Cybersecurity: Intrusion Detection and Cyber Forensics

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 security policies
  - 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 defense (in addition to firewalls)
  - IDS deployment, customization and management is generally not trivial

Intrusion Detection — Manual Approach

- Taken 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 — Manual Approach

- 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or /bin/csh) and check all programs that are specified in /etc/inetd.conf to verify that they are correct and haven't been replaced by Trojan horse programs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)

- An *Intrusion Detection System* is a piece of software or hardware (or both) added to a system in an effort to detect and warn of intrusions, ideally as they happen
- Has features that are similar to
  - Logging tools for networks
  - Scanning tools for operating systems
  - Anti-virus systems
  - Firewalls

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*:
  - Something abnormal (as defined by the IDS) is reported, but it is not an intrusion
  - Too many false positives ⇒ you will turn off your IDS
- *False negatives*:
  - An intrusion is going on, 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, not possible
  - Low false negatives often imply high false positives (and vice versa)
  - You want to be able sleep confidently at night but without your IDS constantly calling you or your security staff
  - How much noise can you tolerate?

Characteristics of a good IDS

- It must run continually without human supervision
  - The system must be reliable enough to allow it to run in the background on the system being observed
- It should not be a “black box”
  - Its internal workings should be examinable from outside
- It should be fault tolerant
  - It must survive system crashes without requiring its knowledge-base to be rebuilt at restart
- It should resist subversions
  - The system should monitor itself to ensure that it has not been subverted

Characteristics of a good IDS

- It should impose minimal overhead on the system
  - A system that slows a computer will simply not be used
- It should be easy to tailor it for 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

Characterization of IDS

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

Models of Intrusion

- Misuse intrusions
  - typically follow well-known patterns and can be detected through 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

Models of Intrusion

- Anomaly intrusions
  - intrusions are detected by observing significant deviations from “normal behavior”
- Classic model for anomaly detection
  - a model is built which contains metrics derived from “normal” system operation
  - *metrics* are statistics representing quantitative measures accumulated over a period
- Examples
  - average CPU load, number of network connections per minute, number of processes per user, etc.

Models of Intrusion

- Other examples of anomalies:
  - Unusual problems with system hardware or software
  - Unusual consumption of system resources
  - Unusual messages from system daemons
  - Unexplained system performance problems
  - Unusual behavior of user processes
  - Unexplained entries in the audit logs

Models of Intrusion

- Classic models
  - 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 (more complex) models based on
  - neural networks
  - ***machine learning*** classification techniques
  - mimicking biological immune systems

Anomaly vs Misuse Detection

- Misuse Detection
  - Low number of false positive, quite fast and reliable
  - Unable to detect new attacks
- Anomaly Detection
  - Very flexible, can improve their performance
  - Difficult to identify data to monitor
  - Usually require “training”

Characterization of IDS

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

Technology Overview

- Host Based IDS
  - Typically monitors system, event, and security logs
  - Checks key system files and executables via checksums at regular intervals for unexpected changes
  - Can use powerful regular-expressions to 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

- Can verify success or failure of an attack
  - 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

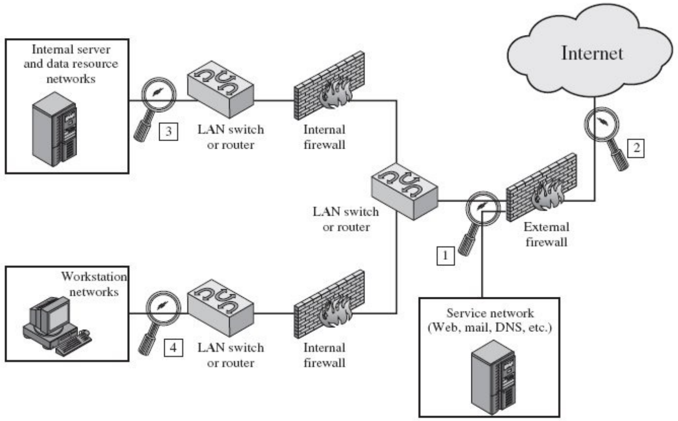
- Lower cost of ownership
  - Fewer detection points required
  - Greater view
  - More manageable, less intrusive
- Detects attacks that host-based systems miss
  - IP based Denial of Service
  - Packet or Payload Content
- More difficult for an attacker to remove evidence
  - Uses live network traffic
  - 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
- Operating system independence
  - Does not require information from the target OS
  - Does not have to wait until events are logged
  - No impact on the target

Network-Based IDS Placement

- Possible placement locations for a network-based IDS
- Placed outside the DMZ, can see attempts blocked by a firewall



Honeypots

- *Honeypot Systems* are decoy servers or systems setup to gather information regarding an attacker or intruder to your system
- Honeypot system is setup to be an easier prey for intruders than true production systems but with minor system modifications so that their activity can be logged of traced
- Goals
  - Learn how intruders probe and attempt to gain access to your systems
  - Gather forensic information required to aid in the apprehension or prosecution of intruders

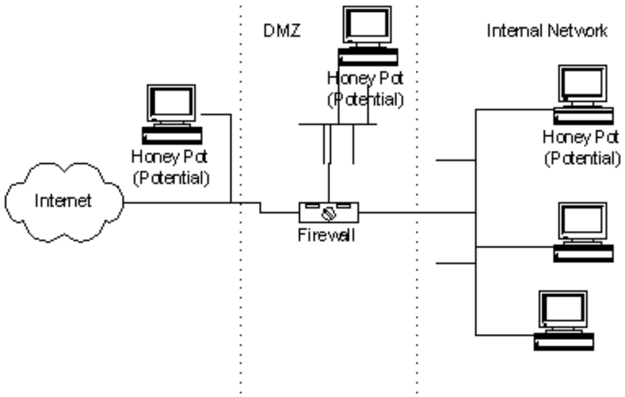
Honeypots

- The honeypot system should appear as generic as possible
- Limit the traffic you allow the intruder to send back out to the Internet so that the honeypot does not become a launch point for attacks against other entities on the Internet
- Make your honeypot an interesting site by placing “Dummy” information or make it appear as though the intruder has found an “Intranet” server

Types of Honeypots

- *Low Interaction Honeypots*: allow only limited interaction for an attacker
  - All services are emulated
  - Not themselves vulnerable
  - Less effective in tracing information
- *High Interaction Honeypots*: use of the actual vulnerable service or software
  - Complex solutions (real operating systems and applications)
  - A far more precise picture of an intrusion
  - Help in identifying unknown vulnerabilities
  - Prone to infections
  - Increases risks, can be used to attack

Honeypot placement



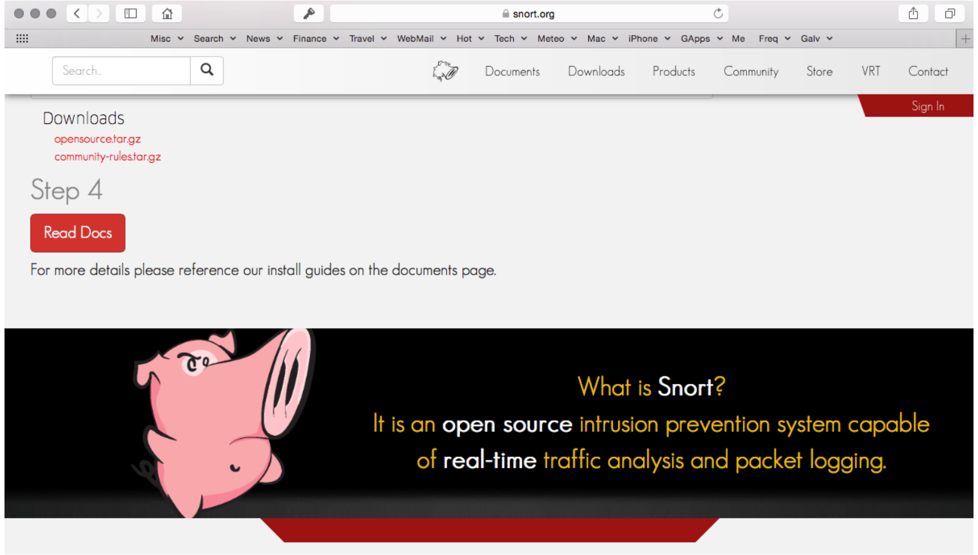
Existing IDS systems — Research

- 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 Algorithm 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

- 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

Network-Based IDS Example: SNORT



Downloads

opensource.tar.gz

community-rules.tar.gz

Step 4

Read Docs

For more details please reference our install guides on the documents page.

What is Snort?

It is an open source intrusion prevention system capable of real-time traffic analysis and packet logging.

Network-Based IDS Example: SNORT

- Open-source
- Lightweight: 600Kb of source code and an efficient mechanism for pattern-matching
- Realtime: continuous monitoring of a subnet
- Passive response: does not block “malicious” packets but informs the administrator trough the log file, mail or alarms
- Modular and adaptable: extensible through plug-ins and customizable for dealing with new attacks
- Portable: Linux, Windows, MacOS X, *BSD

Cyber Forensics

- You realize that your system has been compromised
- What do you do?
- Similar to what forensic investigators do after a crime — CSI
- Steps to follow:
  - Analyze all available information
  - Communicate with relevant parties
  - Collect and protect information
  - Contain the intrusion
  - Eliminate the vulnerabilities exploited by the attack
  - Restore the system to a safe state

Cyber Forensics

- Analyze all available information
  - Which attacks were used to gain access?
  - Which systems and what data were compromised?
  - How did the attacker gain access?
  - What is the current state of the attack?

Cyber Forensics

- Communicate with relevant parties
  - Those used by the attacker as a springboard
  - Those attacked starting from here
  - Those visited starting from here
  - Those used by the attacker to hide her tracks
  - Law enforcement authorities to report the attack
  - National CERT to report the attack

Cyber Forensics

- Collect and protect information
  - Data collection
  - Conservation of evidence
  - Protecting the chain of evidence

Cyber Forensics

- Contain the intrusion
  - Temporary shutdown of compromised system
  - Disconnecting all network connections of compromised system
  - Deactivating all logins and services
  - Verify if any redundant system has also been compromised

Cyber Forensics

- Eliminate the vulnerabilities exploited by the attack
  - Change passwords
  - Clean install of all compromised systems
  - Eliminate all other vulnerabilities used by the attack
    - Check for any installed programs that may be backdoor or trojan horse
    - Check for any newly-created accounts
    - Check for modified configuration files