Resilient Networking

Thorsten Strufe

Module 7: Incident Management and Intrusion Detection

Disclaimer: this course has been created with very valuable input from Mathias Fischer

Dresden, SS 16
Lecture Outline

- Security Incident Management

- Intrusion Detection
  - Classification of IDS
  - Requirements to IDS
  - Collaborative IDS
  - Evasion Attacks
Cyber Attacks - Evolution

Evolution of Cyber Attacks 2000-2009 - HostExploit.com

- **DDoS Arsenal**: SYN, ICMP, GET/POST, UDP, Bad Protocols, Connection Floods, Applications Back-end, P2P, Browser Malware, DNS Spoofing.
- **Motivations**: Revenge, Extortion, Industrial/Competitive Sabotage, Political Activism, Terrorism, Theft.
Security Incident Management

„Security Information and Event Management“ (SIEM)

• Monitor different aspects of deployed system (infrastructure)
  • Different components
    – Network: routers, switches, links
    – Metal: racks, servers, OS (power, temperature...)
  • Different incidents (anomalies, failures, attacks)
  • Different sensors: (H/N)IDS, Honeypots, Honeynets

• Present situation in a simple way
  • Consider the user: admin/SysOp or NetOp who monitors 24/7

• Facilitate direct response
  • „Drill down“ diagnosis
  • Repair

• Plus social/insitutional aspects: process of information propagation
Cyber Attack Monitors

Übersicht über die aktuellen Cyberangriffe (aufgezeichnet von 101 Sensoren)

Live-Ticker

<table>
<thead>
<tr>
<th>Datum</th>
<th>Ursprung</th>
<th>Angriff auf</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-07-01 14:02:45</td>
<td>Taiwan</td>
<td>Netzwerkdiensete</td>
<td>dionaea.smbd.port.445</td>
</tr>
<tr>
<td>2013-07-01 14:02:45</td>
<td>Venezuela</td>
<td>Netzwerkdiensete</td>
<td>dionaea.smbd.port.445</td>
</tr>
<tr>
<td>2013-07-01 14:02:43</td>
<td>Italy</td>
<td>Netzwerkdiensete</td>
<td>dionaea.smbd.port.445</td>
</tr>
<tr>
<td>2013-07-01 14:02:43</td>
<td>Italy</td>
<td>Netzwerkdiensete</td>
<td>dionaea.smbd.port.445</td>
</tr>
<tr>
<td>2013-07-01 14:02:41</td>
<td>Indonesia</td>
<td>Netzwerkdiensete</td>
<td>dionaea.smbd.port.445</td>
</tr>
</tbody>
</table>

Top 15 der Ursprungsänder von Angriffen des Vormonats

<table>
<thead>
<tr>
<th>Quelle des Angriffes</th>
<th>Anzahl der Angriffe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulgaria</td>
<td>3.763.735</td>
</tr>
<tr>
<td>Russian Federation</td>
<td>1.339.175</td>
</tr>
<tr>
<td>Taiwan</td>
<td>1.189.247</td>
</tr>
<tr>
<td>United States</td>
<td>891.531</td>
</tr>
<tr>
<td>Germany</td>
<td>752.663</td>
</tr>
<tr>
<td>Hungary</td>
<td>407.022</td>
</tr>
<tr>
<td>Romania</td>
<td>357.859</td>
</tr>
<tr>
<td>Poland</td>
<td>274.120</td>
</tr>
<tr>
<td>Italy</td>
<td>232.267</td>
</tr>
<tr>
<td>Venezuela, Bolivarian Republic of</td>
<td>228.552</td>
</tr>
<tr>
<td>Israel</td>
<td>217.203</td>
</tr>
<tr>
<td>Brazil</td>
<td>201.602</td>
</tr>
<tr>
<td>Argentina</td>
<td>196.907</td>
</tr>
<tr>
<td>China</td>
<td>196.169</td>
</tr>
<tr>
<td>Japan</td>
<td>130.401</td>
</tr>
</tbody>
</table>

Top 5 der Angriffsarten des Vormonats

<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>Anzahl der Angriffe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angriff auf SMB Protokoll</td>
<td>5.824.973</td>
</tr>
<tr>
<td>Honeytrap Attacker on Port 19</td>
<td>3.358.253</td>
</tr>
<tr>
<td>Angriff auf Dienst am Port 33434</td>
<td>404.558</td>
</tr>
<tr>
<td>Angriff auf SSH Protokoll</td>
<td>352.860</td>
</tr>
<tr>
<td>Angriff auf Netbios Protokoll</td>
<td>234.482</td>
</tr>
</tbody>
</table>

Source: www.sicherheitstacho.eu
Intrusion Detection

• Definition:
  – An *intrusion* is an action or set of actions aimed at compromising the confidentiality, integrity or availability of a service or system

• Principal defense categories:
  – Prevention
  – Detection
  – Response (Mitigation & Restoration)
Why is prevention not sufficient

• Because it is too expensive to prevent all potential attack techniques
• Because legitimate users get annoyed by too many preventive measures and may even start to circumvent them (introducing new vulnerabilities)
• Because preventive measures may fail:
  – Incomplete or erroneous specification / implementation / configuration
  – Inadequate deployment by users (just think of passwords...)
Goals of IDS

• Overall goal:
  – Supervision of computer systems and communication infrastructures in order to detect intrusions and misuse

• What can be attained with intrusion detection?
  – Detection of attacks and attackers
  – Detection of system misuse (includes misuse by legitimate users)
  – Limitation of damage (if response mechanisms exist)
  – Gain of experience in order to improve preventive measures
  – Deterrence of potential attackers
Tasks of an Intrusion Detection System

• **Audit:**
  • Recording of all security relevant events of a supervised system
  • Preprocessing and management of recorded audit data

• **Detection:**
  • Automatic analysis of audit data
  • Principle Approaches:
    – *Misuse detection* (signature analysis, also called knowledge-based intrusion detection)
    – *Anomaly detection*
  • Types of errors:
    – False positive: a non-malicious action is reported as an intrusion
    – False negative: an intrusion is not detected (a “non-event”)

• **Response:**
  • Reporting of detected attacks (alarms)
  • Potentially also initiating countermeasures (reaction)
Schematic Overview of Intrusion Detection

Events → Audit System → Audit Log → Analysis → Response → IDS

Reaction → Alarm
Classification of IDS (1)

- **Scope**
  - Host-based: analysis of system events
  - Network-based: analysis of exchanged information (IP packets)
  - Hybrid: combined analysis of system events and network traffic

- **Time of analysis**
  - Post mortem analysis
  - Online analysis

- **Detection mechanism**
  - Signature-based
  - Anomaly-based
Classification of IDS (2)

- Active vs. passive monitors
  - Passive monitors
  - Active monitors: Honeypots

- Collaboration
  - Isolated IDSs
  - Collaborative IDSs (CIDS)
    - Centralized
    - Decentralized/Hierarchical
    - Distributed
Scope: Host vs. Network Intrusion Detection

- **Host Intrusion Detection Systems (HIDS)**
  - Directly on the monitored system, e.g. OS-Logs, application-logs, timestamps
  - Pros: insider attacks (file modification, file access, installation of rootkit/trojan)
  - Problems: has to be installed on every system, produces lots of information, often no realtime-analysis but predefined time intervals, hard to manage a large number of systems

- **Network Intrusion Detection System (NIDS)**
  - Monitors the network, mainly packets sniffed from network layer.
  - Commonly combination of signature detection (stateful), protocol decoding, statistical anomaly analysis
  - Can detect DoS with buffer overflow attacks, invalid packets, attacks on application layer, DDoS, spoofing attacks, port scans
  - Often used on network hubs, to monitor a segment of the network
  - Problems: Can’t detect tampering on machines (via network protocol)
Scope: IDS Placement

- **Internet**
  - Monitors all incoming traffic
  - High load
  - High rate of false alarms

- **LAN**
  - Monitors all traffic within the corporate LAN
  - Possible detection of misuse by insiders
  - Possible detection of intrusion via mobile machines (notebooks...)

- **DMZ**
  - Monitors all traffic to and from systems in the DMZ
  - Reduced amount of Data
  - Can only detect Intrusions on these Computers
Requirements to Intrusion Detection Systems

• High accuracy
  (= low rate of false positives and false negatives)

• Easy to integrate into a system / network
• Easy to configure & maintain
• Autonomous and fault tolerant operation
• Low resource requirements

• Self protection, so that an IDS itself can not easily be deactivated by a deliberate attack (in order to conceal subsequent attacks)
Detection Quality

Source: [ET04]
Recording Audit Data

• Recording audit data is an elementary security function:
  • Recording security relevant events
  • Collects the input of intrusion detection and response mechanisms
  • Can be seen as “the sensory apparatus” for reactive defense

• Audit data delivers information on:
  • who accessed
  • when, where and how
  • whose and which resource?

• Audit data requires integrity protection:
  • Otherwise, an attacker could wipe out traces of intrusive behavior
  • ? How can this be achieved?
Types of Audit Data

• Events recorded in a **computer system**:  
  • Opening of files  
  • Execution of programs  
  • Detected access violation  
  • Failed password verification  
  • etc.

• Events recorded in a **network**:  
  • Connection establishment and release  
    – Commonly labeled into flows  
  • Packets transferred from / to specific systems / ports  
  • Specific signaling events, e.g. ICMP network unreachable message, etc.

• **Application specific** events:  
  • Have to be programmed for a specific application  
  • Events are application specific and indicate security relevant activities
• **Basic idea:**
  - Some attack patterns can be described with sufficient detail → specification of “attack signatures”
  - The event audit analyzed if it contains known attack signatures

• **Identifying attack signatures:**
  - Analyzing vulnerabilities
  - Analyzing past attacks that have been recorded in the audit

• **Specifying attack signatures:**
  - Based on identified knowledge so-called rules describing attacks are specified
  - Most IDS offer specification techniques for describing rules

• **Drawbacks of *misuse detection*:**
  - Requires prior knowledge of potential attacks
  - Signature database requires continuous updating
  - High rate of false negatives if signature database is not up to date
Network IDS and intrusion prevention system

- Analysis of IP packets in real time
- Mainly signature based, each intrusion needs a predefined rule

```plaintext
alert tcp $HOME_NET any -> any 9996 \
(msg:"Sasser ftp script to transfer up.exe"; \ 
content:"|5F75702E657865|"; depth:250; flags:A+; classtype: misc-activity; \ sid:1000000; rev:3)
```

- Three step processing of captured information (capturing is done by libpcap):
  - Preprocessing (normalized and reassembled packets)
  - Detection Engine works on the data and decides what action should be taken
  - Action is taken (log, alert, pass)
Basic idea – detect behavior that differs significantly from normal use:

- **Users have certain habits in their system usage:**
  - Duration of usage
  - Login times
  - Amount of file system usage
  - Executed programs, accessed files, etc

- **Assumption:** “normal user behavior” can be described statistically
  - Requires a learning phase / specification of normal behavior

- **Analysis:** compares recorded events with reference profile of normal behavior

- **Advantage:**
  - An attack scenario needs not to be defined a priori
  - This approach can, in principle, detect unknown attacks
• Network operation anomalies
  – Caused by configuration changes

Source: [Bar01]
• “Flash crowd anomalies”
  – Caused by software releases or special interest in a web site

Source: [Bar01]
Anomaly Detection (4)

- Network abuse anomalies
  - DoS flood attacks
  - Port scans

Source: [Bar01]
• Generic anomaly detection system
Anomaly Detection – Challenges (1)

- **Defining normal behavior**
  - Defining normal region covering every possible normal behavior is difficult
  - Boundaries between normal and anomalous behavior are often not precise
  - Anomalous observation that lies close to boundary can actually be normal or vice versa (false positive and false negatives)

- **Adaptive adversaries**
  - Attackers often adapt themselves to make anomalous observation appear normal (false negatives)
  - Renders task of defining normal behavior more difficult

- **Normal behavior is not static and evolves over time**

[Chan09]
Anomaly Detection – Challenges (2)

- **Notion of an Anomaly**
  - Notion of an anomaly is different for different application domains
  - Applying a technique developed in one domain to another, is not straightforward

- **Availability of labeled data**
  - Labeled data for training/validation of models used by anomaly detection techniques is important

- **Data Noise**
  - Data often contains noise that tends to be similar to the actual anomalies and hence is difficult to distinguish and remove

- **Privacy**
  - Collecting user specific usage patterns
  - Work-related or personal habits

[Chan09]
• **Point Anomalies**
  - If an individual instance can be considered as anomalous with respect to the rest of the data \((O_1, O_2)\)
  - Focus of majority of research on anomaly detection

• **Contextual/Conditional Anomalies**
  - Data instance is anomalous in a specific context, but not otherwise
  - Notion of context is induced by structure of data set and has to be specified as part of problem formulation
  - Data instance might be contextual anomaly in given context, but identical data instance could be considered normal in different context, e.g., for time series data
• **Collective Anomalies**
  
  • A collection of related data instances is anomalous with respect to the entire data set.
  
  • Individual data instances in a collective anomaly may not be anomalies by themselves, but their occurrence together as a collection is anomalous.
  
  • Example: sequence of actions occurring in a computer:
    
    - … *http-web, buffer-overflow, http-web, http-web, smtp-mail, ftp, http-web, ssh, smtp-mail, http-web, ssh, buffer-overflow, ftp, http-web, ftp, smtp-mail, http-web* ...
    
    - Collection of events is an anomaly, while individual events are probably no anomalies when they occur in other locations in the sequence.
  
  • Collective anomalies have been explored for sequence data and spatial data.
  
  • Point and collective anomaly detection can be transformed to contextual anomaly detection problem by incorporating context information.
• **Label** associated with data instance denote whether that instance is **normal or anomalous**

• **Obtaining labeled data** that is accurate and representative of all types of behaviors is **expensive** (labeling usually done via human experts)

• Getting **labeled set of anomalous data** instances that covers **all possible types of anomalous behavior** more difficult than getting labels for normal behavior

• **Anomalous behavior often dynamic in nature**, thus new anomaly types arise for which there is no labeled training data
Based on extent of label availability, anomaly detection operates in following modes:

- **Supervised Anomaly Detection**
  - Training data available with labeled instances for normal as well as anomaly classes
  - Build predictive model for normal vs. anomaly classes and compare any unseen data against model to determine class it belongs to

- **Semisupervised Anomaly Detection**
  - Assumption that training data has labeled instances only for normal class
  - Build a model for normal class and use model to identify anomalies in the test data

- **Unsupervised Anomaly Detection**
  - No training data required
  - Implicit assumption: normal instances far more frequent than anomalies in test data
  - If assumption is not true: high false positives rate
Anomaly Detection – Classification

- Classification-based Anomaly Detection
- Nearest Neighbor-based Anomaly Detection
- Clustering-based Anomaly Detection
- Statistical Anomaly Detection
- Information Theoretic Anomaly Detection
- Spectral Anomaly Detection

[Chan09]
Classification-based Anomaly Detection (1)

- Requires two phases
  - Classification Phase: Learn a model from a set of labeled data instances
  - Testing Phase: classifies a test instance as normal or anomalous, using the classifier

- Classes of classification-based anomaly detection
  - Multi-class: teach a classifier to distinguish between each normal class and rest of the classes
  - One-class: learn boundary around normal instances using one-class classification algorithm

- Examples:
  - Neural Networks
  - Bayesian Networks
  - Support Vector Machines
  - Rule-based approaches

![Multi-class Classifier](image1)

![One-class Classifier](image2)
• **Pros**
  – Especially multi-class techniques can make use of powerful algorithms for distinguishing between instances belonging to different classes
  – Testing phase is fast, since each test instance is compared against pre-computed model

• **Cons**
  – Multi-class techniques rely on availability of accurate labels for various normal classes, which are often not available
  – Assign label to each test instance, which is disadvantage when meaningful anomaly score is desired for test instances
Clustering-based Anomaly Detection (1)

- Clustering is used to group similar data instances into clusters
- Primarily an unsupervised technique though semisupervised clustering has also been explored lately

- Three categories along the following assumptions
  1. Normal data instances belong to a cluster in the data, while anomalies do not belong to any cluster
  2. Normal data instances lie close to their cluster centroid, while anomalies are far away from their closest cluster centroid
  3. Normal data instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters
Clustering-based Anomaly Detection (2)

- **Pros**
  - Unsupervised mode possible
  - Techniques can be adapted to other complex data types by simply plugging in a clustering algorithm that can handle particular data type
  - Testing phase for clustering is fast, since number of clusters is a small constant

- **Cons**
  - Performance dependent on effectiveness of clustering algorithm
  - Many techniques not optimized for anomaly detection
    - anomalies detected as byproduct of clustering
  - Several algorithms force every instance to be assigned to some cluster
    - Might result in anomalies getting assigned to large cluster (false negatives)
  - Techniques only effective when anomalies do not form significant clusters among themselves
  - Computational complexity of clustering data
**Statistical Anomaly Detection (1)**

- Principle: *An anomaly is an observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed*

- Assumption: Normal data instances occur in high probability regions of stochastic model, while anomalies occur in low probability regions of stochastic model

- Statistical techniques fit a statistical model to given data and then apply a statistical interference test to determine if an unseen instance belongs to this model or not

- Instances that have low probability of being generated from learned model are declared as anomalies
• **Pros**
  
  – If assumption regarding underlying data distribution holds true, statistical techniques provide statistically justifiable solution for anomaly detection
  
  – Anomaly score provided is associated with confidence interval that can be used as additional information while making decision regarding any test instance
  
  – If distribution estimation step is robust to anomalies in data, unsupervised setting feasible

• **Cons**
  
  – Relying on assumption that data is generated by a particular statistical distribution, which often does not hold true
  
  – Several hypothesis test statistics that can be applied to detect anomalies, but choosing best statistic is not trivial
  
  – Histogram-based techniques simple to implement, but for multivariate data they cannot capture interactions between different attributes
Information Theoretic Anomaly Detection

- Analyzing information content of data set using different information theoretic measures such as Kolmogorov Complexity, entropy, relative entropy, ...
- Assumption: Anomalies in data induce irregularities in information content of data set

**Pros:**
- Can operate in unsupervised setting
- Do not make any assumption about underlying statistical distribution of data

**Cons:**
- Performance highly dependent on choice of information theoretic measure
- When applied to sequences and spatial data, size of substructure required, which is nontrivial to obtain
- Difficult to associate an anomaly score with a test instance using an information theoretic technique
Spectral Anomaly Detection

• Spectral techniques try to find an approximation of data using a combination of attributes that capture bulk of variability in data

• Assumption: Data can be embedded into a lower dimensional subspace in which normal instances and anomalies appear significantly different

• Pros
  – Automatic dimensionality reduction and thus suitable for handling high dimensional data sets. Can be used as preprocessing step followed by application of any existing anomaly detection
  – Unsupervised setting

• Cons
  – Useful only if normal and anomalous instances are separable in lower dimensional embedding of data
  – High computational complexity
Summary of Anomaly Detection (1)

a) **Tight cluster**
   - Any anomaly detection technique will detect anomalies in this scenario

b) **Several clusters**
   - One-class classification-based technique might learn circular boundary around entire data set and will not be able to detect anomalies in between.
   - Multi-class technique might be able to learn boundaries around each cluster and hence be able to detect anomalies in center.
   - Statistical technique may be able to detect anomalies. Clustering and nearest neighbor-based techniques will be able to detect anomalies as they are far from all other instances.

c) **Anomalous instances form a tight cluster**
   - Clustering-based and nearest neighbor-based techniques will treat these instances as normal (high false negative rate)

[Chan09]
Active vs. Passive Monitoring (1)

• **Passive Monitors**
  – Apply signature-based or anomaly-based detection
  – Network-level or application-level monitors

• **Honeypots**
  – Active components
  – Distract attackers
  – Collecting information about attacks and attacker behavior
  – Each access to a honeypot is an attack
Honeypots

- Definition: “A security resource who's value lies in being probed, attacked or compromised”
- We want to get compromised!
- Certainly not a standalone security mechanism.
- Why?
  - FUN!
  - No false-positives (in theory)!
  - Research: Malware analysis/reverse engineering
  - Reducing available attack surface/early warning system
Honeypot Classification - Interaction

• Low interaction
  • Passive honeypot that simulates network operations (usually at tcp/ip level)
  • Do not modify network traffic in any way and do not interact with attacker
  • No operating system for the attacker to interact with
  • Analysis of spammers and as active countermeasure against worms

• Medium interaction
  • Interaction with the attacker
  • Simulate network operations and provide attackers with illusion of an OS
  • Logging and analysis of more complex attacks

• High interaction
  • Real systems (e.g., VMs)
  • Attacker is provided with real OS to interact with
Honeynets and Honeytokens

• Honeynets
  – Most advanced and complex honeypots
  – Entire network of computers to be attacked
  – Primary purpose: Capturing extensive information on (internal and external) threats

• Honeytokens
  – Not a computer, but a digital entity (e.g., an Excel spreadsheet, database entry, login and password)
  – Any interaction with honeytoken indicates unauthorized or malicious activity
Honeypots – Advantages

- **Small data sets**
  - Collection of data only when someone or something is interacting with them

- **Reduced false positives**
  - Any activity with honeypots is by definition unauthorized

- **Catching false negatives**
  - Identification of new attacks as any activity with the honeypot is an anomaly

- **Encryption**
  - Capturing of attack activity independent from malicious activity being encrypted as honeypot is a communication endpoint

- **Highly Flexible**
  - Honeypots are extremely adaptable to a wide range of environments

- **Minimal Resources**
  - Even on large networks only minimal resources required
Honeypots - Disadvantages

• Risk
  – An attacker can use honeypot to attack other systems
  – Risk varies with type of honeypot used

• Limited field of view
  – Honeypots see or capture only what interacts with them
  – They are not capturing activities between all other systems
Honeypots - Dionaea

• Low Interaction honeypot for collecting malware
• Protocol simulated:
  – Basic protocol: SMB (port 445)
  – Others: HTTP, HTTPS, FTP, TFTP, MSSQL and SIP (VOIP)
• Also support for IPv6 and TLS

• Malware files: stored locally or/and sent to 3rd party entities (CWSandbox, Norman Sandbox, Anubis, VirusTotal)
Honeypots - Kippo (1)

• Low interaction SSH honeypot

• Features:
  – Presenting a fake (but “functional”) system to the attacker (resembling a Debian 5.0 installation)
  – Attacker can download his tools through `wget`, and Kippo can save them for later inspection
  – Session logs are stored in an UML-compatible format for easy replay with original timings

• Easy to install, but hard to get hackers!
Problems of IDS – Audit Data

• **Amount of log data**
  – Auditing often generates a rather high data volume
  ⇒ Significant storage capacities are required
  ⇒ Processing of audit data should be automated as much as possible

• **Location of audit data storage**
  – Alternatives: on specific “log server” or the system to be supervised
  ⇒ If stored on log server, data must be transferred to this server
  ⇒ If stored on the system to be supervised, the log uses significant amounts of resources of the system

• **Protection of audit data**
  – If a system gets compromised, audit data stored on it might get compromised either

• **Expressiveness of audit data**
  – Which information is relevant?
  – Audits often contain a rather low percentage of useful information
Problems of IDS – Privacy ("Datenschutz")

- **User identifying data elements are logged**, e.g.,
  - *Directly identifying elements*: user ids
  - *Indirectly / partly identifying elements*: names of directories and subdirectories (home directory), file names, program names
  - *Minimally identifying elements*: host type + time + action, access rights + time + action

- **IDS audits may violate the privacy of users**
  - Violation of the user’s right to determine himself which data is collected regarding his person
  - Collected information might be abused if not secured properly
  - Recording of events puts a psychological burden on users
    (→ “big brother is watching you”)

- **Potential solution**
  - *Pseudonymous audit*: log activities with user pseudonyms and ensure, that they can only be mapped to user ids upon incident detection
Problems of IDS - Analysis

• **Limited efficiency of analysis**
  - Most IDS follow a centralist approach for analysis: so-called *agents* collect audit data and one central *evaluation unit* analyzes this data
    ⇒ No (partial) evaluation in agents
    ⇒ Performance bottleneck
  - Insufficient efficiency, especially concerning attack variants and attacks with parallel actions

• **High number of false positives**
  - In practice, many IDS report too many false alarms (some publications report up to 10,000 per month)
  - Potential countermeasure: alarm correlation (→ hierarchical approach)
Further Problems of IDS

- **Self-protection**
  - Against insider attacks
  - Strategies to cope with high load

- **High maintenance overhead**

- **Cooperation between multiple IDS**
Collaborative IDSs (2)

- **Collaborative Intrusion Detection Systems (CIDS)**
  - Detection of large-scale coordinated attacks
  - Exchanging alert data between different isolated IDSs / monitors

**Advantages**

- Teamwork
  - Coordinated Decisions
  - Sharing of tasks

- Holistic View
  - Higher accuracy
  - Distributed Attack Awareness

- Architecture
  - Scalability
  - Availability
Challenges of Collaborative Intrusion Detection

- **System architecture**
  - Architectures determine how alerts from individual sensors are being shared and processed
  - Position of detection and correlation units will affect scalability and performance of CIDS

- **Alert correlation**
  - Goal of CIDS: Detect network wide attacks, reduce false positives
  - How alerts from individual sensors are correlated determines detection accuracy of a CIDS

- **Data privacy**
  - Important issue when sharing information between organizations
  - When not considered organizations may be unwilling to share their alerts

- **Security and trust**
  - Since detection accuracy of a CIDS depends on correctness of alert information provided by each participating IDS, trustworthiness of alerts need to be verified
System Architecture - Overview

Collaborative IDSs

CIDSs

Centralized

Hierarchical

Distributed

- Centralized Analysis Unit
- Hierarchical Analysis Unit
- Distributed Analysis Unit
Centralized CIDS

Main problems
• Single Point of Failure
• Scalability
Open Source CIDS

• Sensors
  – Transparent bridge between the client network and the tunnel server

• Tunnel server
  – Multiple honeypots to analyze the incoming traffic
  – OpenVPN for each tunnel

• Web – Logging server as web interface
  – Stores the analysis information from the honeypot server in database
Centralized CIDS – HoneyMap
Hierarchical CIDS

Main problems
- Accuracy
- [Single Point of Failure]
Hierarchical CIDS - GrIDS

GrIDS: Graph Based Intrusion Detection System

• Several groups of entities (departments) that form a hierarchical tree
• Detection based on activity graph construction representing hosts and their network activities
• Each host is controlled by a department which in turn are controlled by parental departments

• Departments contain
  – **Modules**: Monitor activity on hosts and networks
  – **Software manager (S)**: responsible for keeping local state and managing distributed modules
  – **Graph Engine (E)**: construction of activity graphs that are aggregated at each level

• Detection of suspicious behavior on basis of detection rules expressed through policy language

[Cheung96]
Hierarchical CIDS - HIDE

HIDE: Hierarchical Intrusion Detection using statistical preprocessing and neural network classification

- Hierarchical structure: Tiers
- Each tier contains several monitors that apply host or network intrusion detection
- Each monitor contains five components
Distributed CIDS

Main problems
- Accuracy
- Overhead
Zhou05: “A p2p collaborative Intrusion Detection System”

- Basic idea: p2p-based blacklisting of malicious IP-addresses
- Structured p2p approach (via Chord)
  - Routing evidence between peers
- Publish/Subscribe mechanism
  - Publish detected IPs
  - Subscribe to information about IPs that exist on its blacklist

- Limitations
  - Accuracy (detection is simplistic)
DOMINO: the Distributed Overlay for Monitoring Internet Outbreaks

- Hybrid architecture: axis overlay, satellite communities and terrestrial contributors
- **Axis nodes**: central component /connected through a P2P overlay
  - Highly trusted nodes (authentication through a PKI)
- **Satellite communities**: smaller networks of satellite nodes that locally implement a version of the DOMINO protocol
  - Organized in a hierarchy led by an axis node
- **Terrestrial contributors**: non-trustworthy peers who supply summaries of their detected attacks, without implementing the DOMINO protocol
Evasion Techniques to Bypass IDS

• **Signature Evasion**
  – Attack Obfuscation
  – Packet Splitting
  – Duplicate Insertion
  – Packet Overlapping

• **Anomaly Evasion**
  – Training Data Injection
  – Mimicry Attacks
  – Covert Channel Attacks
Signature Evasion - Attack Obfuscation

- Transformation of malicious code into semantically equivalent one
- As the signature will defer from the original it will not be detected

Depending on the level of mutation

- Payload mutation
  - Change the signature of the payload of the packet

- Shellcode mutation
  - Obfuscate the shellcode with polymorphic techniques
  - Easily done via popular penetration testing tools like Metasploit Framework
Signature Evasion - Packet Splitting (1)

- **IP fragmentation**
  - Process of breaking up IP diagrams into multiple packets of smaller size
  - If the IDS cannot successfully reassemble the fragments, there is a chance of missing an attack, especially when the IDS is signature based.

- **TCP segmentation:**
  - Similarly, non-reassembled TCP stream segments can lead to false negatives.

- **Example**
  - IDS looks for signature `/bin/sh` in packet payload
  - Attacker splits payload with signature in two segments, one containing `/bin` and the other containing `/sh`
Signature Evasion – Packet Splitting (2)

- MAC OS
- Linux
- Windows

NIDS

- att
- ack

Internet

- att
- ack

att

attack

attack
Signature Evasion – Duplicate Insertion

• IDS and possible victim may handle duplicated or fragments differently
• IDS lacks information about network topology and operating system of victim

• Duplicate Insertion
  – Attacker inserts segments with small TTL, so that they will be dropped before victim
  – If IDS cannot predict whether segments reach victim it will not be able to reassemble segments and see same content as victim
• Observation: different operating systems may handle packets differently
Anomaly Evasion – Mimicry Attacks

- Attack transformation by imitating normal activity
- Usually achieved with the insertion of “dummy” system calls
- Final system sequence looks normal

```c
setreuid(0, 0), chroot("pub"),
chdir("././././././././././././."). chroot="/",
open("/etc/passwd", O_APPEND|O_WRONLY),
write(fd, "toor:AAAAAAAZzzz:0:0::/bin/sh", 33),
close(fd), exit(0)
```
Covert Channel Attacks

• Aim to hide very existence of communication
• Using means of communication not normally intended to be used for it
• Covert channels similar to techniques hiding information in audio, visual, or textual content (steganography)
  – Steganography requires some form of content as cover
  – Covert channels require some network protocol as carrier
• Covert channels as method to evade IDS

[Zander07]
Covert Channels – Basic Model

Basic model: Prisoner Problem

- Two people, Alice and Bob, are thrown to prison and intend to escape, need to communicate, while Wendy the warden monitors all their messages.
- If Wendy finds any signs of suspicious messages she will place Alice and Bob into solitary confinement → information has to be hidden.
- Different warden types:
  - Passive: Can only spy on channel but cannot alter messages
  - Active: Able to slightly modify messages, but without altering semantic context
  - Malicious: Arbitrary altering of messages

Alice → Wendy → Bob

Covert Message

Shared Secret

overt channel

overt + covert channel

Covert Message

overt channel
Covert Channel Attacks – Techniques (1)

- Unused header bits, header extensions and padding
- IP identification and fragment offset
- TCP initial sequence number field
- Checksum field
- Modulating the IP TTL field
- Modulating address fields and packet lengths
- Modulating timestamp fields
- Packet rate and timing
- Message sequence timing
- Packet loss and packet sorting
- Application protocols, e.g., HTTP or DNS
- ...

Privacy and Security
Packet rate and packet timing

- Encode covert information by varying packet rates (equivalent to modulate packet timing)
- Covert sender varies packet rate between two (binary channel) or multiple packet rates each time interval
- Binary channel can transport one bit and multi-rate channel can transmit $\log_2 r$ bits per time interval ($r$ is number of different packet rates)
Summary

• Classification of IDS
  – Signature-based IDS vs. Anomaly-based IDS
  – Passive vs. active (e.g., honeypots) monitoring

• Collaborative IDS
  – Detection of large-scale and coordinated attacks
  – Centralized, hierarchical, and distributed CIDS
  – Alert correlation as major challenge

• Evasion Attacks to bypass IDS
  – Signature Evasion
  – Anomaly Evasion, e.g., via covert channel attacks
References


References
