Intrusion Detection Systems

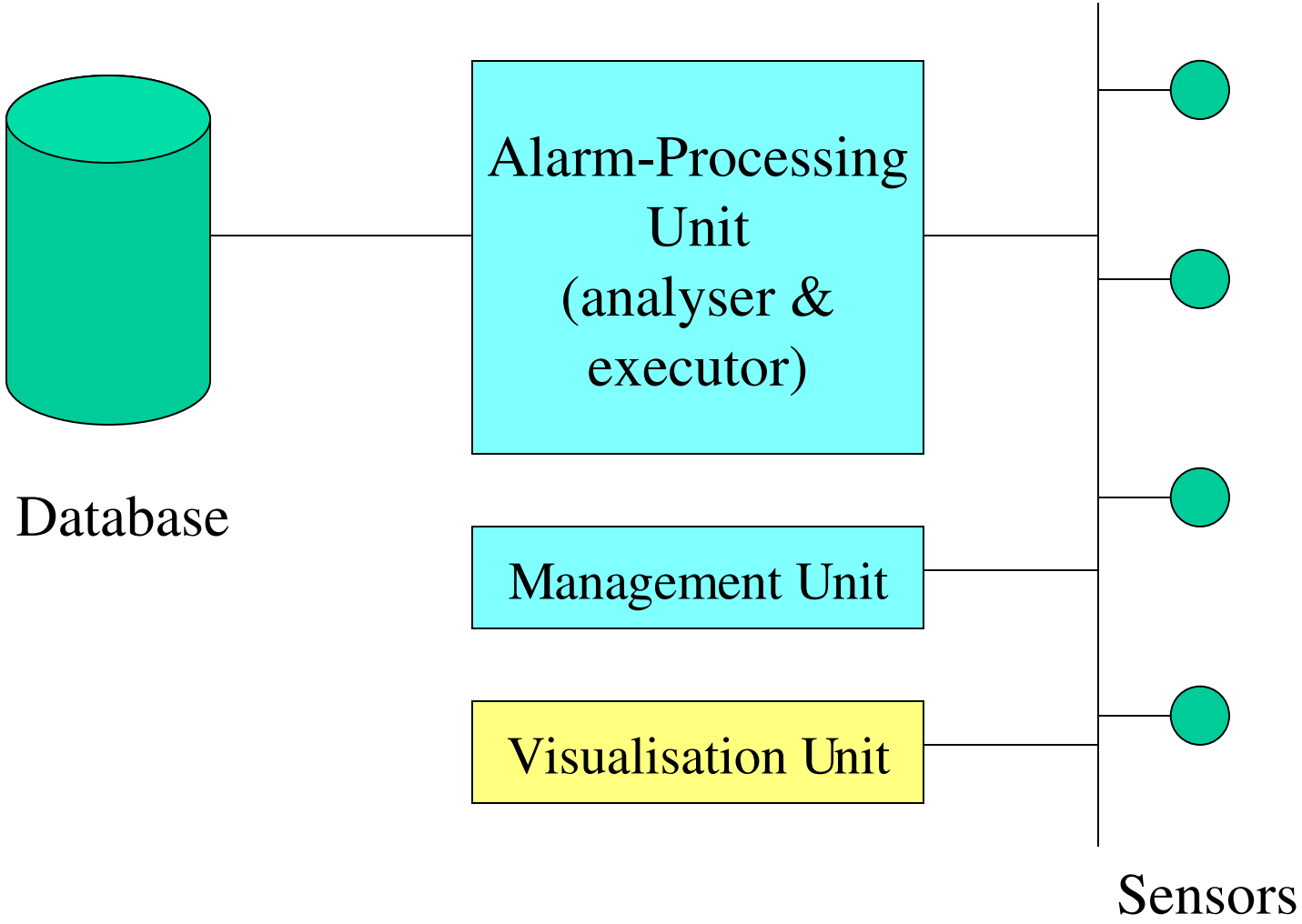
- Cosa sono gli Intrusion Detection Systems (IDS) e a cosa servono
- Snort
- Demarc, Acid e SnortSnarf
- Argus

Cosa sono gli IDS

Gli IDS sono sistemi software (oppure hardware), che cercano di identificare intrusioni o tentativi di compromissione della sicurezza di computer o di reti.

Un IDS è formato da tre componenti principali:

- uno o più sensori,
- un database,
- un'unità che processa gli allarmi.


Alarm-Processing Unit (analyser & executor)
Database
Management Unit
Visualisation Unit
Sensors

Perchè usare gli IDS

- per rilevare attacchi o altre violazioni alla sicurezza che non sono prevenuti da altri sistemi;
- per fornire utili informazioni su intrusioni avvenute, fare diagnosi e correzioni di eventuali debolezze;
- eventualmente, per avere risposte automatiche come la chiusura di una connessione, l'aumento della sensibilità di un IDS o lo spegnimento di un host sotto attacco.

Come valutare un IDS

Per valutare l'efficienza di un IDS è necessario conoscere due parametri:

- accuratezza: allarmi corretti / allarmi totali;
- completezza: allarmi corretti / numero delle intrusioni.

	Intrusion	No Intrusion
IDS Alarm	Allarme corretto	Falso allarme (o falso positivo)
IDS Rejection	Rifiuto falso	Rifiuto corretto

Snort

- Sono disponibili in rete diversi programmi che possono essere utilizzati come IDS, tuttavia molti sono a pagamento o ancora in versione beta.
- Snort è gratuito, testato, diffuso e ben supportato.
- Può funzionare sia come normale sniffer, sia come IDS. Se utilizzato in questo secondo modo, viene gestito dai seguenti file: snort.conf, classification.conf e da tutte le regole racchiuse negli *.rules.
- Gli alert possono essere salvati in binario in formato tcpdump, il che permette una maggiore velocità di scrittura e un minore utilizzo di spazio.
- Snort è quindi in grado di leggere i file in formato tcpdump per un'analisi posticipata.

snort.conf

snort.conf contiene:

- alcune variabili come HOME_NET, HTTP_SEVERS, DNS_SERVERS e altre, importanti per una buona configurazione e per diminuire i falsi positivi;
- la configurazione dei preprocessori che permettono un'analisi più estesa dei pacchetti.
- la configurazione dell'output: per esempio verso un database.
- l'elenco delle regole da usare.

*.rules e classification.conf

- le rules sono gli strumenti per identificare i pacchetti che fanno scattare gli allarmi.
- due esempi:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 8080
(msg:"SCAN Proxy attempt";flags:S; classtype:attempted-recon;
sid:620; rev:1;)
```

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80
(msg:"WEB-IIS cmd.exe access";flow:to_server; flags:
A+;content:"cmd.exe"; nocase;classtype:web-application-
attack;sid:1002; rev:3;)
```

- classification.conf contiene le priorità degli allarmi.

conclusioni su Snort

- Snort viene utilizzato come sensore e come Alarm-Processing Unit.
- per controllare il traffico di una rete deve vedere tutto il traffico che passa per il router.
- nel caso di reti fisiche diverse bisogna utilizzare più sensori (Network IDS).
- come utilizzare in maniera efficace questa grossa quantità di dati?

SnortSnarf

- SnortSnarf è un programma in Perl che prende i file di alert di Snort e produce un output html.
- non si ha un aggiornamento continuo degli alert.

Demarc

- Demarc è una management unit che utilizza diverse applicazioni per fornire i dati in tempo reale e in modo grafico;
- utilizza snort per i sensori e come analizzatore,
- Apache con OpenSSL,
- MySQL.

Demarc

- L'installazione è piuttosto laboriosa (versione 1.05) specialmente se alcune delle applicazioni necessarie a Demarc sono già presenti sul server;
- gli alert dei vari sensori sono ben configurabili;
- l'utilizzo dei servizi è user-friendly;
- è possibile fare ricerche e grafici per host, signature, porte e protocolli in diversi intervalli temporali.

Acid

- Acid fornisce una interfaccia grafica, basata su PHP, per analizzare gli eventi raccolti in un database;
- per i sensori e come analizzatore usa Snort;
- database: MySQL oppure PostgreSQL;
- web-server: va bene uno qualsiasi supportato da PHP (es: Apache).

Acid

- I servizi forniti da Acid sono molto simili a quelli di Demarc, con qualche opzione in meno;
- non è possibile configurare i sensori dall'interfaccia grafica in maniera centralizzata ma è necessario collegarsi al pc su cui si trovano;
- è maggiormente versatile, non avendo il controllo di tutti i componenti ma accedendo solo al database.

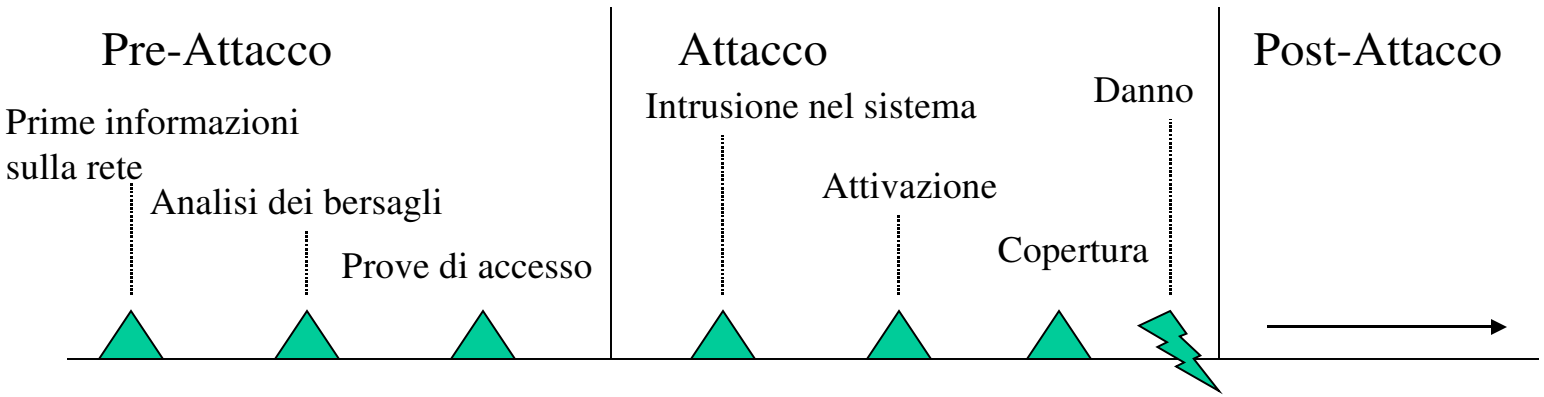
Argus

- Argus non è un IDS, perchè non analizza i pacchetti che riceve;
- registra protocollo, host e porta, sorgente e destinazione di tutte le connessioni;
- non registra il payload;
- scrive i dati in formato binario molto compatto;
- i dati possono essere analizzati tramite il comando ra e possono essere fatte statistiche con i comandi ramon e racount.

Confronti

- Snort può avere dei problemi se ha molti allarmi da registrare, in questo caso ne perde alcuni;
- Argus non elaborando i dati che riceve, registra quantità maggiori di pacchetti;
- Demarc è più completo e ricco di funzioni di Acid, però meno versatile.

Conclusioni



Gli IDS e Argus, allo stato attuale, possono essere molto utili, in una fase post-attacco per capire le debolezze del sistema e cercare collegamenti tra gli host coinvolti nell'attacco.

Riferimenti

Snort, www.snort.org

SnortSnarf, www.silicondefense.com/software/snortsnarf

Demarc, www.demarc.org

Acid, www.cert.org/kb/acid

Argus, www.quosient.com/argus