IDS: Intrusion Detection Systems

Ozalp Babaoglu

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

What Is an Intrusion?

- An intrusion can be defined as:
  - any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource
- All intrusions are defined relative to a security policy
  - A security policy defines what is permitted and what is denied on a system
  - Unless you know what is and is not allowed on your system, it is pointless to attempt to detect intrusions

Intrusion Detection and Response

- Issues
  - Threats are both internal and external
  - Firewall logs will not always alert you about intrusions and allow reconstruction
  - Intrusion detection is a necessary second line of defence (in addition to firewalls)
  - IDS deployment, customisation and management is generally not trivial

Intrusion Detection and Response

- "Manual" approach is not recommended (from CERT advisory):
  - Examine log files for connections from unusual locations or other unusual activity. For example, look at your 'last' log, process accounting, all logs created by syslog, and other security logs
  - Look for setuid and setgid files (especially setuid root files) everywhere on your system. Intruders often leave setuid copies of /bin/sh or /bin/ time around to allow them root access at a later time
  - Check your system binaries to make sure that they haven't been altered. We've seen intruders change programs on UNIX systems such as login, su, telnet, netstat, ifconfig, ls, find, du, df, libc, sync, any binaries referenced in /etc/inetd.conf, and other critical network and system programs and shared object libraries
  - Check your systems for unauthorized use of a network monitoring program, commonly called a sniffer or packet sniffer

Intrusion Detection and Response

- "Manual" approach (continued):
  - Examine all files that are run by 'cron' and 'at.' Intruders leave back doors in files run from 'cron' or submitted to 'at.' These techniques can let an intruder back on the system (even after you believe you had addressed the original compromise)
  - Check for unauthorized services. Inspect /etc/inetd.conf for unauthorized additions or changes. In particular, search for entries that execute a shell program (for example, /bin/sh or /bin/csh) and check all programs that are specified in /etc/inetd.conf to verify that they are correct and haven't been replaced by Trojan horse programs
  - Examine the /etc/passwd file on the system and check for modifications to that file. In particular, look for the unauthorized creation of new accounts, accounts with no passwords, or UID changes (especially UID 0) to existing accounts
  - Check your system and network configuration files for unauthorized entries
  - Look everywhere on the system for unusual or hidden files (files that start with a period and are normally not shown by 'ls')

Intrusion Detection Systems (IDS)

- Goal of Intrusion Detection Systems:
  - to detect an intrusion as it happens and be able to respond to it
- False positives:
  - A false positive is a situation where something abnormal (as defined by the IDS) is reported, but it is not an intrusion.
  - Too many false positives ⇒
    - you will quit monitoring your IDS because of noise.
- False negatives:
  - A false negative is a situation where an intrusion is really happening, but your IDS does not report it
  - One false negative ⇒
    - the system is compromised

Intrusion Detection Systems (IDS)

- Goal of Intrusion Detection Systems (revised):
  - You want to minimize both false negatives and false positives
  - Often, having low false negatives means high false positives — depending on IDS
- Rationale:
  - You also want to be able sleep at night without your IDS constantly paging you and your security staff
  - How much noise can you tolerate?

Characterization of IDS

- Based on data source
  - Host based
    - audit data from a single host is used to detect intrusions
  - Multihost based
    - audit data from multiple hosts is used to detect intrusions
  - Network based
    - network traffic data, along with audit data from one or more hosts, is used to detect intrusions

Characterization of IDS

- Based on model of intrusions
  - Anomaly detection model
    - the intrusion detection system detects intrusions by looking for activity that is different from a user's or system's normal behavior
  - Misuse detection model
    - the intrusion detection system detects intrusions by looking for activity that corresponds to known intrusion techniques (signatures) or system vulnerabilities

Model of intrusions

- Misuse intrusions
  - as they follow well-defined patterns, they can be detected by doing pattern matching on audit-trail information.
  - Example:
    - an attempt to create a setuid file can be caught by examining log messages resulting from system calls

Model of intrusions

- Anomaly intrusions
  - intrusions are detected by observing significant deviations from normal behavior
- Classic model for anomaly detection [Denning 87]
  - a model is built which contains metrics that are derived from system operation
  - A metric is defined as a random variable representing a quantitative measure accumulated over a period
  - Example
    - average CPU load, no. of network connections per minute, no. of processes per user, etc

Model of intrusions

- Classic model for anomaly detection
  - exploitation of a system's vulnerabilities involves abnormal use of the system;
  - therefore, security violations could be detected from abnormal patterns of system usage
- Other mechanisms:
  - neural networks
  - machine learning classification techniques
  - mimicking of the biological immune system

Characteristics of a Good IDS

- An intrusion detection system should address the following issues, regardless of what mechanism it is based on:
  - It must run continually without human supervision
    - The system must be reliable enough to allow it to run in the background of the system being observed.
  - It should not be a "black box"
    - Its internal workings should be examinable from outside
  - It must be fault tolerant
    - It must survive a system crash and not have its knowledge-base rebuilt at restart
  - It must resist subversion
    - The system can monitor itself to ensure that it has not been subverted

Characteristics of a Good IDS

- An intrusion detection system should address the following issues, regardless of what mechanism it is based on:
  - It must impose minimal overhead on the system
    - A system that slows a computer will simply not be used.
  - It must observe deviations from normal behavior
  - It must be easily tailored to the system in question
    - Every system has a different usage pattern, and the defense mechanism should adapt easily to these patterns
  - It must cope with changing system behavior over time as new applications are being added
    - The system profile will change over time, and the IDS must be able to adapt

Existing IDS systems

- Research IDS
  - AID (Adaptive Intrusion Detection system)
  - ASAX (Advanced Security audit trail Analysis on uniX
  - Autonomous Agents for Intrusion Detection
  - EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances)
  - GASSATA (Genetic Alg. for Simplified Security Audit Trail Analysis)
  - GrIDS (Graph-based Intrusion Detection System)
  - Misuse Detection Project
  - NADIR (Network Anomaly Detection and Intrusion Reporter)
  - NID (Network Intrusion Detector)
  - USTAT (State Transition Analysis Tool for UNIX)

Existing IDS systems

- Commercial IDS
  - SNORT (Opensource)
  - VCC/TripwireTM
  - CMDS (Computer Misuse and Detection System) by SAIC
  - INTOUCH NSA (Network Security Agent) by TTI
  - Kane Security Analyst by Intrusion Detection, Inc
  - NetRanger by Wheelgroup
  - OMNIGUARD Intruder Alert by Axent
  - POLYCENTER Security Intrusion Detector by Digital
  - Real Secure by ISS
  - Stalker by Haystack Labs
  - Watch Dog by InfoStream
  - G-Server by Gilian Technologies

Technology Overview

- Host Based IDS
  - Typically monitors system, event, and security logs on Windows NT and syslog in Unix environments
  - Checks key system files and executables via checksums at regular intervals for unexpected changes
  - Can use powerful regular-expressions to clearly define signatures
  - Some products listen to port activity and alert administrators when specific ports are accessed

Technology Overview

- Network Based IDS
  - Uses network packets as the data source
  - Typically utilizes a network adapter to analyze all traffic in real-time as it travels across the network
- The attack recognition module uses three common techniques to recognize attack signatures:
  - pattern, expression or bytecode matching
  - frequency or threshold crossing
  - statistical anomaly detection

Strength of Host-based IDS

- Verifies success or failure of an attack
  - Was it successful? - Log verification
- Monitors specific system activities
  - File access activity
  - Logon/logoff activity
  - Account changes
  - Policy changes
- Detects attacks that network-based IDS may miss
  - Keyboard attacks
  - Brute-Force Logins

Strength of Network-Based IDS

- Lower cost of ownership
  - Fewer detection points required
  - Greater view
  - More manageable, unintrusive
- Detects attacks that host-based systems miss
  - IP based Denial of Service
  - Packet or Payload Content
- More difficult for an attacker to remove evidence
  - Uses live network traffic
  - Captured network traffic

Strength of Network-Based IDS

- Real-time detection and response
  - Faster notification and responses
  - Can stop before damage is done - (TCP Reset)
  - Detects unsuccessful attacks and malicious intent
- Outside a DMZ - see attempts blocked by firewall
  - Critical information obtained - Policy refinement
- Operating system independence
  - Does not require information from the target OS
  - Does not have to wait until the event is logged
  - No impact on the target

Intrusioni: risposta

- I passi da seguire:
  - Analisi di tutte le informazioni disponibili
  - Comunicazione con le parti pertinenti
  - Raccolta e protezione delle informazioni
  - Contenimento dell'intrusione
  - Eliminazione dei metodi di accesso degli attaccanti
  - Riportare il sistema al normale funzionamento

Intrusioni: risposta

- Analisi di tutte le informazioni disponibili
  - Quali attacchi sono stati usati per ottenere l'accesso
  - Quali sistemi e quali dati sono stati compromessi
  - Come ha fatto l'attaccante ad ottenere accesso
  - Qual'è il comportamento attuale dell'attaccante

Intrusioni: risposta

- Comunicazione con le parti pertinenti
  - Informazioni ad altri siti interessati
    - Quelli usati dall'attaccante per attaccare il proprio sistema
    - Quelli attaccati partendo dal proprio sistema
    - Quelli visitati dall'attaccante partendo dal proprio sistema
    - Quelli sfruttati dall'attaccante per confondere le proprie tracce

Intrusioni: risposta

- Raccolta e protezione delle informazioni
  - Raccolta di tutte le informazioni
  - Conservazione delle prove
  - Salvaguardia della catena di conservazione delle prove
  - Contatti con le autorità di polizia

Intrusioni: risposta

- Come contenere un'intrusione
  - Arresto temporaneo del sistema
  - Disattivazione del sistema compromesso dalla rete
  - Disattivazione dell'accesso, dei servizi e degli account
  - Verifica che i sistemi ridondanti non siano stati compromessi

Intrusioni: risposta

- Eliminazione degli accessi all'attaccante
  - Cambiamento password
  - Reinstallazione dei sistemi compromessi
  - Eliminazione di ogni mezzo utilizzato per l'intrusione
    - Presenza di codice ostile (backdoor, trojan horse)
    - Nuovi utenti
    - File di configurazione danneggiati

Esempio di IDS: SNORT

- SNORT (http://www.snort.org/) è un Network-based IDS:
  - **Open-source**
  - **Ligthweight**: 600 Kb di sorgenti e un efficiente meccanismo di pattern-matching
  - **Realtime**: in continuo ascolto su un segmento di rete
  - **A risposta passiva**: non blocca i pacchetti "maliziosi" ma informa l'amministratore con file di log, notifica via mail, segnali acustici

Esempio di IDS: SNORT

- Altre caratteristiche vincenti di SNORT:
  - **Modulare:** estendibile con plug-in che analizzano i pacchetti prima di arrivare al motore di analisi interno (*HTTPDecode*, *Frag2*)
  - **Adattabile:** a nuove tipologie di attacchi
  - **Portabile:** Linux, Windows, MacOS X, *BSD.

Esempio di IDS: SNORT

- SNORT si basa su regole, anche molto dettagliate, che permettono di ispezionare tutti i campi dei pacchetti ed informare l'amministratore di un eventuale attacco.
- Si compone di tre moduli:
  - **Packet Sniffer**: monitora il traffico di rete usando uno sniffer esterno (*Libpcap*)
  - **Parser delle Regole:** analizza le policy dell'IDS
  - **Analizzatore dei pacchetti:** ispeziona i pacchetti ed esegue le azioni indicate nelle regole

Esempio di IDS: SNORT

- Esempi di regole:

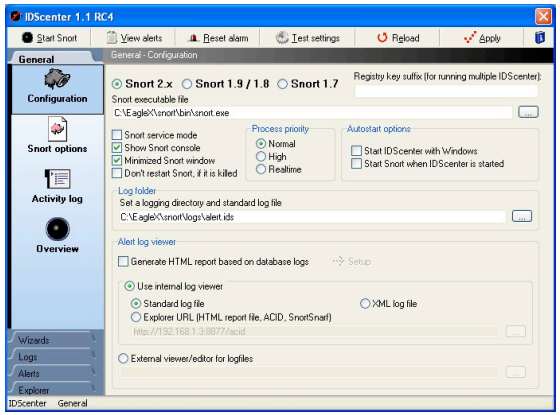
```
alert any any -> 127.0.0.1 5555 (msg:"Port 5555
  attempt";)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
  (msg:"WEB-ATTACKS /bin/ps command attempt";
  flow:to_server,established; uricontent:"/bin/ps";
  nocase; classtype:web-application-attack; sid:1328;
  rev:6;)

alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306
  (msg:"MYSQL root login attempt";
  flow:to_server,established; content:"|0A 00 00 01 85
  04 00 00 80|root|00|"; classtype:protocol-command-
  decode; sid:1775; rev:2;)
```


Esempio di IDS: SNORT

- SNORT è molto flessibile e personalizzabile. E' possibile configurare:
  - Le regole
  - Il tipo di alert (file, database, pagine HTML, notifica via mail, segnale acustico)
  - Livello di dettaglio dei log
  - Livelli di sicurezza
  - Analisi periodiche
  - Processori specializzati per i vari protocolli
  - …

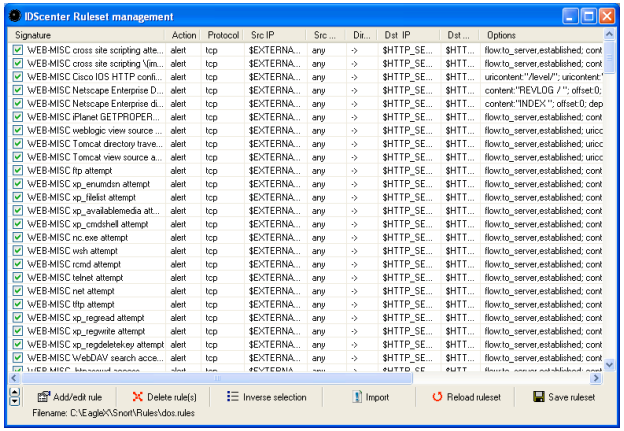
Esempio di IDS: SNORT

- IDSCenter è un front-end per SNORT su piattaforma Windows:



Esempio di IDS: SNORT

- Configurare le regole di SNORT:



Esempio di IDS: SNORT

- Configurare i meccanismi di alert: