CSE508  Network Security

3/23/2016  Intrusion Detection

Michalis Polychronakis

Stony Brook University
Intrusion

“Any set of actions that attempt to compromise the integrity, confidentiality or availability of information resources” [Heady et al.]

“An attack that exploits a vulnerability which results to a compromise of the security policy of the system” [Lindqvist and Jonsson]

Most intrusions...

Are carried out remotely
Exploit software vulnerabilities
Result in arbitrary code execution or unauthorized data access on the compromised host

Not the only way!
Intrusion Method

Social engineering (phishing, spam, scareware, phone call, …)

Viruses/malware (disks, CD-ROMs, USB sticks, downloads, …)

Network traffic interception (access credentials, keys, tokens, …)

Password guessing (root:12345678, brute force cracking, …)

Physical access (reboot, keylogger, screwdriver, …)

Software vulnerability exploitation

…
Attack Source

Local

- Unprivileged access ➔ privilege escalation
- Physical access ➔ USB and other I/O ports, BIOS, wiretapping, memory/storage acquisition, bugging input devices, physical damage, …

Remote

- Internet
- Local network (Ethernet, WiFi, 3/4G, bluetooth, …)
- Infected media (disks, CD-ROMs, USB sticks, …)
- Phone (social engineering)
  
  *Less risk, more targets*
Attack Outcome

Arbitrary code execution
Privilege escalation
Disclosure of confidential information
Unauthorized access
DoS
Erroneous output
Destruction

…
Intrusion Detection

Intrusion detection systems monitor networks or hosts for malicious activities or policy violations

Detection (IDS): just generate alerts and log identified events

Prevention (IPS): in addition, react by blocking the detected activity
Defense in Depth

An IDS is not a silver bullet solution
  Just an additional layer of defense
  Complements existing protections, detectors, and policy enforcement mechanisms

There will always be new vulnerabilities, new exploitation techniques, and new adversaries
  Single defenses may fail
  Multiple and diverse defenses make the attacker’s job harder
Defense in Depth

Securing systems retroactively is not always easy

   WiFi access points, routers, printers, IP phones, mobile phones, legacy devices, TVs, IoT, …

Detecting and blocking an attack might be easier/faster than understanding and fixing the vulnerability

   Immediate response vs. long-term treatment

Focus not only on detecting attacks

   But also on their side effects, and unexpected events in general

Example: extrusion detection/data leak prevention → detect data exfiltration
Situational Awareness

Understanding of what is happening on the network and in the IT environment

Confirm security goals
Identify and respond to unanticipated events

Diverse sources of data

Passive/active network/host monitoring, scanning/probing, performance metrics/statistics, server/transaction logs, external (non IT) indicators, …

Use data analytics to make sense of the increasing amount of data: identify features, derive models, observe patterns, …

Data mining, machine learning, …
Basic Concepts: Location

An IDS can be a separate device or a software application
  Operates on captured audit data
  Off-line (e.g., periodic) vs. real-time processing

Network (NIDS)
  NetFlow records, raw packets, reassembled streams, …
  Passive (IDS) vs. in-line (IPS) operation
  Examples: Snort, Bro, Suricata, many commercial boxes, …

Host (HIDS)
  Login times, resource usage, user actions/commands,
  process/file/socket activity, application/system log files, registry
  changes, API calls, system calls, executed instructions, …
  Examples: OSSEC, El Jefe, AVs, registry/process/etc. monitors, network
  content scanners, …
Basic Concepts: Location

LAN

Wi-Fi

DMZ

Internet

NIDS: packets

Router/Gateway/Proxy: NetFlow records, packets, TCP streams

HIDS: Any network/host data/events

Necessary for observing internal network events
Deployment

NIDS: protect many hosts with a single detector
HIDS: install detector on each host (might not always be feasible)

Visibility

NIDS: can observe broader events and global patterns
HIDS: observes only local events that might not be visible at the network

Context

NIDS: packets, unencrypted streams (unless proxy-level SSL inspection)
HIDS: full picture

Overhead

NIDS: none (passive)
NIPS/Proxy: adds some latency
HIDS: eats up CPU/memory (overhead from negligible to complete hogging)

Subversion

NIDS: invisible in the network
NIPS/Proxy: failure may lead to unreachable network
HIDS: attacker may disable it and alter the logs (user vs. kernel level, in-VM vs. out-of-VM, remote audit logs)
Basic Concepts: Detection Method

**Misuse detection**

- Predefined patterns (known as “signatures” or “rules”) of known attacks
- Rule set must be kept up to date
- Manual vs. automated signature specification (latter is *hard*)
- Can detect only *known* attacks, with adequate precision

**Anomaly detection**

- Rely on models of “normal” behavior
- Requires (re)training with an adequate amount of data
- Can detect previously unknown attacks
- Prone to false positives
IDS Challenges

Conflicting goals
  Zero-day attack detection
  Zero false positives

Resilience to evasion

Detection of targeted and stealthy attacks

Adaptability to a constantly evolving environment
  New threats, new topology, new services, new users, …
  Rule sets must be kept up to date according to new threats
  Models must be updated/retrained (concept drift)

Coping with an increasing amount of data
Popular Open-source Signature-based NIDS

Snort

Bro

Suricata
Use Case: Snort

Captured packets (Libpcap, PF_RING, …)

Packet decoding

Preprocessors

Detection engine

Alert generation

Flow tracking

TCP stream reassembly

Protocol decoding

Byte matching

String searching

Regular expression matching

Protocol field properties

DataBase

Log file

...
What is a Signature?

An attack description as seen at Layer 2-7

Witty worm Snort signature example:

```
action  protocol  source/destination  content

alert udp any 4000 -> 193.92.123.0/24 any (msg:"ISS PAM/Witty Worm Shellcode"; content:"\x65 \x74 \x51 \x68 \x73 \x6f \x63 \x6b \x54 \x53\"; depth:246; sid:1000078; rev:1;)
```
More Examples

String searching

alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE Linux shellcode"; content:"|90 90 90 E8 C0 FF FF FF|/bin/sh"; classtype:shellcode-detect; sid:652; rev:9;)

Strsearch + regexp matching + stateful inspection

alert tcp $EXTERNAL_NET any -> $HOME_NET 10202:10203 (msg:"CA license GCR overflow attempt"; flow:to_server,established; content:"GCR NETWORK<"; depth:12; offset:3; nocase; pcre:"/^\S{65}\S+\S{65}\S+\S+\S+\S{65}/ Ri"; sid:3520;)
Stateful Inspection

Semantic gap: NIDS processes individual packets, while applications see a contiguous stream (TCP)

*Potential for evasion*

![Diagram showing the semantic gap between NIDS and applications](image)

Solution: IP defragmentation, TCP stream reassembly

Flow-level tracking: group packets into flows, track TCP state
Stream reassembly: normalize and merge packets into streams

Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection – 1998
Different TCP stacks may treat corner cases differently...
Anomaly Detection

Training phase: build models of normal behavior

Detection phase: alert on deviations from the model

Many approaches

Statistical methods, rule-based expert systems, clustering, state series modeling, artificial neural networks, support vector machines, outlier detection schemes, …

Good for noisy attacks

Port scanning, failed login attempts, DoS, worms, …

Good for “stable” environments

E.g., web server vs. user workstation
Anomaly Detection

Learning

*Supervised*

Labels available for both benign data and attacks

*Semi-supervised*

Labels available only for benign data

*Unsupervised*

No labels: assume that anomalies are very rare compared to benign events

Many possible features

Packet fields, payload content, connection properties, traffic flows, network metrics, system call sequences, code fragments, file attributes, statistics, …
Evaluating Intrusion Detection Systems

Accuracy is not a sufficient metric!

Example: data set with 99.9% benign and 0.1% malicious events

Dummy detector that marks everything as benign has 99.9% accuracy...

False positive: legitimate behavior was detected as malicious
False negative: an actual attack was not detected

<table>
<thead>
<tr>
<th>Actual Event</th>
<th>Detection Result</th>
<th>Precision = TP / (TP+FP)</th>
<th>Recall = TP / (TP + FN) (sensitivity)</th>
<th>FP rate = FP / (FP + TN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive (malicious)</td>
<td>Positive (alert)</td>
<td>TP</td>
<td>FP</td>
<td></td>
</tr>
<tr>
<td>Negative (benign)</td>
<td>Negative (silence)</td>
<td>FN</td>
<td>FP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Receiver Operating Characteristic (ROC) Curve

Concise representation of a detector’s accuracy

- **Y axis:** success rate of detecting signal events
- **X axis:** error rate of falsely identifying noise events
Evasion – “Stay under the radar”

Both anomaly and misuse detection systems can be evaded by breaking the detector’s assumptions

- Detectors rely on certain features
- Make those features look legitimate or at least non-suspicious

Many techniques

- Fragmentation
- Content mutation/polymorphism/metamorphism
- Mimicry
- Rate adjustment (slow and stealthy vs. fast and noisy)
- Distribution and coordination (e.g., DoS vs. DDoS)
- Spoofing and stepping stones

...