Intrusion Detection Systems
an introduction

Project in the course Secure computer systems
Uppsala University
2005

by
Pablo Barron
Miroslav Horsky
Jonas Persson
Introduction

This project is an introduction to the concept of Intrusion detection systems. Our first approach to the subject was looking it up on Wikipedia.

http://en.wikipedia.org/wiki/Intrusion-detection_system

Intrusion detection systems are there covered by dividing the subject into three topics.

- Misuse Detection vs. Anomaly Detection
- Network-based vs. Host-based Systems
- Passive System vs. Reactive System

We choose to use the same division of the subject in our report.
Anomaly vs Misuse Intrusion Detection Systems

In order to get a precise view on what Anomaly and Misuse detections are and how they actually work we'll present briefly each of them.

Misuse detection is a system based on rules, either preconfigured by the system or setup manually by the administrator. The rules are looking for signatures on network and system operations trying to catch a well known attack that should be considered as Misuse. You can think of Misuse detection as a specific deny rule firewall. Exemple:
- Using one of the many SMTP/SSH exploits
- Detecting a port scan
- Parsing user commands looking for abuse

Anomaly detection on the other hand proceeds by comparing every phenomenon to what a "normal" situation would be. It seems obvious that such system needs a profile of the network/system which may be a problem in the way that it takes time and resources to train an anomaly detection sensor in order to build a profile that is reflecting a normal system / network usage. Think of Anomaly detection as an alarm for strange system behavior. Exemple:
- Excessive bandwidth usage
- Excessive system calls from a process
- More than one entity using a service

Note that both systems produce many false errors and need constant updating or training.

Now that we are familiar with the basics of Anomaly and Misuse detection we can go deeper into the advantages of using one rather then the other.

First of all let's clarify that there are very few cases when one would need to setup an Anomaly detection since creating profile may take sometimes unreasonable time and basically it is considering as "playing with the system". As every system administrator can attest, things change. Users are added, services are removed, new machines are introduced. Each of those factors requires an updated profile. In a large, diverse system, an ADS would require constant training, which makes from ADS an unappropriate tool to install on an end-system for practical real-world intrusion detection.

The good thing with Misuse detection is that it's easy to update your rules and there are also rule servers specific for many IDS that update automatically the rules.

Nowadays there is a lot of research going on in the ADS field mainly in the mathematical and statistical approach of what a "normal usage profile" is, but also in languages that could help administrators describing their network/process profile.

The most advanced distributed project in ADS is the "SecurityFocus DeepSight Threat Management System" which makes use of statistical data to detect potential Internet threats. This is a fascinating area of both IDS and ADS, since the security community works together to produce dangerous trend warnings. At the lowest level - the sensors, this is pure intrusion detection. They are reporting actual scans and attacks. Taken as a whole, however, this turns out to be a massive
anomaly detection system. In this case, the entire Internet is the system, and the individual incidents are statistical anomalies.

Without a doubt, anomaly detection techniques are also being incorporated into modern intrusion detection systems. While they might not be advertised specifically as an ADS, IDS products of the near future will generate alerts based on deviant system behavior.

**Host based vs Network bases Intrusion Detection systems**

**Host Based Intrusion Detection Systems**

* A Host-based Intrusion Detection System consists of an agent on a host which sensors identify intrusions by analyzing some of the following:
  - System calls
  - System and application logs,
  - Suspicious file-system modifications (binaries, password files, capability/acl databases)
    * Checks on permissions
    * Checks on file integrity
    * Rootkit detection (modified system binaries to hide the attacker)
  - Network attacks upon the specific system:
    * Known signature attacks
    * Port scans
    * Remote shellcode attacks (heuristic detection)
    * Backdoor checking

- Practical example: NGSEC's NGSecureWeb
  - Reactive IDS, application specific, integrated into one software piece
    --> looks for well-known attacks as well as for heuristics and logs them
    --> actively stops the requests and rises alarms
  - Can be controlled remotely through a cyphered channel
  - Installed as a module for IIS, Apache and iPlanet/SunONE
  - Checks the following:
    * Known attacks (through SNORT-compatible signatures)
    * Directory transversal attacks in CGIs
    * Forbidden words (like /bin/sh)
    * Shellcode heuristic recognition (even against polymorphic shellcodes)
    * Specific buffer overflow checking: Long HTTP headers, long GET/POST/URL values
      * Non printable characters
      * Method protection (other than GET, HEAD, POST + specified)

**Network Intrusion Detection Systems [NIDS]**

Following the schema on Sensors/Console/Engine:
The most specific part on NIDS are "Sensors", which are deployed all over the network which is to be protected.

**Sensors**

Deployment points:
- The DMZ
- Firewall
- Parts of the Network directly connected to the Internet
- Servers which provide VPN services
- Any critical servers or sub-networks.

Sensor activity:
- Server/service availability
  Checking out if all servers are running and if their services are working correctly.
- Protocol verification
  Some intrusions can use weaknesses in the underlying IP, TCP, UDP, and ICMP protocols. The NIDS can therefore have a system which can detect and flag invalid packages, or detect other attempts such like port scanning.
- Application protocol verification
  Intruders can use an application level protocol in an invalid way and then be able to get into a system. The NIDS therefore may have knowledge of the protocols being used and check that the they are used in the right way.
- Shellcode detection
  Some IDS provide with capabilities to raise alarms on data which could have a shellcode inside. For example, on attacking i386 machines it is usual to have lots of 90h (NOP) or other 1-byte instructions
- Honeypots
  Honeypots are a very specific sort of sensors; they are systems which look like vulnerable, and are installed in order to log hack attempts and even activities inside the machine once hacked.

Console and Engine implementations may vary.

- Practical example: BigSister
  * Open source network monitoring tool
    --> Can be fully re-programmed
Alarms and checks can be easily added, and external programs can be used and added to the checks.

* Sensors and Console are installed together in critical machines.

  --> They check out network availability and traffic like the center of a star, as well as the state of the machine itself (including auditing logs)

  --> They send the current state to the Engine regularly (number of seconds determined by the sysadmin)

* Centralized Engine

  --> Usually installed in an HTML server (can be private)
  --> Receives information from the different Consoles
  --> Generates HTML results/statistics and a control screen
  --> Rises alarms, mails administrators with them, etc

Passive System vs. Reactive System

In a passive system the IDS detects an intrusion and then alerts the user in some way. There is several different ways an IDS can do this.

Examples:

- It can in some way show the alert in the user's GUI, for example as a message in a console.
- It can log the event in detail
- It can in some external way notify the user; email, sms, pager etc

In a reactive system the IDS does something more concrete when an intrusion is detected.

Examples:

- It can block the intruders acces to the system, for example with reconfiguration of routers/firewall's ACL lists
- Reset the TCP connection
- In host based IDS disable the user account of the intruder or just terminate the user's session
- Trace the origin of the intruder

Some also define a third version which is called proactive IDS. In a proactive IDS the system doesn't wait for the intrusion to happen and then reacts. It stops the intrusion to happen altogether before it has succeeded with doing it's evil work. Proactive IDS is also sometimes used as another name for reactive IDS.

Final words

Though we have divided the subject into three catgeories most IDS are a combination of all of these categories. For instance it can exist an anomaly based passive network based Intrusion detection system or any other combination.
Sources

http://en.wikipedia.org/wiki/Intrusion_detection_systems
http://www.ticm.com/kb/faq/idsfaq.html
http://online.securityfocus.com/infocus/1600http://www.phrack.org/phrack/56/p56-0x0b
http://www.ngsec.com
http://bigsister.graeff.com/