Intrusion Detection Systems

Introduzione e concetti di base

Ing. Stefano Zanero – Politecnico di Milano – Webb.IT '03, Padova, 11/05/03

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

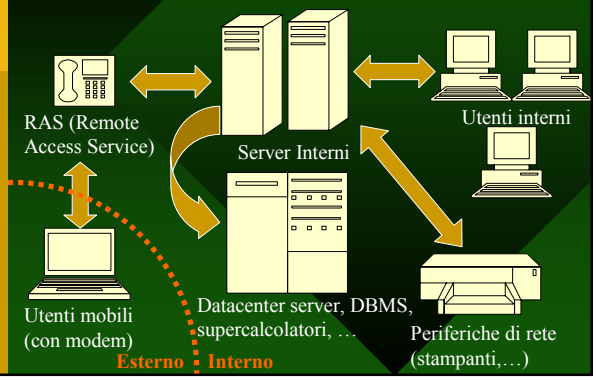
- Purtroppo il parallelo con la logica regge fino a un limite, ovvero fino al limite in cui i programmi si comportano conformemente alla specifica
- Per il teorema di Rice non è possibile "provare" in modo automatico la correttezza di un programma (in particolare rispetto ad ingressi eccezionali)
- La 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"

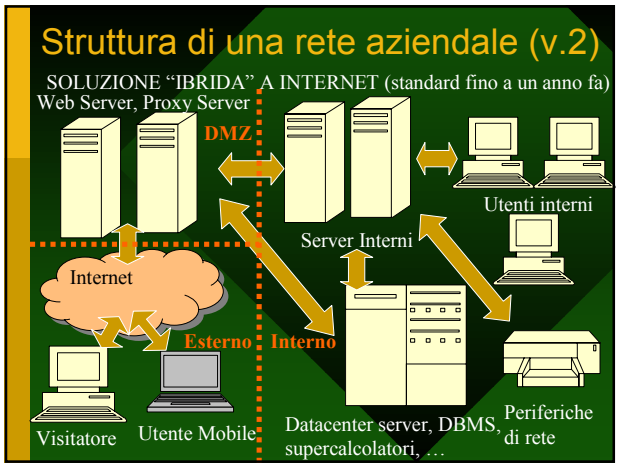
- Non è possibile "accedere" al sistema se non fornendo login e password. Ma è vero ?
- Ricordiamoci che i "demoni" offrono servizi di rete, a volte anche ad utenti non autenticati.
- Ricordiamoci che i "demoni" usano per le restrizioni il SUID, che a volte è "root". Quindi un demone può avere un accesso 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)

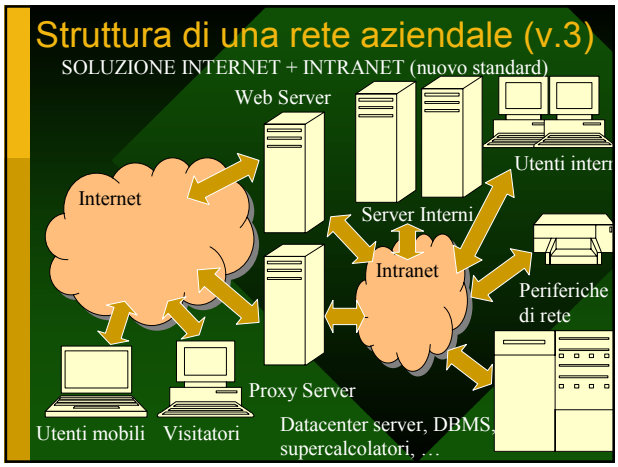
PRIMA DELL'INTRODUZIONE DI INTERNET



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*

Tassonomia degli IDS (1)

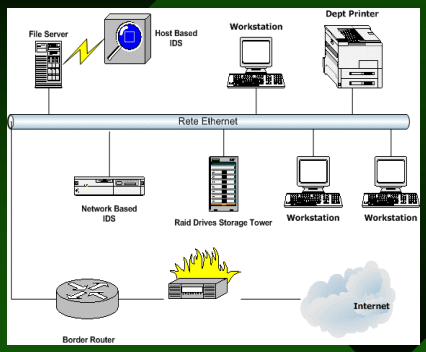
Anomaly Detection Model

- Descrivere comportamento normale e segnalare deviazioni
- In teoria, può riconoscere ogni attacco
- Dipende fortemente dal modello, dalle metriche e dalla scelta dei threshold
- Le sue segnalazioni sono di tipo statistico

Misuse Detection Model

- Descrivere i vari tipi di attacco informatico e riconoscerli
- Riconosce solo gli attacchi per cui esiste una firma
- Il modello di regole usate per esprimere gli attacchi può avere problemi di espressività
- Le segnalazioni sono molto precise e possono essere usate per risposte attive

Tassonomia degli IDS (2)



- *Host Based*: opera su una singola macchina
- *Network Based* controlla il traffico sulla rete

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.

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...
- Inoltre: problema di uncertain reasoning e sequenzialità

Anomaly Detection: i problemi

- Scelta delle metriche (cosa misurare)
- Scelta dei threshold (soglia d'allarme) e delle funzioni
- Scelta dei modelli di base: cosa succede se l'attacco compare solo in variabili che non abbiamo modellato ?
- Segnalazione di tipo statistico che va interpretata da un esperto umano

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 sull'apprendimento non supervisionato

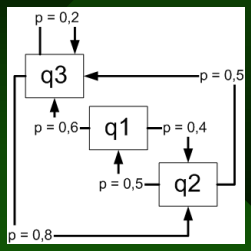
Behavior Engineering

- 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

Apprendimento supervisionato e non

- Supervisionato significa che esiste una conoscenza umana che l'algoritmo deve emulare
- Fornisco esempi di input ed output "corretti"
- Es.: rete neurale con algoritmo di back propagation
- Non sempre esiste o è formalizzabile una conoscenza umana
- Vorremmo che l'algoritmo individuasse "gruppi interessanti" nel dominio
- "Addestramento" in questo caso ha un significato evidentemente molto diverso !

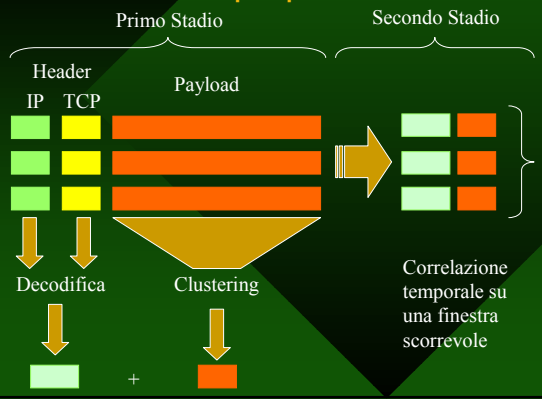
Le caratteristiche desiderate

- Quali caratteristiche degli algoritmi di apprendimento non supervisionato ci interessano ?
- Outlyer Detection: individuazione del concetto di "strano" all'interno dei fenomeni
- Robustezza e generalizzazione: un modello di apprendimento è più resistente al polimorfismo rispetto a un modello basato su regole
- Adattamento: un modello di apprendimento può migliorare progressivamente le proprie prestazioni e adattarsi allo specifico ambiente d'uso
- Non supervisionato: non vogliamo utilizzare conoscenza umana, ma vogliamo che sia l'algoritmo a determinare le "anomalie"

Un problema dimensionale

- Un IDS network based deve osservare una finestra di pacchetti per poter essere di qualche utilità
- I pacchetti TCP/IP hanno una lunghezza variabile tra i 20 e i 1500 byte (su una rete Ethernet)
- Non esistono algoritmi non supervisionati di apprendimento che possano analizzare dati a decine di migliaia di dimensioni: non potremmo mai mostrare a un algoritmo non supervisionato una sequenza di pacchetti di lunghezza significativa.
- **Molti approcci proposti in letteratura "risolvono" il problema scartando a priori il contenuto dei pacchetti**

Architettura proposta



Esempio di correlazione (1)

Denial-of-Service mediante globbing (codice ArachNIDS IDS487):

```
Variante 1:
4C 49 53 54 20 2A 2F 2E 2E 2F 2A 2F 2E 2E 2F 2A  LIST */../*/../*
2F 2E 2E 2F 2A 2F 2E 2E 2F 2A 2F 2E 2E 2F 2A 2F  /../*/../*/../*/
2E 2E 2F 2A 2F 2E 2E 2F 2A 2F 2E 2E 2F 2A 2F 2E  ../*/../*/../*/.
2E 2F 2A 2F 2E 2E 2F 2A 2F 2E 2E 2F 2A 2F 2E 2E  ./*/../*/../*/..
2F 2A 0D 0A                                      /*..

Variante 2:
4C 49 53 54 20 2A 2F 2E 2A 2F 2A 2F 2E 2A 2F 2A  LIST */.*/*/.*/*
2F 2E 2A 2F 2A 2F 2E 2A 2F 2A 2F 2E 2A 2F 2A 2F  /.*/*/.*/*/.*/*/
2E 2A 2F 2A 2F 2E 2A 2F 2A 2F 2E 2A 2F 2A 2F 2E  .*/*/.*/*/.*/*/.
2A 2F 2A 2F 2E 2A 2F 2A 2F 2E 2A 2F 2A 2F 2E 2A  */*/.*/*/.*/*/.*
2F 0D 0A                                         /..

Variante 3:
4C 49 53 54 20 2E 2A 2E 2F 2A 3F 2F 2E 2A 2E 2F  LIST .*./*?/.*./
2A 3F 2F 2E 2A 2E 2F 2A 3F 2F 2E 2A 2E 2F 2A 3F  *?/.*./*?/.*./*?
2F 2E 2A 2E 2F 2A 3F 2F 2E 2A 2E 2F 2A 3F 2F 2E  /.*./*?/.*./*?/.
2A 2E 2F 2A 3F 2F 2E 2A 2E 2F 2A 3F 2F 2E 2A 2E  *./*?/.*./*?/.*.
2F 2A 3F 2F 0D 0A                                /*?/..
```

Esempio di correlazione (2)

La rete S.O.M. classifica tutte e tre le varianti nel nodo 97, che normalmente non ospita traffico destinato alla porta 21.

Le firme per un IDS basato su signature, nell'intento di generalizzare, fanno un match solo su '/*'. In tale modo finiscono per scattare anche con traffico lecito, generando falsi positivi.

Ulteriori spunti per la ricerca

- Bisognerebbe studiare accuratamente, su un prototipo più avanzato, il problema di determinare le soglie di pericolo per ridurre i falsi positivi
- I nostri risultati preliminari indicano che il sistema non dovrebbe porre problemi di throughput, ma ciò andrebbe verificato approfonditamente.
- Il sistema non genera conoscenza umanamente comprensibile come risultato del training; bisognerebbe studiare possibili meccanismi di interazione correttiva da parte di un esperto, con un sistema semi-supervisionato di apprendimento

Questions ? :-)

Contatti e riferimenti

- E-Mail: zanero@elet.polimi.it
- Slides disponibili su www.elet.polimi.it/upload/zanero
- "Diario di un Security Manager", rubrica settimanale sulla sicurezza informatica

www.cwi.it

COMPUTERWORLD IDG INTERNATIONAL DATA GROUP