Intrusion Detection
CS 239
Computer Software
March 9, 2005

Outline

- Introduction
- Characteristics of intrusion detection systems
- Some sample intrusion detection systems

Introduction

- Many mechanisms exist for protecting systems from intruders
  - Access control, firewalls, authentication, etc.
- They all have one common characteristic:
  - They don't always work

Intrusion Detection

- Work from the assumption that sooner or later your security measures will fail
- Try to detect the improper behavior of the intruder who has defeated your security
- Inform the system or system administrators to take action

Why Intrusion Detection?

- If we can detect bad things, can't we simply prevent them?
- Possibly not:
  - May be too expensive
  - May involve many separate operations
  - May involve things we didn't foresee

For Example,

- Your intrusion detection system regards setting uid on root executables as suspicious
  - Yet the system must allow the system administrator to do so
- If the system detects several such events, it becomes suspicious
  - And reports the problem

Couldn't the System Just Have Stopped This?

- Perhaps, but -
- The real problem was that someone got root access
  - The changing of setuid bits was just a symptom
- And under some circumstances the behavior is legitimate

Intrusions

- "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource"[1]
- Which covers a lot of ground
  - Implying they're hard to stop

[1]Heady, Luger, Maccabe, and Servilla, "The Architecture of a Network Level Intrusion Detection System," Tech Report, U. of New Mexico, 1990.

Is Intrusion Really a Problem?

- Is intrusion detection worth the trouble?
- Yes, at least for some installations
- Consider the experience of NetRanger intrusion detection users

The NetRanger Data

- Gathered during 5 months of 1997
- From all of NetRanger's licensed customers
- A reliable figure, since the software reports incidents to the company

NetRanger's Results

- 556,464 security alarms in 5 months
- Some serious, some not
  - "Serious" defined as attempting to gain unauthorized access
- For NetRanger customers, serious attacks occurred .5 to 5 times per month
  - Electronic commerce sites hit most

Kinds of Attacks Seen

- Often occurred in waves
  - When someone published code for a particular attack, it happened a lot
  - Because of "Script Kiddies"
- 100% of web attacks were on web commerce sites

Where Did Attacks Come From?

- Just about everywhere
- 48% from ISPs
- But also attacks from major companies, business partners, government sites, universities, etc.
- 39% from outside US
  - Only based on IP address, though

Kinds of Intrusions

- External intrusions
- Internal intrusions

External Intrusions

- What most people think of
- An unauthorized (usually remote) user trying to illicitly access your system
- Using various security vulnerabilities to break in
- The typical case of a hacker attack

Internal Intrusions

- An authorized user trying to gain privileges beyond those he is entitled to
- No longer the majority of problems
  - But often the most serious ones
- More dangerous, because insiders have a foothold and know more

Basics of Intrusion Detection

- Watch what's going on in the system
- Try to detect behavior that characterizes intruders
- While avoiding improper detection of legitimate access
- Hopefully all at a reasonable cost

Intrusion Detection and Logging

- A natural match
- The intrusion detection system examines the log
  - Which is being kept, anyway
- Secondary benefits of using the intrusion detection system to reduce the log

On-Line Vs. Off-Line Intrusion Detection

- Intrusion detection mechanisms can be complicated and heavy-weight
- Perhaps better to run them off-line
  - E.g., at nighttime
- Disadvantage is that you don't catch intrusions as they happen

Failures In Intrusion Detection

- False positives
  - Legitimate activity identified as an intrusion
- False negatives
  - An intrusion not noticed
- Subversion errors
  - Attacks on the intrusion detection system

Desired Characteristics in Intrusion Detection

- Continuously running
- Fault tolerant
- Subversion resistant
- Minimal overhead
- Must observe deviations
- Easily tailorable
- Evolving
- Difficult to fool

Host Intrusion Detection

- Run the intrusion detection system on a single computer
- Look for problems only on that computer
- Often by examining the logs of the computer

Advantages of the Host Approach

- Lots of information to work with
- Only need to deal with problems on one machine
- Can get information in readily understandable form

Network Intrusion Detection

- Do the same for a local (or wide) area network
- Either by using distributed systems techniques
- Or (more commonly) by sniffing network traffic

Advantages of Network Approach

- Need not use up any resources on users' machines
- Easier to properly configure for large installations
- Can observe things affecting multiple machines

Network Intrusion Detection and Data Volume

- Lots of information passes on the network
- If you grab it all, you will produce vast amounts of data
- Which will require vast amounts of time to process

Network Intrusion Detection and Sensors

- Use programs called sensors to grab only relevant data
- Sensors quickly examine network traffic
  - Record the relevant stuff
  - Discard the rest
- If you design sensors right, greatly reduces the problem of data volume

Styles of Intrusion Detection

- Misuse intrusion detection
  - Try to detect things known to be bad
- Anomaly intrusion detection
  - Try to detect deviations from normal behavior
- Specification intrusion detection
  - Try to detect deviations from defined "good states"

Misuse Detection

- Determine what actions are undesirable
- Watch for those to occur
- Signal an alert when they happen
- Often referred to as *signature detection*

Level of Misuse Detection

- Could look for specific attacks
  - E.g., Syn attacks or IP spoofing
- But that only detects already-known attacks
- Better to also look for known suspicious behavior
  - Like trying to become root
  - Or changing file permissions

How Is Misuse Detected?

- By examining logs
  - Only works after the fact
- By monitoring system activities
  - Often hard to trap what you need to see
- By scanning the state of the system
  - Can't trap actions that don't leave traces
- By sniffing the network
  - For network intrusion detection systems

Pluses and Minuses of Misuse Detection

+ Few false positives
+ Simple technology
+ Hard to fool
– Only detects known problems
– Gradually becomes less useful if not updated
– Sometimes signatures are hard to generate

Misuse Detection and Commercial Systems

- Essentially all commercial intrusion detection systems detect misuse
  - Primarily using signatures of attacks
- Many of these systems are very similar
  - With only different details
- Differentiated primarily by quality of their signature library
  - How large, how quickly updated

Anomaly Detection

- Misuse detection can only detect known problems
- And many potential misuses can also be perfectly legitimate
- Anomaly detection instead builds a model of valid behavior
  - And watches for deviations

Methods of Anomaly Detection

- Statistical models
  - User behavior
  - Program behavior
  - Overall system/network behavior
- Expert systems
- Misuse detection and anomaly detection sometimes blur together

Pluses and Minuses of Anomaly Detection

+ Can detect previously unknown attacks
– Hard to identify and diagnose nature of attacks
– Unless careful, may be prone to many false positives
– Depending on method, can be expensive and complex

Anomaly Detection and Academic Systems

- Most academic research on IDS in this area
  - More interesting problems
  - Greater promise for the future
- But few really effective systems currently use it
  - Not entirely clear that will ever change

Specification Detection

- Define some set of states of the system as good
- Detect when the system is in a different state
- Signal a problem if it is

How Does This Differ From Misuse and Anomaly Detection?

- Misuse detection says that certain things are bad
- Anomaly detection says deviations from statistically normal behavior are bad
- Specification detection specifies exactly what is good and calls the rest bad
- A relatively new approach

Some Challenges

- How much state do you have to look at?
  - Typically dealt with by limiting observation to state relevant to security
- How do you specify a good state?

Pluses and Minuses of Anomaly Detection

+ Allows formalization of what you're looking for
+ Limits where you need to look
+ Can detect unknown attacks
- Not very well understood yet
- Based on locating right states to examine

Customizing and Evolving Intrusion Detection

- A single intrusion detection solution is impossible
  - Good behavior on one system is bad behavior on another
  - Behaviors change and new vulnerabilities are discovered
- Intrusion detection systems must change to meet needs

How Do Intrusion Detection Systems Evolve?

- Manually or semi-automatically
  - New information added that allows them to detect new kinds of attacks
- Automatically
  - Deduce new problems or things to watch for without human intervention

A Problem With Evolving Intrusion Detection Systems

- Very clever intruders can use the evolution against them
- Instead of immediately performing dangerous actions, evolve towards them
- If the intruder is more clever than the system, the system gradually accepts the new behavior

Practicalities of Operation

- Most commercial intrusion detection systems are add-ons
  - They run as normal applications
- They must make use of readily available information
  - Audit logged information
  - Sniffed packets
  - Output of systems calls they make
- And performance is very important

Practicalities of Audit Logs for IDS

- Operating systems only log certain stuff
- They don't necessarily log what an intrusion detection system really needs
- They produce large amounts of data
  - Expensive to process
  - Expensive to store
- If attack was successful, may be corrupted

What Does an IDS Do When It Detects an Attack?

- Automated response
  - Shut down the "attacker"
  - Or more carefully protect the attacked service
- Alarms
  - Notify a system administrator
  - Who investigates and takes action

Consequences of the Choices

- Automated
  - Too many false positives and your network stops working
  - Is the automated response effective?
- Alarm
  - Too many false positives and your administrator ignores them
  - Is the administrator able to determine what's going on fast enough?

Intrusion Prevention Systems

- Essentially a new buzzword for IDS that takes automatic action when intrusion is detected
- Goal is to quickly take remedial actions to threats
- Since IPSs are automated, false positives could be very, very bad
- "Poor man's" version is IDS controlling a firewall

Sample Intrusion Detection Systems

- Emerald
- NetRanger
- CIDF

Emerald

- From SRI
- In a family of intrusion detection systems
  - IDES and NIDES were earlier versions
- Addresses practical intrusion detection problems
  - Heterogeneity
  - Scaling
  - Multiple levels of abstraction

Emerald Characteristics

- Combines multiple approaches to detecting problems
- Has built-in capabilities to invoke code to deal with problems
- Component-based architecture
- Intended to scale well

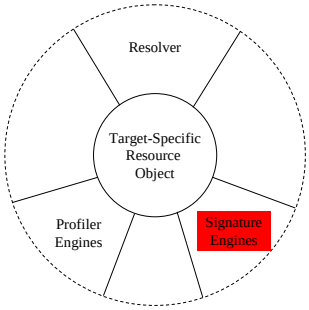
Emerald Architecture

- Divided into generic components and specific object components
- Generic components provide base engine for intrusion detection
  - No code relating to specific events or characteristics here
- Bulk of code in specific object components

Object Monitors

- Code intended to watch for intrusions on particular types of system objects
  - Types of services (FTP, HTTP)
  - Network elements (firewalls, routers)
  - Possible kinds of attacks

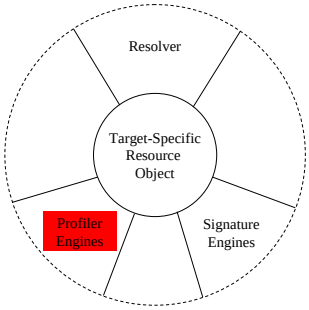
Object Monitor Architecture



Signature Engines

- Analyzes behavior to find known problems
- Uses expert systems technology
  - Allowing detection beyond pattern matching of signatures
- But also watches for problems expert system knows about

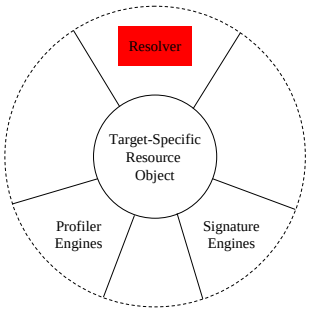
Object Monitor Architecture



Profiler Engines

- Statistically-based subsystem to watch for unusual behavior
- Types of statistical variables:
  - Categorical (discrete types)
  - Continuous (numerical qualities)
  - Traffic intensity (volume over time)
  - Event distribution (e.g., meta-measure of other measures)

Object Monitor Architecture



Resolver

- Coordinator of monitor's external reporting system
- Implements monitor's response policy
  - E.g., could shut down all HTTP traffic if things look very bad
  - Or could simply request more detailed monitoring

Customizing Emerald

- On installation, administrator chooses from library of resource objects
  - Depending on what his system does and what threats he anticipates
- Can also develop new resource objects for new/particular threats
- Goal is high reusability of code

Analyzing Systems From Multiple Perspectives

- Emerald is designed to allow correlation of multiple analyses
- E.g., detecting common types of events from different monitors
- Or combining low-rate events from different monitors
- Or analyzing the same system from multiple perspectives

NetRanger

- Now bundled into Cisco products
- For use in network environments
  - "Sensors" in promiscuous mode capture packets off the local network
- Examines data flows
  - Raises alarm for suspicious flows
- Using misuse detection techniques
  - Based on a signature database

The Common Intrusion Detection Framework (CIDF)

- An attempt to allow intrusion detection systems to interoperate
- Possibly combining advantages of all
- An architecture, a communication specification, and a language
- IETF also working on intrusion detection standard

Basic CIDF Architecture

- Several kinds of components:
  - Event generators (E-boxes)
  - Event analyzers (A-boxes)
  - Event databases (D-boxes)
  - Response units (R-boxes)

CIDF Generalized Intrusion Detection Objects (Gidos)

- The means of communicating among other components
- Some examples:
  - Encoding occurrence of particular event at particular time
  - Encoding a conclusion about a set of events
  - Transporting instruction to carry out an action

Conclusions

- Intrusion detection systems are helpful enough that those who care about security should use them
- They are not yet terribly sophisticated
  - Which implies they aren't that effective
- Much research continues to improve them