FTP). Soluzioni: in realtà, nessuna.
- I pacchetti TCP/IP non sono autenticati e la comunicazione su Internet non è riservata. Soluzioni: IPSec, VPN…

Forza e debolezza: il punto chiave

- Continuando a usare il paradigma classico "Who are you? What can you do?" complichiamo terribilmente le cose: una volta era il login/password, ora sono intrecci di criptografie.
- Il principio dell'identificazione e dell'associazione ai permessi è ancora fondamentale ma non "scala" facilmente alle dimensioni di una WAN, o dell'Internet
- Gli hacker non utilizzano la forza, ma sfruttano le debolezze intrinseche dei sistemi. Il paradigma classico su scala di rete è insicuro, ma non può essere "aggiornato".
- Logica KISS: Keep It Simple, Stupid.
- Bisogna trovare un nuovo paradigma che complementi quello classico, ovviando alle sue debolezze

Why are you doing this ?

- Torniamo alle origini: *confidenzialità, integrità, disponibilità* hanno un comune denominatore
- Il sistema informatico ha uno *scopo,* e deve servire a quello scopo evitando compromissioni
- Ogni violazione del paradigma CID è visibile perché il sistema compie azioni "anomale"
- Invece di limitarci a chiedere "Who are you ? What can you do ?" cerchiamo di capire: "Why are you doing this?"
- INTRUSION DETECTION SYSTEM, rivelatore di intrusione

Intrusion Detection System

- Un IDS non si sostituisce ai normali controlli, ma piuttosto cerca di scoprire i loro fallimenti
- Chi entra in un sistema informatico abusivamente compie alcuni tipi di azione che un utente normale non farebbe mai; identificando queste azioni "anomale" possiamo scoprire un intruso
- Due metodi principali per farlo:
  - Anomaly Detection: determinare statisticamente modelli di "comportamento normale", e segnalare eventuali deviazioni "significative"; conoscenza *a posteriori*
  - Misuse Detection: confrontare gli eventi con "schemi" predefiniti di attacchi; conoscenza *a priori*

Anomaly Detection Model

- Determiniamo statisticamente dei modelli di comportamento “normale” dell’utente, e segnaliamo deviazioni significative
- Generico: ha in generale il vantaggio di potersi adattare a “nuovi” schemi d’attacco
- Varie tecniche e approcci: Autoclass (Bayes); identificazione predittiva; uso di reti neurali; model-based detection.
- Problemi principali: scelta delle metriche (cosa misurare) e dei threshold (soglia d’allarme); scelta dei modelli di base

Misuse Detection Model

- Il sistema cerca di rappresentare i “misuse case”, ovvero gli eventi che costituiscono una intrusione
- Le tecnologie:
  - Utilizzo di linguaggi basati su regole e sistemi esperti
    - Problema della sequenzialità
    - Problemi nella uncertain-reasoning
  - State–transition analysis (es. in STAT); ancora rule-based ma approccio diverso (analisi su grafo con ordinamento parziale)
  - Utilizzo del pattern-matching (es. in IDIOT) con algoritmi Discrete Approximate Matching – Longest Common Subsequence
    - Problema di interleaving degli eventi
- Molti sistemi IDS combinano feature di anomaly e di misuse detection

On-line vs. off-line operations

- On-line operations: il sistema lancia delle alert analizzando gli eventi correnti; normalmente usa una finestra di dati. Spesso per problemi di complessità computazionale questi alert sono limitati a regole attivate da trigger.
- Off-line (batch) operations: il sistema analizza i log (registrati) degli eventi. Può generalmente utilizzare maggiore potenza di calcolo. Può analizzare finestre illimitate nel passato, anche l’intera sessione.
- Integrazione tra i due principi: ISOA; trigger per l’attivazione di alert on-line, l’analisi off-line viene utilizzata per completare le ricerche su eventi giudicati interessanti dall’operatore.

Network Based vs. Host Based

- HOST based: i primi IDS erano host-based; un IDS host-based si appoggia al sistema operativo e controlla le system call (esecuzione e controllo dei processi) e gli accessi (al sistema, ai device…)
- NETWORK based: controllano il traffico sulla rete cercando nel flusso di pacchetti le tracce di una intrusione
- Prossima frontiera: interoperabilità, correlazione, normalizzazione
- Entrambi possono essere Anomaly o Misuse based.

Misuse detection: pessima idea?

- I sistemi di misuse detection hanno molti problemi ma ne presentano uno in particolare: la necessità di gestire una knowledge base degli attacchi
- Problemi di aggiornamento (solo gli attacchi conosciuti vengono segnalati) e di ingegnerizzazione delle signature (in qualsiasi modo vengano gestite...)
- Problema del polimorfismo negli attacchi: ADMutate, encoding UTF...

Nuove frontiere per gli IDS

- Verranno presentate ora le direzioni di ricerca in cui ci stiamo orientando

1. Utilizzo di tecniche proprie della behavior engineering
2. Utilizzo di algoritmi basati su riconoscitori a rete neurale

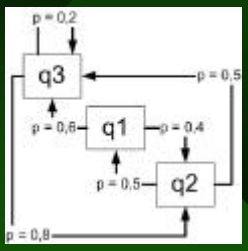
Behavior Engineering

- Campo nato dalle ricerche per lo sviluppo di agenti intelligenti e autonomi
- Tradizionalmente il compito della behavior engineering è studiare i comportamenti per provare a riprodurne versioni “sintetiche”
- Abbiamo proposto l’uso inverso della behavior engineering nella classificazione dei comportamenti

Classificare i comportamenti

- Esistono e sono in corso ricerche per la classificazione dei comportamenti dell’utente di un sito web mediante un modello di Markov, a scopo predittivo
- Catena di Markov (del primo ordine): un insieme di n stati, per ognuno dei quali è definita la probabilità di passaggio a un altro stato all’istante successivo (una matrice n x n di probabilità)
- Modello di Markov di ordine n: dipendente da n istanti precedenti (matematicamente molto più complesso)
- Modello di Markov nascosto: filtrato da una funzione di uscita

Catena di Markov: esempio



	q1	q2	q3
q1	0.0	0.4	0.6
q2	0.5	0.0	0.5
q3	0.8	0.0	0.2

Etologia (?!?)

- Studio e classificazione dei comportamenti degli animali
- FAP: Fixed Action Patterns, sequenza di azioni atomica e non interrompibile
- MAP: Modal Action Pattern, una variante dei FAP proposta da Barlow
- Una intrusione è modellabile con un MAP, a grandi linee
- Il comportamento di un animale può venire modellato tramite un etogramma (elenco di comportamenti tipici) su cui si può costruire un modello di Markov

Modelli e riconoscimento

- Il modello di Markov del comportamento di un animale si modifica sensibilmente se sta svolgendo un MAP o se è in una particolare condizione psicofisica (es. fight or flight)
- Dal comportamento si può (con un test statistico e un determinato indice di confidenza) determinare la probabilità che appartenga a un determinato modello, e quindi che l'animale sia in una determinata condizione
- Modelli di Markov nascosti !!!

IDS Behaviorale

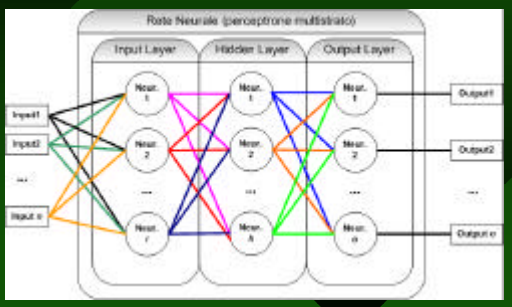
- Dal comportamento degli utenti costruiamo un modello markoviano (di ordine 1 ? Di ordine n ?)
- Cerchiamo di capire se il comportamento "corrisponde" al modello "utente normale", al modello "utente che cerca di fare qualcosa che non va", o al modello "ehi, questo non è un mio utente…"
- Algoritmo di forward-backward, altrimenti noto come "algoritmo di Viterbi": individuazione di modelli di Markov nascosti

Neural Networks Demistified

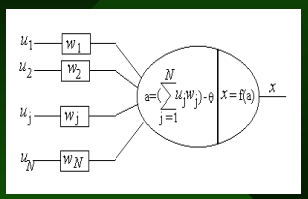
- Cos'è una NN ?
- Un algoritmo capace di:
  - interpolare funzioni
  - classificare oggetti

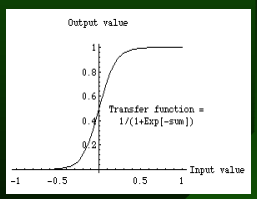
  basandosi su esempi
- Cosa non è una NN?
- Non è magia ma matematica
- Non è intelligenza artificiale
- Non riproduce il cervello o i neuroni

Come è fatta una rete neurale



Neurone Artificiale (zoom)





Il neurone somma gli input, moltiplicati ognuno per i pesi, e fornisce in output un valore filtrato dalla funzione a destra

Come si usa una rete neurale

- Si individua uno spazio di n variabili di ingresso (problema del "mapping")
- Si individuano n variabili di uscita (problema dei threshold)
- Si costruisce una rete (la struttura è un'arte, non c'è un metodo)
- Modelli base: percettrone multistrato, ma anche reti di Hopfield, Recurrent Networks, …
- La rete va addestrata, ovvero vanno selezionati opportunamente i pesi degli ingressi dei neuroni, affinché la rete fornisca l'uscita desiderata

Addestramento

- Calibrazione dei parametri affinché la rete neurale riproduca una particolare funzione tra gli ingressi e le uscite
- Questa funzione può anche non essere nota a priori
- La funzione può essere vista come un classificatore
- Teorema di approssimazione universale: un percettrone multistrato, in dipendenza dal numero dei neuroni, può approssimare qualsiasi funzione per quanto complessa con una precisione arbitraria

Addestramento supervisionato e non

- Supervisionato significa che esiste una conoscenza umana
- Fornisco alla rete neurale esempi di input ed output “corretti”
- Algoritmo di back propagation (numero di esempi almeno pari al doppio del numero dei parametri)
- Non sempre esiste o è formalizzabile una conoscenza umana
- Vorremmo che la rete individui “gruppi interessanti” nel dominio
- È possibile addestrare la rete “online”, durante le operazioni; ad ogni modo “addestramento” in questo caso ha un significato molto diverso

Caratteristiche “desiderate”

- Robustezza e generalizzazione: un modello a rete neurale può approssimare il concetto di “simile”, purché vi sia un mapping appropriato
- Adattamento: la precisione della rete può migliorare con l’uso, se i dati vengono riutilizzati per l’addestramento
- Outlyer Detection: individuazione del concetto di “strano” all’interno dei fenomeni

Problemi, problemi…

- Una rete neurale è prona al fenomeno dei “false positives”, ciò nella intrusion detection è inaccettabile
- Una rete neurale non genera conoscenza umanamente comprensibile come risultato del training
- Problema di mapping: quali dati considerare ? In che formato ? Il numero di input di una rete neurale è fisiologicamente limitato

Alcuni spunti per la ricerca

- Supervised training: tentativo di riprodurre mediante una NN un sistema di tipo misuse detection: sembra inefficiente
- Unsupervised training: sembra promettente, tuttavia:
  - problema di definire il concetto di outlyer e delle soglie appropriate.
  - Problema di non reattività
  - Problema principale: mai implementato realmente
- Problemi computazionali per il throughput e per l'addestramento !

Questions ? :-)

Contatti e riferimenti

- E-Mail: stefano.zanero@ieee.org
- Slides disponibili su www.sikurezza.org
- "Diario di un Security Manager", rubrica settimanale sulla sicurezza informatica

www.cwi.it

COMPUTERWORLD IDG