Intrusion Detection System e Sistemi Adattativi

Relazione per il corso di Sistemi Adattativi
2005/2006

Guido Vicino

Università del Piemonte Orientale
Amedeo Avogadro

Intrusion Detection System

- Che cos'è un Intrusion Detection System?
- Le componenti di un IDS:
  - Sensori
  - Motore
  - Console
- Risorse, modelli e tecniche

Intrusion Detection System (2)

- Reti o sistemi
  - Network Intrusion Detection System
  - Host based Intrusion Detection System
  - Hybrid Intrusion Detection System
- Sistemi di tipo
  - Passivi
  - Attivi

Intrusion Detection System (3)

- Come valutiamo un IDS?
  - Detection rate
  - False positive rate
- Rilevamento
  - Real Time
  - Offline (forensic analysis)
- Misuse versus Anomaly

Misuse versus Anomaly

- Due tecniche di rilevamento
  - Misuse detection
  - Anomaly detection
- Due sistemi di rilevamento
  - Signature based intrusion detection system
  - Anomaly detection system
- Vantaggi e svantaggi

Data Mining ed IDS

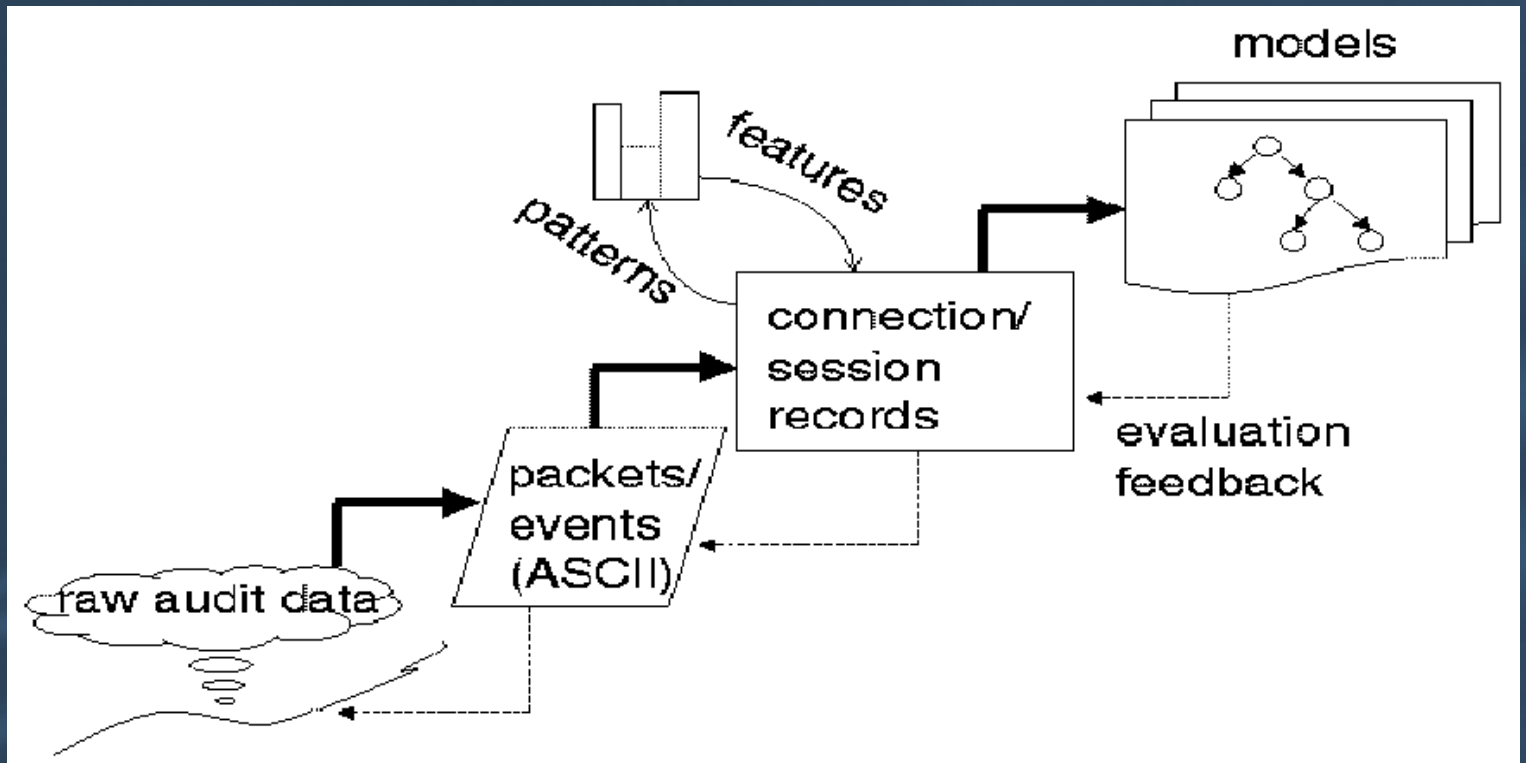


Figure 1: The Data Mining Process of Building ID Models

Data mining sui dati di Audit

- Tecniche utili
  - Algoritmi di classificazione
  - Analisi di relazioni e collegamenti
  - Analisi di sequenze
- Cosa si vuole costruire
  - Modello del comportamento “normale”
  - Episodi frequenti
  - Pattern d'intrusione

Regole di associazione

- Sia *A* un insieme di attributi, ed *I* un insieme di valori per *A* chiamati *item*. Ogni sotto insieme di *I* è chiamato un *item set*. Il numero di item in un item set definisce la *lunghezza* dello stesso. Sia *D* una base di dati con *n* attributi(o colonne).

$X \rightarrow Y$, *confidenza, supporto*

Regole di associazione (2)

- Esempio

ls → /home/john, 0.4, 0.2

cat → /etc/passwd, 0.01, 0.2

- Frequent Episode Rule

$X,Y \rightarrow Z,$

confidenza, supporto, finestra

Regole di associazione (3)

- Attributi essenziali
  - source address
  - destination address
- Attributi Axis
  - servizio offerto
  - flag della connessione
- Attributi di referenza
  - soggetto ed azione

Frequent Episode Rule

time	duration	service	src_host	dst_host	src_bytes	dst_bytes	flag	...
1.1	0	http	spoofed_1	victim	0	0	S0	...
1.1	0	http	spoofed_2	victim	0	0	S0	...
1.1	0	http	spoofed_3	victim	0	0	S0	...
1.1	0	http	spoofed_4	victim	0	0	S0	...
1.1	0	http	spoofed_5	victim	0	0	S0	...
...	...	...	...	...	...	...	...	...
10.1	2	ftp	A	B	200	300	SF	...
13.4	60	telnet	A	D	200	2100	SF	...
...	...	...	...	...	...	...	...	...

Table 1: Network Connection Records

Frequent episode	Meaning
(service = http, flag = S0), (service = http, flag = S0) → (service = http, flag = S0) [0.93, 0.03, 2]	93% of the time, after two *http* connections with *S0* flag are made (to a host *victim*), within 2 seconds from the first of these two, the third similar connection is made, and this pattern occurs in 3% of the data

Table 2: Example Frequent Episode Rule

Costruzione delle Feature

- Dalle *regole* di associazione vogliamo ricavare *feature*
- Dalle *feature* vogliamo costruire il *modello*
- Le *feature* devono possedere una *codifica* che le renda facilmente manipolabili

Codifica delle Feature

- Per associazioni simili per struttura si usino numeri vicini
- Si scelga un ordine con cui codificare ciascun attributo

es:

flag, axis, referenza, essenziali, altri

Codifica delle Feature (2)

association	encoding
$(flag = SF, service = http, src\_bytes = 200)$	11001
$(service = icmp\_echo, dst\_host = host_B)$	02100
$(flag = S0, service = http, src\_host = host_A)$	21010
$(service = user\_app, src\_host = host_A)$	03010
$(flag = SF, service = icmp\_echo, dst\_host = host_B, src\_host = host_C)$	12120
...	...

Table 3: Encodings of Associations

Codifica delle Feature (3)

- Per effettuare il confronto tra due episodi basta un semplice confronto *digit wise.*
- Il confronto digit wise ci dà un punteggio di differenza.
- Tramite il punteggo di differenza *diff* possiamo ricavare i pattern *intrusion only*.

Pattern Intrusion Only

- Come si calcolano
  - Si codificano tutti i pattern.
  - Si calcolano i punteggi di differenza tra i pattern del *data set* dell'intrusione e quello relativo al comportamento normale.
  - Il pattern con il più basso *diff score* sarà il punteggio d'intrusione per il pattern.
  - Si restituiscono come intrusivi tutti i pattern con *diff* diverso da *zero*.

Costruzione di Feature Aggiuntive

- Date le feature esistenti si vuole ricavarne di ulteriori
  - *count*
  - *percent*
  - *average*
- Esempio di feature aggiuntiva:

  “Il conteggio delle connessioni verso lo stesso *dst_host* negli ultimi due secondi, e fra queste la percentuale di quelle che hanno lo stesso servizio di quella corrente, e la percentuale di quelle che hanno la flag *S0.*”

Input: a frequent episode, and the set of existing features in connection records, $\mathcal{F}$
Output: the updated $\mathcal{F}$
Begin

(1) Let F_0 (e.g., *dst_host*) be the reference attribute used to mine the episode;
(2) Let w, in seconds, be the minimum width of the episode;
/* all the following features consider only the connections in past w
* seconds that share the same value in F_0 as the current connection
*/
(3) Let $count\_same_{F_0}$ be the number of these connections;
(4) $\mathcal{F} = \mathcal{F} \cup \{count\_same_{F_0}\}$;
(5) **for** each "essential attribute" F_1 other than F_0 **do begin**
(6) **if** the same F_1 value is in all the itemsets **then begin**
(7) Let $percent\_same_{F_1}$ be the percentage of connections that share the same F_1 value as the current connection;
(8) $\mathcal{F} = \mathcal{F} \cup \{percent\_same_{F_1}\}$;
end else
/* there are different F_1 or no F_1 values at all */
(9) Let $percent\_diff_{F_1}$ be the percentage of different F_1 values in the connections;
(10) $\mathcal{F} = \mathcal{F} \cup \{percent\_diff_{F_1}\}$;
end
end
(11) **for** each value V_2 of an "non-essential" attribute F_2 **do begin**
(12) **if** V_2 is in all the itemsets **then begin**
(13) Let $percent\_same_{V_2}$ be the percentage of connections that share the same V_2 value as the current connection;
(14) $\mathcal{F} = \mathcal{F} \cup \{percent\_same_{V_2}\}$;
(15) **end else if** F_2 is a numerical attribute **then begin**
(16) Let $average_{F_2}$ be the average of the F_2 values of the connections;
(17) $\mathcal{F} = \mathcal{F} \cup \{average_{F_2}\}$;
end
end
end

Figure 2: Constructing Features from Frequent Episode

Dalle feature alle Regole d'Intrusione

- Dopo aver ricavato le feature relative alle intrusioni, vogliamo costruire le regole per l'IDS
- Come procedere:
  - Definendole manualmente.
  - Utilizzarndo un sistema automatico ad esempio tramite un algoritmo di *Machine Learning*.

Creazione delle regole tramite RIPPER

- Si vogliono definire delle regole *if/then* relative al comportamento del sistema:

if for the past 2 seconds, the *count of connections to the same dst_host* is greater than 4;

and *the percentage of those that have the same service* is greater than 75%;
and *the percentage of those that have the "S0" flag* is greater than 75%;

then there is a *syn_flood* attack.

RIPPER

- *Input:* un vettore di caratteristiche basato sulle feature estratte
- *Overfit & Simplify:*
  - Si costruisce un albero che “*overfitta*”
  - Si converte l'albero in regole
  - Si eliminano le regole inutili una per una (*pruning*)
  - Si ordinano le regole per aumentare l'accuratezza
  - Si considera l'ordine durante la classificazione

RIPPER (2)

- Procedura per la costruzione di regole
  - Si divide (casualmente) il *training set* in *growing set* (2/3) and *pruning set* (1/3)
  - Si “crescono” le regole dal growing set
  - Si eliminano immediatamente le regole (*pruning)*
    - Cancellazione della sequenza finale delle condizioni
      - cancellare la condizione che massimizza *v*
        fino a che nessuna futura cancellazione migliori *v*

$$v(Rule, prunePos, pruneNeg) \equiv \frac{p-n}{P+N}$$

  - aggiungere la regola *potata* all'insieme di regole
  - eliminare ogni esempio che è soddisfatto dalla regola (p/n)

RIPPER (3)

- Regola di arresto usata: *Minimum Description Length*
- La *description length è:*

  #codifica(H) + #codifica(E)
- Dove:
  - *H* è l'ipotesi di classificazione
  - *E* sono gli errori/eccezioni a questa ipotesi

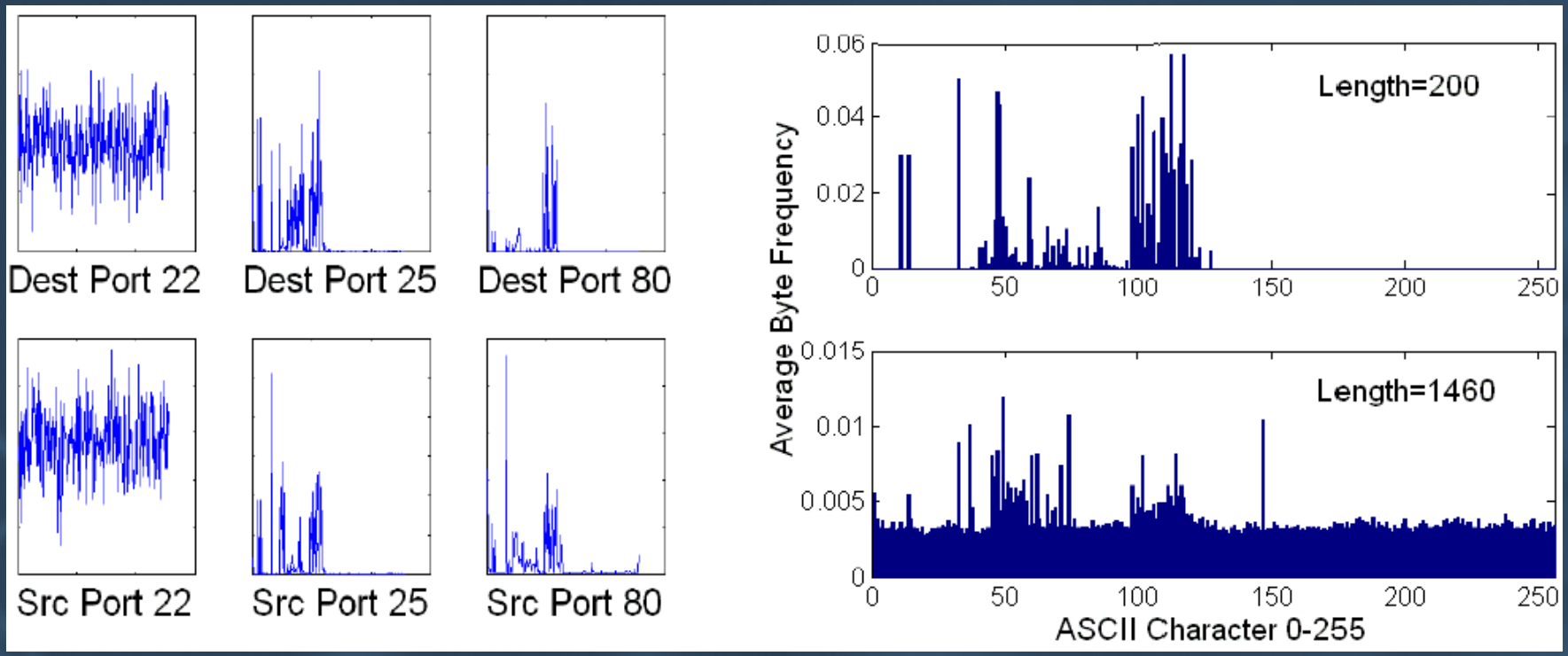
Un altro approccio...PAYL

- Passare dallo studio specifico delle informazioni sul traffico di rete, allo studio statistico del *payload*
- Costruire diversi modelli di payload differenziati per:
  - lunghezza del payload
  - porta
  - direzione del flusso di dati (in/out)

Analisi *n-gram*

- *n-gram:* una sequenza di *n* byte adiacenti all'interno del payload
- *feature vector:* la frequenza relativa a ciascun n-gram (1 char)
- *feature aggiuntive:*
  - *media*
  - *varianza*
  - *deviazione standard*

Analisi *n-gram*
Distribuzione sui caratteri



Costruzione di un modello

- Dato il *training set* si costruisca una serie di modelli M_{ij} dove:
  - *i* è la lunghezza del payload osservato
  - *j* è la porta
- M_{ij} contiene *media* e *deviazione standard*

Costruzione di un modello (2)

- Il modello costruito rappresenta il comportamento normale del sistema
- Durante il rilevamento ogni pacchetto viene preso e se la distribuzione sul payload differisce dal modello *M* si genera un avvertimento

Distanza semplificata di Mahalanobis

- *Distribuzione semplificata di Mahalanobis*
  - nuova osservazione *x*
  - modello *y*
  - *n = 256* (codifica ASCII)
  - fattore di smoothing

$$d(x, \overline{y}) = \sum_{i=0}^{n-1} \left(\left| x_i - \overline{y_i} \right| / (\overline{\sigma_i} + \alpha) \right)$$

PAYL possibilità offerte

- Approccio incrementale
  - Decadimento dei dati
  - Aggiornamento della frequenza
  - Aggiornamento della deviazione standard
- Ridurre i dati tramite raggruppamento
- Apprendimento non supervisionato

Conclusioni

- Anomaly based *vs* Signature based
- Laboratorio *vs* Implementazione reale
- Tecniche realmente adatte all'analisi *online* oppure a quella *offline*
- Scarsa varietà di autori in letteratura