Lecture 5
Intrusion Detection

Print version of the lecture in *CPS633 Computer Security*

presented on Week of October 7, 2019

by Ali Miri ©2019from Department of Computer Science at Ryerson University

5.1

Learning Objectives (page 252)

- Understand the basic principles of and requirements for intrusion detection.
- Discuss the key features of host-based intrusion detection and network-based intrusion detection
- Define the intrusion detection exchange format.
- Explain the purpose and the role of honeypots.

This set of slides closely follows Chapter 8 of the textbook.

5.2

Contents

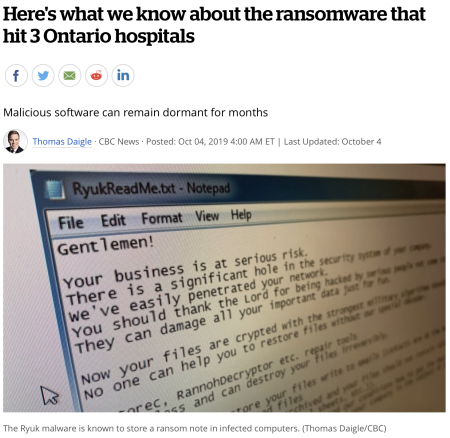
5.3

1 Preliminaries

1.1 Definitions

Definitions

Definition (*Merriam-Webster*) : *the act of intruding or the state of being intruded*
especially : the act of wrongfully entering upon, seizing, or taking possession of the property of another



source:https://www.cbc.ca/news/technology/ransomware-ryuk-ontario-hospitals-1.5308180

5.4

Definitions

" `Intrusion detection` is the process of monitoring the events occurring in a **computer system** or **network** and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of *computer security policies*, *acceptable use policies*, or *standard security practices*."

NIST 800-94: Guide to Intrusion Detection and Prevention Systems (IDPS), available at https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf

5.5

1.2 Considerations

Design Considerations

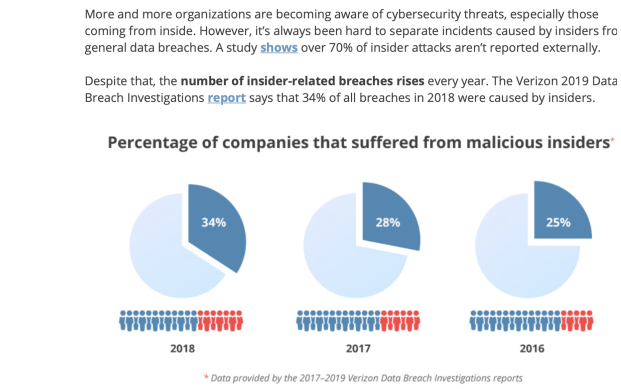
- What to monitor?
- Where to do the monitoring?
- How to analyze the collected data, and what to do?

5.6

1.3 Observations

Observations

Attacks are not always from outside!!



source:https://www.ekransystem.com/en/blog/insider-threat-statistics-facts-and-figures

5.7

Observations

Attackers may include state actors!!

source:Washington Post article, Jan 18, 2019

" The nation that *Lookout* is profiling started with a $23M budget for spyware ··· The government apparently negotiated with companies – including major spyware players such as Italy's Hacking Team, and Isarel's NSO Group – offering complex hacking tools that cost as little as $50,000 and as much as $7M.

5.8

2 Detection

2.1 Components

Basic Components

- **Sensors**: Responsible for collecting data
- **Analyzers**: Responsible for analyzing sensors' data to determine whether an intrusion is occurred. May also be used to determine the next step, i.e. *response*, and logging the info for future use.
- **User interface**: It provide authorized users an an input and output (console) interface to the IDS.

Complete IDS systems often have other complementary components.

5.9

2.2 Goals

Design Goals

```
Even best designed system will fail!
```

- **False Positives (FP)** or *false alarms*: identifying authorized users/process as intruders. This is in contrast with **True Positive (TP)**
- **False Negative (FP)**: intruders not being identified as intruders. This is in contrast with **True Negative (TN)**

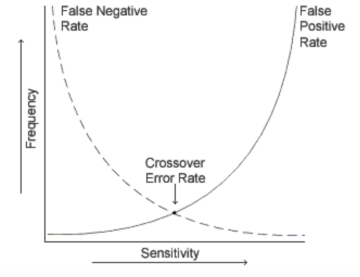
Often, we are interested in the frequency (rate, or percentage) of these occurrences:

- The **detection rate**, or the **true positive rate**: $\frac{\text{TP}}{\text{TP+FP}} \times 100\%$ - the percentage of reported attack events that are real attack events
- The **false alarm rate**, or the **false positive rate**: $\frac{\text{FP}}{\text{FP+TN}} \times 100\%$ - the percentage of non-attack events reported as attack events
- The **accuracy**: $\frac{\text{TP+TN}}{\text{TP+FP+FP+FN}} \times 100\%$

5.10

Design Goals

Improving accuracy is challenging task!



- **Base-Rate Fallacy**
  - If *TP* is low (compared to the actual number of intrusion) ← a false sense of security
  - If *FP* is high (compared to the actual number of intrusion) ← warnings generated may be ignored!

5.11

Design Goals

- Detect a wide variety of intrusions
- Detect intrusions in a timely fashion
- Present the analysis in a simple, easy-to-understand format
- Be accurate

2.3 Requirements

Requirements

- Run continually (minimal supervision)
- Be fault tolerant
- Resist subversion
- Impose a minimal overhead
- Allow for (dynamic) configuration
- Be scalable
- Adapt to changes in system and user behaviour
- Provide graceful degradation of service

5.13

2.4 Approaches

Approaches

Common Detection Methodologies:

"Once bitten, twice shy!"

- Study and analyze past attacks, and use that knowledge/patterns to detect the incoming attacks ⇒ **Signature-Based Detection**

"This does not feel right!"

- Study and analyze the 'normal' behaviour of users and systems, and use (significant) deviation from that behaviour as indication of possible attacks ⇒ **Anomaly Detection**

Other terms used in the literature: *misuse detection*, and *heuristic detection*

5.14

Signature-Based Detection

Advantages:

- Simple - only need to compare against known attacks/patterns
- Low cost -in terms of time and resource use

Disadvantages:

- Signatures need to be large enough to reduce FPs
- Cannot detect new zero-day attacks, variants of known attacks, or those using evasion techniques
- Have little understanding of many network or application protocols and cannot track and understand the state of complex communications
- Typically lack the ability to remember previous events when processing the current event

5.15

Anomaly Detection

Advantages:

- Many different attributes can be used to define a 'normal' behaviour
- Can be very effective at detecting previously unknown attacks.

Disadvantages:

- Need 'dynamic' profiles
- malicious activity may become part of a profile
- May be difficult to build profiles for highly complex computing activities
- May produce a large number of FP
- May be difficult for for analysts to determine the reason(s)

5.16

More on Anomaly Detection

Different approches to anomaly detection

- **Threshold Metrics:** Expects the occurrence of particular event to be in a given range. If outside that range → anomalous.
- **Statistical:** Analysis of the observed behaviour using *univariate*, *multivariate*, or *time-series* models of observed metrics.
- **Knowledge based:** Approaches use an expert system that classifies observed behaviour according to a set of rules that model legitimate behaviour.
- **Machine-learning**: Approaches automatically determine a suitable classification model from the training data using data mining techniques.

5.17

Machine-learning-based IDS

Machine-learning-based anomaly detection techniques:

- Bayesian networks
- Markov models
- Neural networks:
- Fuzzy logic
- Genetic algorithms
- Clustering and outlier detection

5.18

Machine-learning-based IDS

Advantages:

- Flexibility
- Adaptability
- Ability to capture interdependencies between the observed metrics

Disadvantages:

- Dependency on assumption about accepted behaviour
- High FR
- High resource
- Need for 'rich' training set.

5.19

2.5 Placement

Architectural Design Considerations*

- Where to place sensors and analyzers?
- How reliable the solution should be and what to do to achieve it?
- Where the other components of the IDS (management servers, database servers, consoles, $\cdots$ will be located?
- With which other systems the IDS needs to interface? For examples, SIEMS, log servers, email, routers, switches, firewalls, $\cdots$
- Whether or not a management network will be used?

*Based on NIST 800-94: Guide to Intrusion Detection and Prevention Systems (IDPS)

5.20

Placement

Effectiveness and costs of IDS will depend on its placement within a system, as well as the characteristics being monitored.

- **Host-based IDS (HIDS)**
- **Network-based IDS (NIDS)**
- **Distributed or hybrid IDS**

5.21

3 HIDS

3.1 Data Sources

HIDS Data Sources

Q. What should HIDS sensor monitor and log?

- **System call traces**
- **Audit (log file) records**
- **File integrity checksums**
- **Registry access**

5.22

HIDS Data Sources

accept, access, acct, adjtime, aiocancel, aioread, aiowait, aiowrite, alarm, async_daemon, auditsys, bind, chdir, chmod, chown, chroot, close, connect, creat, dup, dup2, execv, execve, exit, exportfs, fchdir, fchmod, fchown, fchroot, fcntl, flock, fork, fpathconf, fstat, fstat, fstatfs, fsync, ftime, ftruncate, getdents, getdirentries, getdomainname, getdopt, getdtablesize, getfh, getgid, getgroups, gethostid, gethostname, getitimer, getmsg, getpagesize, getpeername, getpgrp, getpid, getpriority, getrlimit, getrusage, getsockname, getsockopt, gettimeofday, getuid, gtty, ioctl, kill, killpg, link, listen, lseek, lstat, madvise, mctl, mincore, mkdir, mknod, mmap, mount, mount, mprotect, mpxchan, msgsys, msync, munmap, nfs_mount, nfssvc, nice, open, pathconf, pause, pcfs_mount, phys, pipe, poll, profil, ptrace, putmsg, quota, quotactl, read, readlink, readv, reboot, recv, recvfrom, recvmsg, rename, resuba, rfssys, rmdir, sbreak, sbrk, select, semsys, send, sendmsg, sendto, setdomainname, setdopt, setgid, setgroups, sethostid, sethostname, setitimer, setpgid, setpgrp, setpgrp, setpriority, setquota, setregid, setreuid, setrlimit, setsid, setsockopt, settimeofday, setuid, shmsys, shutdown, sigblock, sigpause, sigpending, sigsetmask, sigstack, sigsys, sigvec, socket, socketaddr, socketpair, sstk, stat, stat, statfs, stime, stty, swapon, symlink, sync, sysconf, time, times, truncate, umask, umount, uname, unlink, unmount, ustat, utime, utimes, vadvise, vfork, vhangup, vlimit, vpixsys, vread, vtimes, vtrace, vwrite, wait, wait3, wait4, write, writev

Ubuntu Linux System Calls

5.23

HIDS Data Sources

comctl32

kernel32

msvcpp

msvcrt

mswsock

ntdll

ntoskrnl

user32

ws2_32

Key Windows DLLs and Executables

5.24

3.2 Distributed HIDS

Distributed HIDS

One vs Many!

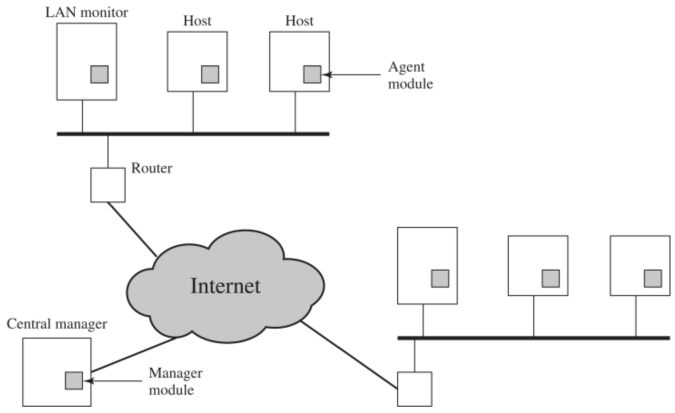
- a more effective defence, if using cooperation HIDSs across the network.

However

- May need to deal with different sensor data formats
- Need to ensure integrity and confidentiality data sent across the network
- In a centralized architecture, it may create a bottle neck, and possible major point of attack
- In a decentralized architecture, it would require coordination

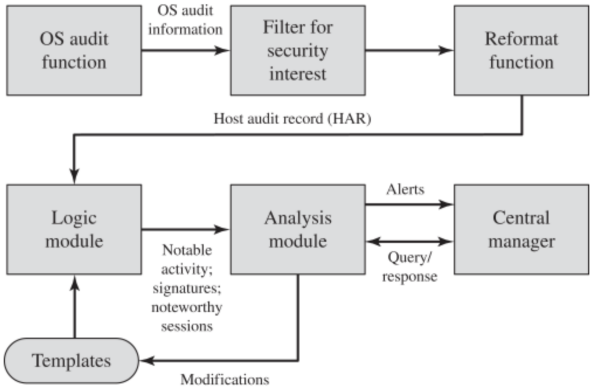
5.25

Distributed HIDS



5.26

Distributed HIDS



5.27

4 NIDS

4.1 Data Sources

NIDS Data Sources

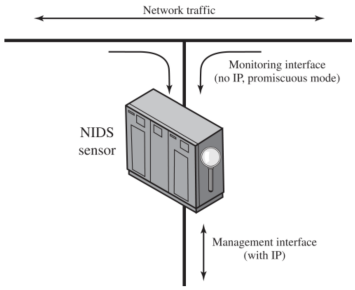
- INIDS sensor monitors and analyzes network activity on one or more network segments → NICs have to be in '*promiscuous mode*'
- Sensors are available in two formats
  - **Appliances**: specialized hardware and sensor software. Typically have a security hardened OS, with no direct, network access
  - **Software Only**: can be installed on any, or customized OS.

5.28

NIDS Sensor Types

NIDS sensors are also categorized into two types, depending on deployment modes:

- **Inline sensor:** It s inserted into a network segment so the traffic that it is monitoring must pass through the sensor
- **Passive sensor:** It monitors a copy of network traffic; the actual traffic does not pass through the device



5.29

NIDS Data Sources

NIDS sensors collect information from different layers of network stack:

- **Application layer information:** DHCP, DNS, HTTP, FTP, IMAP and SMTP, POP, NFS, RPC, SIP, SMP, SNMP, $\cdots$; as well as attack patterns associated with buffer overflows, password guessing, and malware transmission.
- **Transport layer information:** TCP, UDP and other transport layer protocol info, such unusual packet fragmentation, scans for vulnerable ports, and TCP-specific attacks such as SYN floods.
- **Network layer info:** IPv4, IPv6, ICMP and IGMP info, such as spoofed IP addresses and illegal IP header values

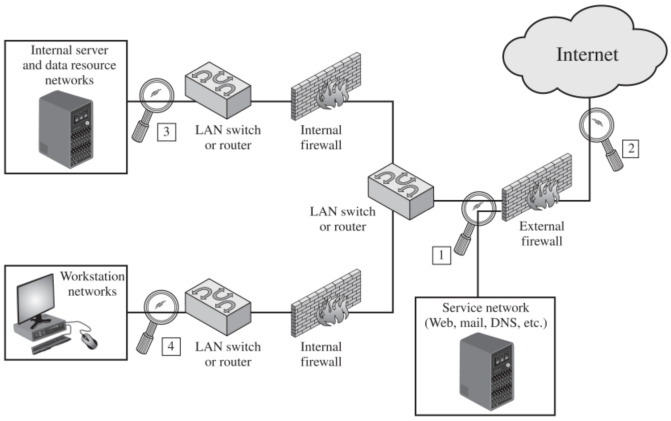
Additionally, NIDS sensors can watch for:

- **Unexpected application services**
- **Policy violations**

5.30

4.2 Deployment

Network Architectures and Sensor Locations



There are advantageous and disadvantageous for each deployment location. What are they?

5.31

4.3 Capabilities

Capabilities

- **Information Gathering Capabilities:**
  - Identifying Hosts
  - Identifying Operating Systems
  - Identifying Applications
  - Identifying Network Characteristics

5.32

Capabilities

- **Logging Capabilities:**
  - Timestamp
  - Connection or session ID
  - Event or alert type
  - Rating (e.g., priority, severity, impact, confidence)
  - Network, transport, and application layer protocols
  - Source and destination IP addresses
  - Source and destination TCP or UDP ports, or ICMP types and codes
  - Number of bytes transmitted over the connection
  - Decoded payload data, such as application requests and responses
  - State-related information

5.33

Capabilities

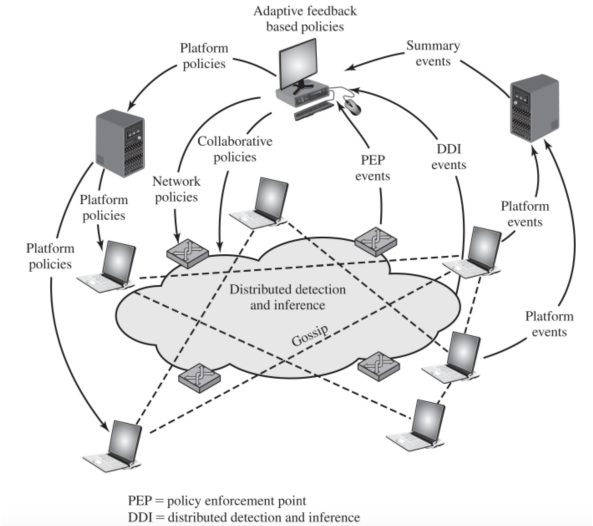
- **Detection Capabilities:**
  - Signature-based detection
  - Anomaly-based detection
  - Stateful protocol analysis techniques

5.34

5 Hybrid IDS

5.1 Architecture

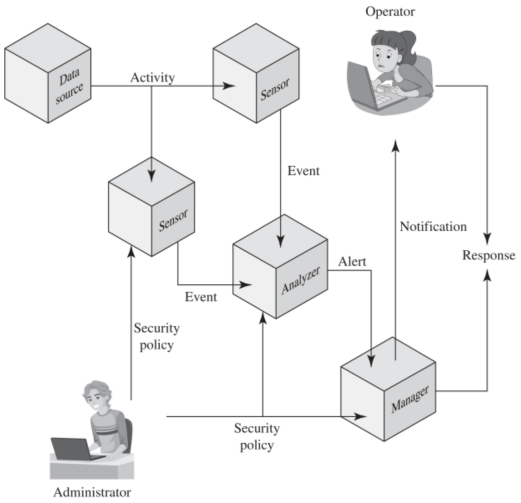
Hybrid IDS



5.35

5.2 Exchange Format

Model



5.36

Exchange Formats

Need to define exchange formats for

- event and alert messages
- message types
- exchange protocols for communication of intrusion detection

IETF has the following 3 RFC's to formalize this (2007):

- **Intrusion Detection Message Exchange Requirements (RFC 4766)**: specifies the exchange format requirements
- **The Intrusion Detection Message Exchange Format (RFC 4765)**: specifies a data model, such as XML to be used
- **The Intrusion Detection Exchange Protocol (RFC 4767)**: specifies an application-level protocol for data exchange between entities. Supports mutual-authentication, integrity, and confidentiality over a connection-oriented protocol

5.37

6 Honeypots

6.1 Concept

Concept

Honeypots are decoys systems designed to

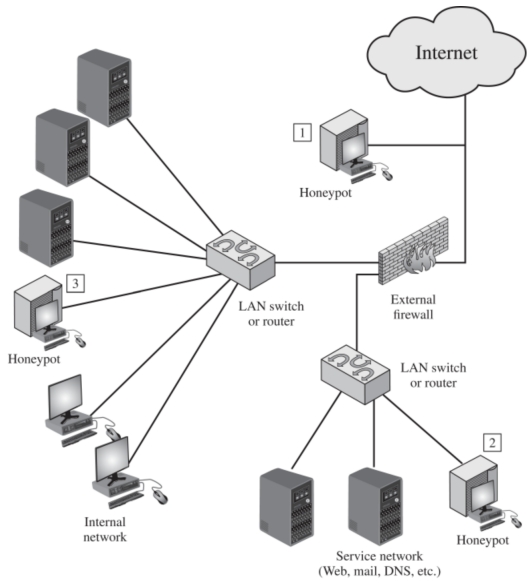
- Divert an attacker from accessing critical systems
- Collect information about the attacker's activity
- Encourage the attacker to stay on the system long enough for administrators to respond

- Low interaction honeypot: Not a full replica, but realistic enough (typically a software package)
- High interaction honeypot: a full-fledge, real-system replica

5.38

6.2 Deployment

Deployment



5.39

7 Snort

7.1 Characteristics

Characteristics

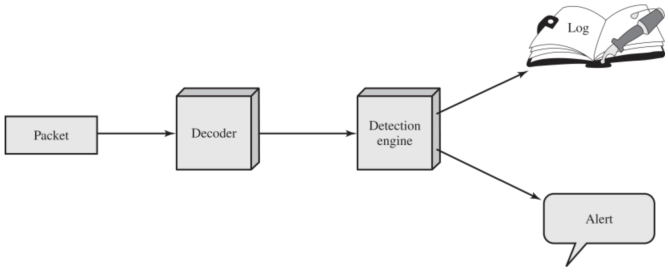
Snort: An open-source (`https://www.snort.org`) NIDS

- Easily deployed on most network nodes
- Efficient operation
- Easily configurable
- Mainly designed to work with TCP, UDP and ICMP

5.40

7.2 Architecture

Characteristics



5.41

7.3 Rules

Rules

Action	Protocol	Source IP address	Source port	Direction	Dest IP address	Dest port

(a) Rule header

Option keyword	Option arguments	. . .

(b) Options

5.42