Computer & Information Security (3-721-460-1)

Intrusion Detection

Dept. of Software and Information Systems Engineering, Ben-Gurion University

Prof. Yuval Elovici, Dr. Asaf Shabtai
{elovici, shabtaia}@bgu.ac.il

Spring, 2019

Intruders

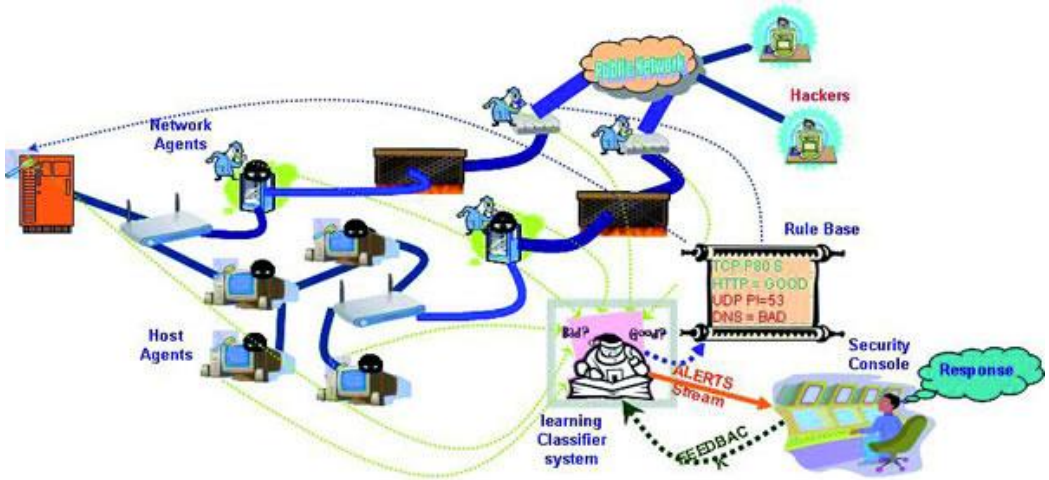
- two most publicized threats to security are malware and intruders
- generally referred to as a hacker or cracker
- An unauthorized individual who penetrates a system to exploit a legitimate user account (masquerader)
- legitimate user who misuses privileges
- Seize supervisory control to evade auditing and access controls or to suppress audit collection

Firewalls and Intrusion Prevention Systems

- effective means of protecting LANs
- Internet connectivity essential
  - for organization and individuals
  - but creates a threat
- could secure workstations and servers
- also use firewall as perimeter defence
  - single choke point to impose security

Intrusion detection

- Activities that violate the security policy of a system are often called intrusions
- Intrusion detection is the process used to identify intrusions and/or intrusion attempts
- Intruder problem led to establishment of computer emergency response teams (CERTs)
- IDS vs. IPS



Examples of Intrusion

- accessing resources without authorization
- remote root compromise
- web server defacement
- guessing / cracking passwords
- copying databases containing sensitive data
- viewing sensitive data without authorization
- running a packet sniffer
- denial of service
- …

Goals of the IDS

- Answer the questions:
  - What happened?
  - Who was affected? Who was the attacker?
  - How are they affected? How did the intrusion occur?
  - Where and when did the intrusion originate?
  - Why were we attacked?

- ID aims to positively identify all attacks and negatively identify all non-attacks

Hacker Behavior - example

- select target
- map network for accessible services
- identify potentially vulnerable services
- brute force (guess) passwords
- install remote administration tool
- wait for admin to log on and capture password
- use password to access remainder of network

Criminal Enterprise, APT, Insiders

- make their activities harder to detect
- among most difficult to detect and prevent
- use Trojan horses to leave back doors for re-entry
- do not stick around until noticed
- make few or no mistakes
- employees have access & systems knowledge
- IDS / IPS may help but also need:
  - least privilege, monitor logs, strong authentication, termination process to block access & mirror data

Definitions (RFC 2828 – Internet Security Glossary)

- **Security Intrusion**: A security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resource) without having authorization to do so.
- **Intrusion Detection**: A security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.

Taxonomy

- Architecture
  - Host-based
  - Network-based
    - inline vs. passive
  - Distributed
  - Hybrid
- Analysis technique
  - Misuse detection
  - Anomaly detection
- Time aspect
  - Real-time
  - Off-line
- Information source/type
  - System calls
  - System measurements and events
  - Network traffic
  - Application logs
  - Sensor alerts
  - File/directory access
  - Native vs. specific audit
- Detection vs. prevention

Taxonomy

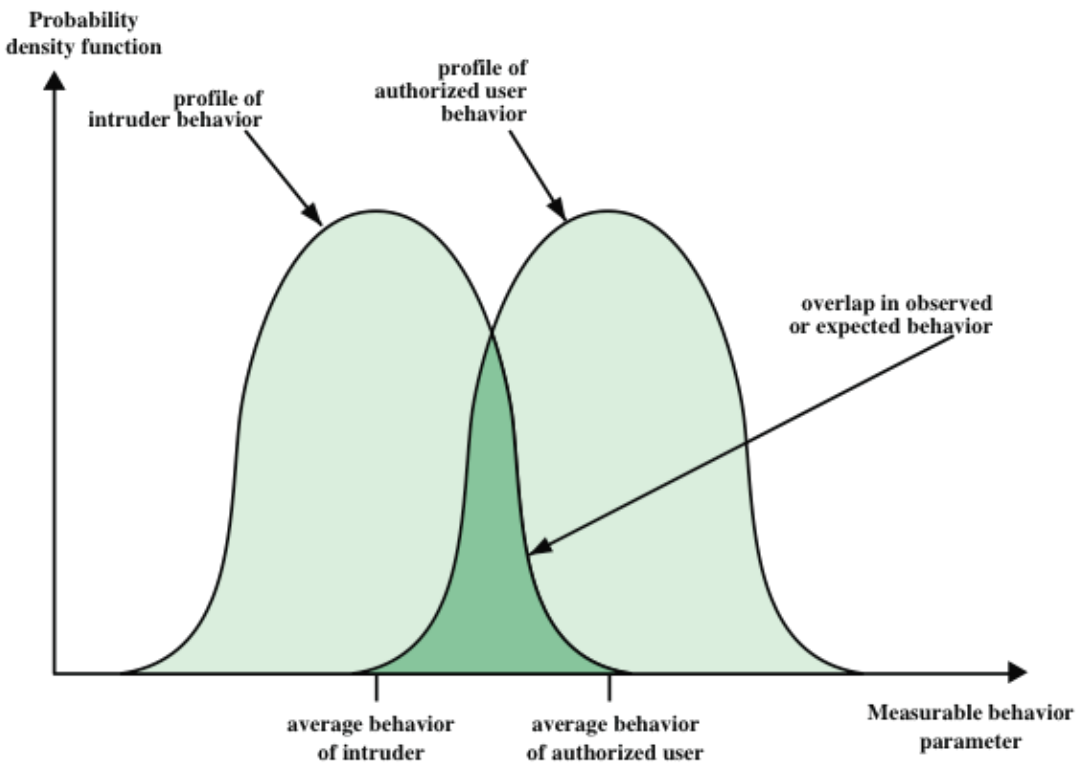
- Aggregation
  - Packet vs. session
  - Application vs. system
  - User
- Analysis technique
  - Statistics
  - Time series / state modeling
  - Association rules
  - Instance-based learning
  - Comparing graphs

Components

- sensors - collect data
- analyzers - determine if intrusion has occurred
- user interface - view output or control system behavior
- Intrusion detection exchange format (The Intrusion Detection Message Exchange Format (IDMEF) – RFC 4765):
  - "The purpose of the Intrusion Detection Working Group is to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems, and to management systems which may need to interact with them."

IDS Principles

- assume intruder behavior differs from legitimate users
- overlap in behaviors causes problems
- false positives
- false negatives



IDS Requirements

- run continually
- be fault tolerant
- resist subversion
- impose a minimal overhead on system
- configured according to system security policies
- adapt to changes in systems and users
- scale to monitor large numbers of systems
- provide graceful degradation of service
- allow dynamic reconfiguration

Measures that may be used for Intrusion Detection

Measure	Model	Type of Intrusion Detected
Login and Session Activity		
Login frequency by day and time	Mean and standard deviation	Intruders may be likely to log in during off-hours.
Frequency of login at different locations	Mean and standard deviation	Intruders may log in from a location that a particular user rarely or never uses.
Time since last login	Operational	Break-in on a "dead" account.
Elapsed time per session	Mean and standard deviation	Significant deviations might indicate masquerader.
Quantity of output to location	Mean and standard deviation	Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data.
Session resource utilization	Mean and standard deviation	Unusual processor or I/O levels could signal an intruder.
Password failures at login	Operational	Attempted break-in by password guessing.
Failures to login from specified terminals	Operational	Attempted break-in.
Command or Program Execution Activity		
Execution frequency	Mean and standard deviation	May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.
Program resource utilization	Mean and standard deviation	An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization.
Execution denials	Operational model	May detect penetration attempt by individual user who seeks higher privileges.
File Access Activity		
Read, write, create, delete frequency	Mean and standard deviation	Abnormalities for read and write access for individual users may signify masquerading or browsing.
Records read, written	Mean and standard deviation	Abnormality could signify an attempt to obtain sensitive data by inference and aggregation.
Failure count for read, write, create, delete	Operational	May detect users who persistently attempt to access unauthorized files.

Host-Based IPS

- identifies attacks using both:
  - signature techniques
    - malicious application packets
  - anomaly detection techniques
    - behavior patterns that indicate malware
- can be tailored to the specific platform
  - e.g. general purpose, web/database server specific
- can also sandbox applets to monitor behavior
- may give desktop file, registry, I/O protection

Network-Based IPS

- inline NIDS that can discard packets or terminate TCP connections
- uses signature and anomaly detection
- may provide flow data protection
  - monitoring full application flow content
- can identify malicious packets using:
  - pattern matching, stateful matching, protocol anomaly, traffic anomaly, statistical anomaly
- cf. SNORT inline can drop/modify packets

History of IDS

An Intrusion Detection Model [Dorothy Denning, 1983]

- First Intrusion Detection Expert System (IDES)
- Assumption – exploiting a system's vulnerability involves abnormal use of the system
- Intrusion detection model
  - Subjects – initiate actions in the target system; user, process, system, groups
  - Objects – files, programs, messages, records, printers,…
  - Audit records
  - Profiles
  - Anomaly records
  - Activity rules
- Monitoring standard operations: logins, commands, program executions, file/device access etc.

History of IDS

An Intrusion Detection Model [Dorothy Denning, 1983]

- Audit records
  - 6-tuples representing actions performed by subject on object
  <subject, action, object, exception-condition, resource-usage, timestamp>
  - action – login, read, execute…
  - exception-condition – raised by the system (e.g. write violation)
  - resource-usage – CPU, memory…

- Profiles – using metrics such as: event counter, between interval timer, resource measure
  - operational model (e.g. threshold)
  - confidence interval
  - multivariate model (e.g. correlations)
  - state transitions (e.g. Markov models)
  - time series

```
(Smith, execute, <Library>COPY.EXE, 0,
       CPU=00002, 11058521678)
(Smith, read,    <Smith>GAME.EXE,   0,
        RECORDS=0, 11058521679)
(Smith, write,   <Library>GAME.EXE, write-viol,
        RECORDS=0, 11058521680)
```

History of IDS

An Intrusion Detection Model [Dorothy Denning, 1983]

- Profile – contains info that identify the statistical model of a random variable
  - Types of profiles: login frequency, location frequency, last login, session time, execution frequency, program resources, exec denied, read/write frequencies…

```
Variable-Name:            SessionOutput
Action-Pattern:           'logout'
Exception-Pattern:        0
Resource-Usage-Pattern:   'SessionOutput=' # → amount
Period:
Variable-Type:            ResourceByActivity
Threshold:                4
Subject-Pattern:          'Smith'
Object-Pattern:           *
Value:                    record of ...
```

Additional IDSs in literature

- Learning patterns from Unix process execution traces
  [Lee and Stolfo, 1997]
  - extracting system calls (sliding window) from normal and abnormal executions

System Call Sequences (length 7)	Class Labels
4 2 66 66 4 138 66	"normal"
…	…
5 5 5 4 59 105 104	"abnormal"
…	…

Table 2. Pre-processed System Call Data. System call sequences of length 7 are labeled as "normal" or "abnormal".

normal:- p_2=104, p_7=112.
[meaning: if p_2 is 104 (*vtimes*) and p_7 is 112 (*vtrace*) then the sequence is "normal"]

normal:- p_2=104, p_7=104.
[meaning: if p_2 and p_7 are 104 (*vtimes*) then the sequence is "normal"]
…

abnormal:- true.
[meaning: if none of the above, the sequence is "abnormal"]

Additional IDSs in literature

- ADAM: Detecting Intrusions by Data Mining *[Barbara et al. 2001]*
  - Combination of Association Rule and Classification
  - – Uses connections as the basic granule

- USTAT *[1992]* and NETSTAT *[1998]*
  - – real-time intrusion detection system for UNIX

USTAT Action	SunOS Event Type
Read	open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt
Write	truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct
Create	mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod
Delete	rmdir, unlink
Execute	exec, execve
Exit	exit
Modify_Owner	chown, fchown
Modify_Perm	chmod, fchmod
Rename	rename
Hardlink	link

Distributed Host-Based IDS

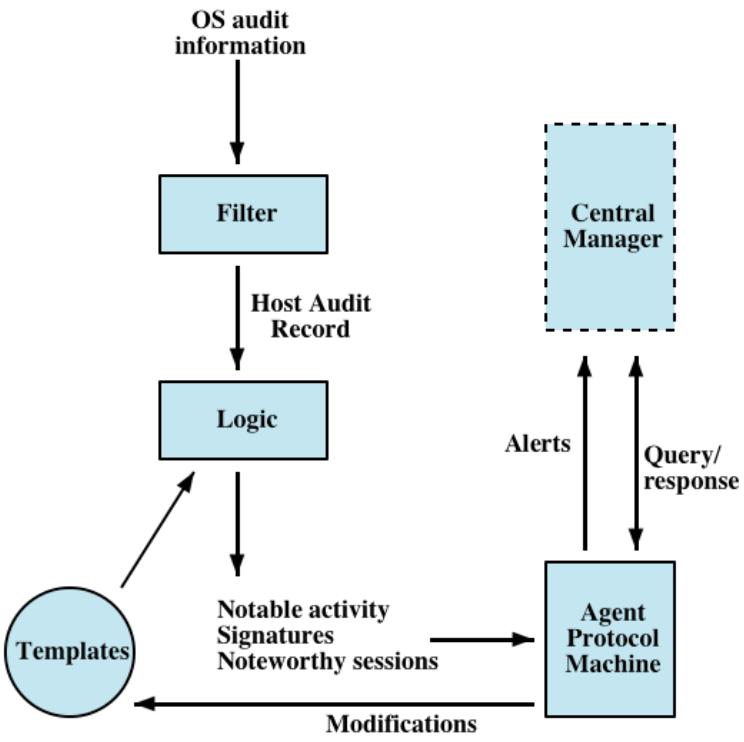


Figure 8.3 Agent Architecture

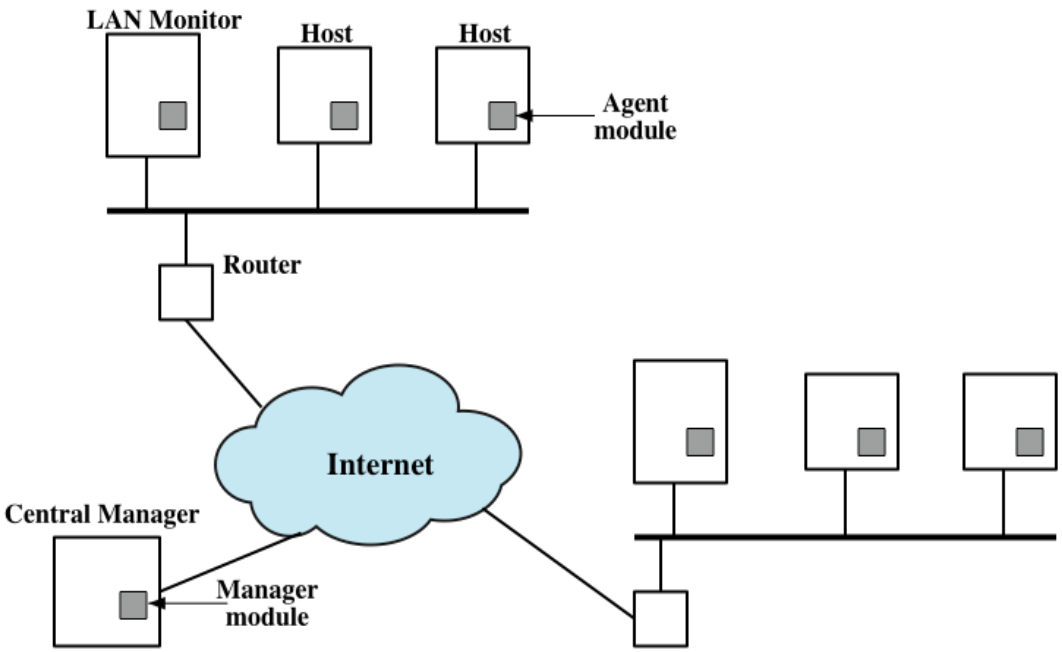


Figure 8.2 Architecture for Distributed Intrusion Detection

NIDS Deployment

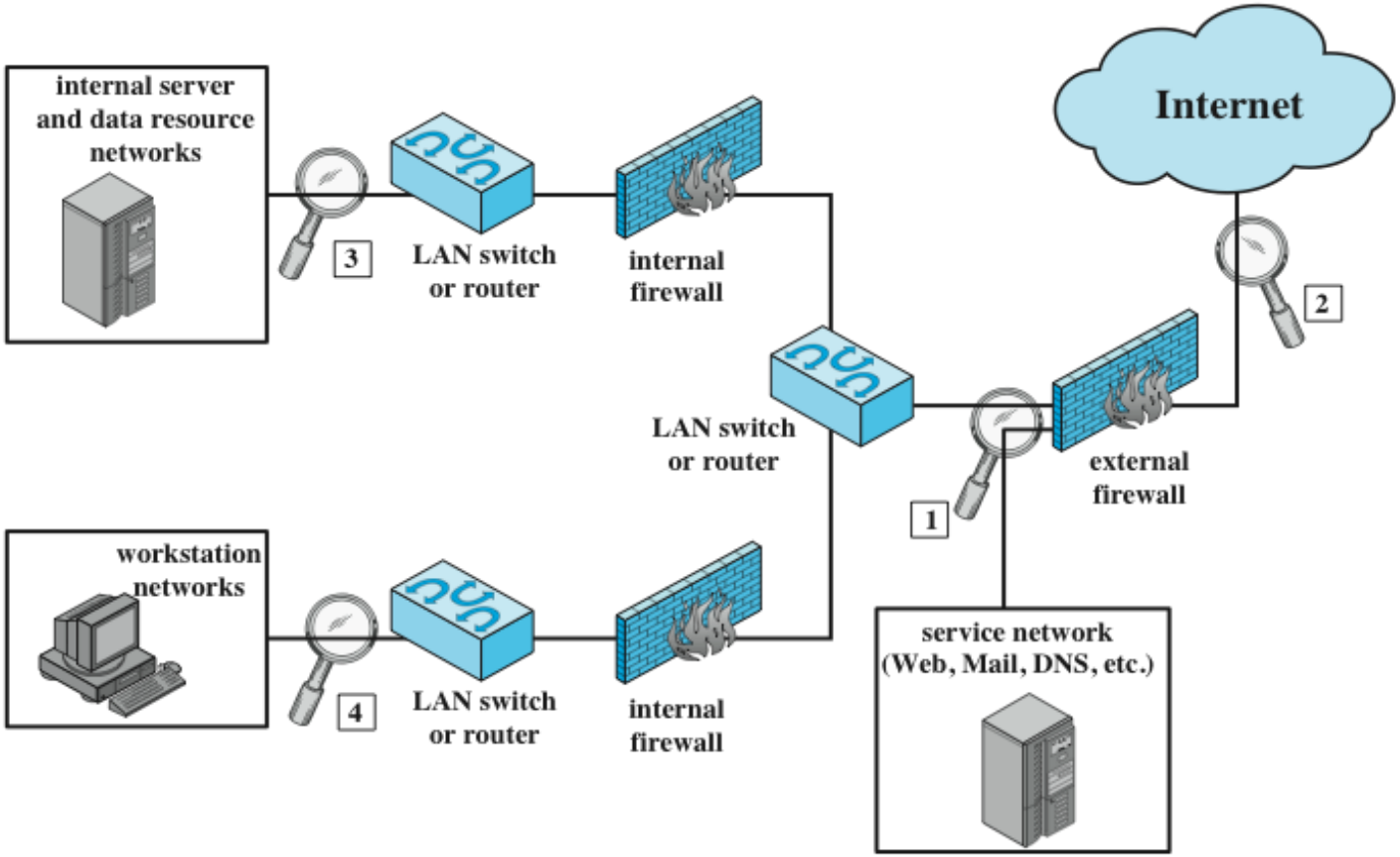


Figure 8.5 Example of NIDS Sensor Deployment

PAYL - payload-based NIDS

Wang, Stolfo 2004

- PAYL: a packet payload-based anomaly detection sensor
- Proposed efficient means of modeling normal payload data
- Basic assumption: new zero-day attack will have content data never before seen by the victim host
- Designed to detect and stop the very first occurrences of an attack that exhibits anomalous content
- Incremental update to changing or drifting environments

PAYL - payload-based NIDS

Length conditioned n-gram payload model

- Each application has its own special protocol and thus has its own payload type
  - SSH payload to port 22 should be encrypted ➔ uniform distribution
  - FTP payload to port 21 should be primarily printable characters entered by a user
  - Larger payloads are more likely to have non-printable characters (pictures, video clips or executable files etc.)

PAYL - payload-based NIDS

Length conditioned n-gram payload model

- For the payload of some inbound or outbound port, the feature vector is
  - Relative frequency count of each 1-gram (average frequency of each ASCII character 0-255)
  - Standard deviation of each frequency

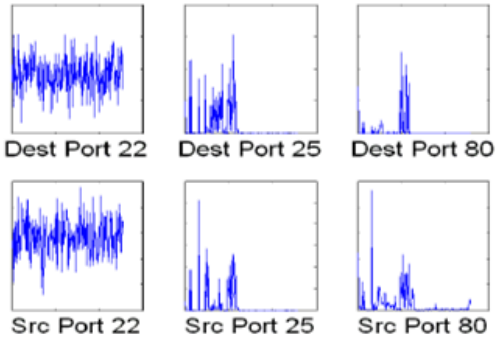


Figure 1. Example byte distributions[2] for different ports. For each plot, the X-axis is the ASCII byte 0-255, and the Y-axis is the average byte frequency.

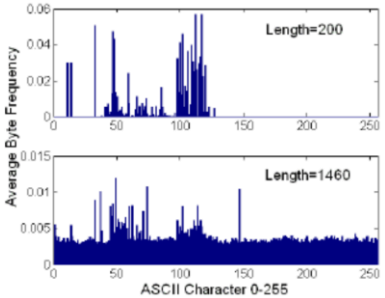


Figure 2. Example byte distribution for different payload lengths of port 80 on the same host server.

PAYL - payload-based NIDS

Length conditioned n-gram payload model

- **Training**
  - create model M_{ij} where avg byte frequency and the std of each byte's frequency (length *i;* port *j*)
- **Detection**
  - for each incoming payload compute its byte value distribution and compare against model M_{ij};
  - if the distribution is significantly different from the norm, the packet is flagged as anomalous and generates an alert

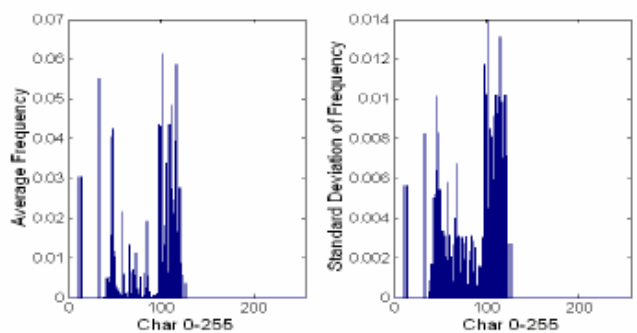
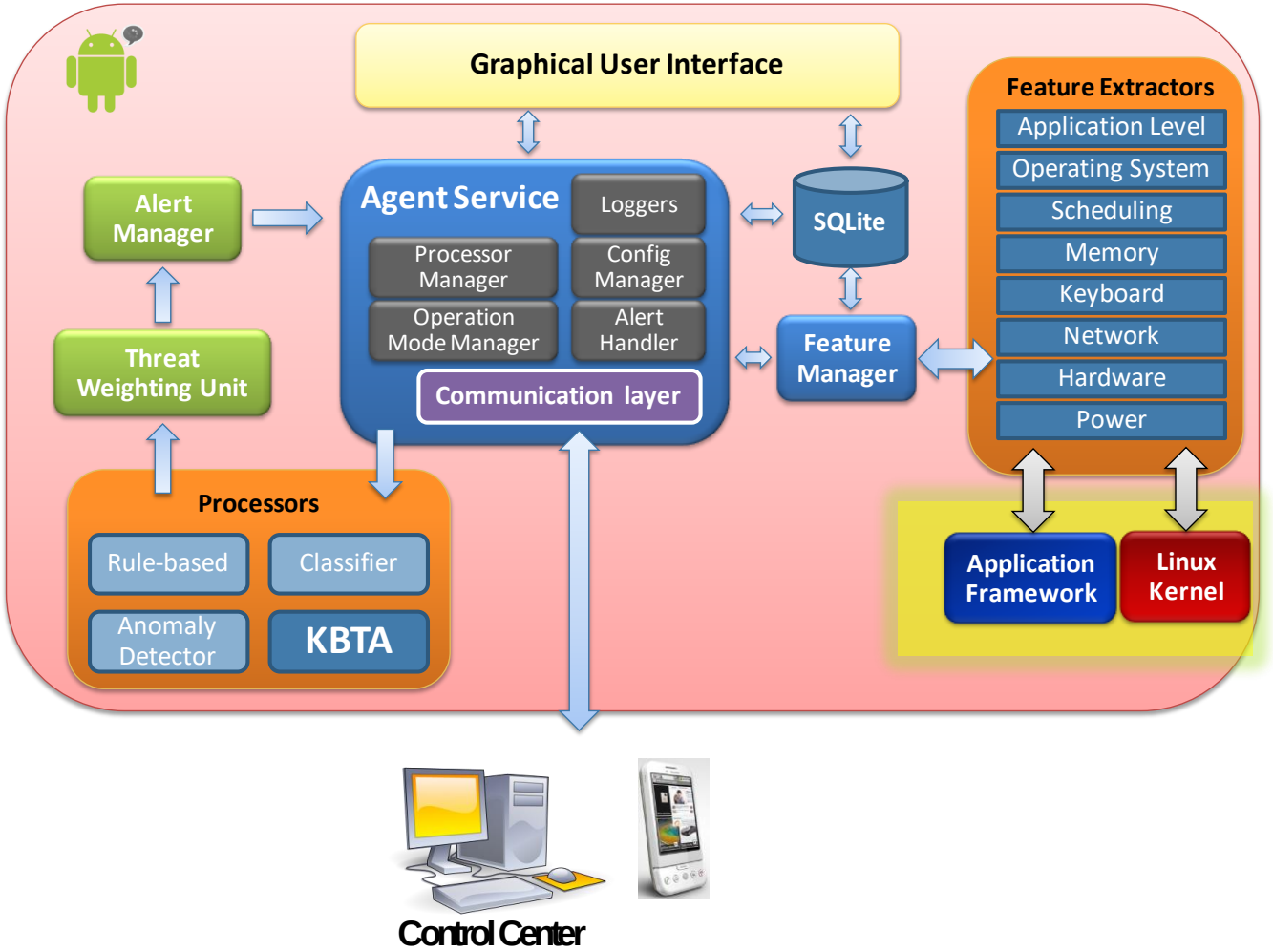


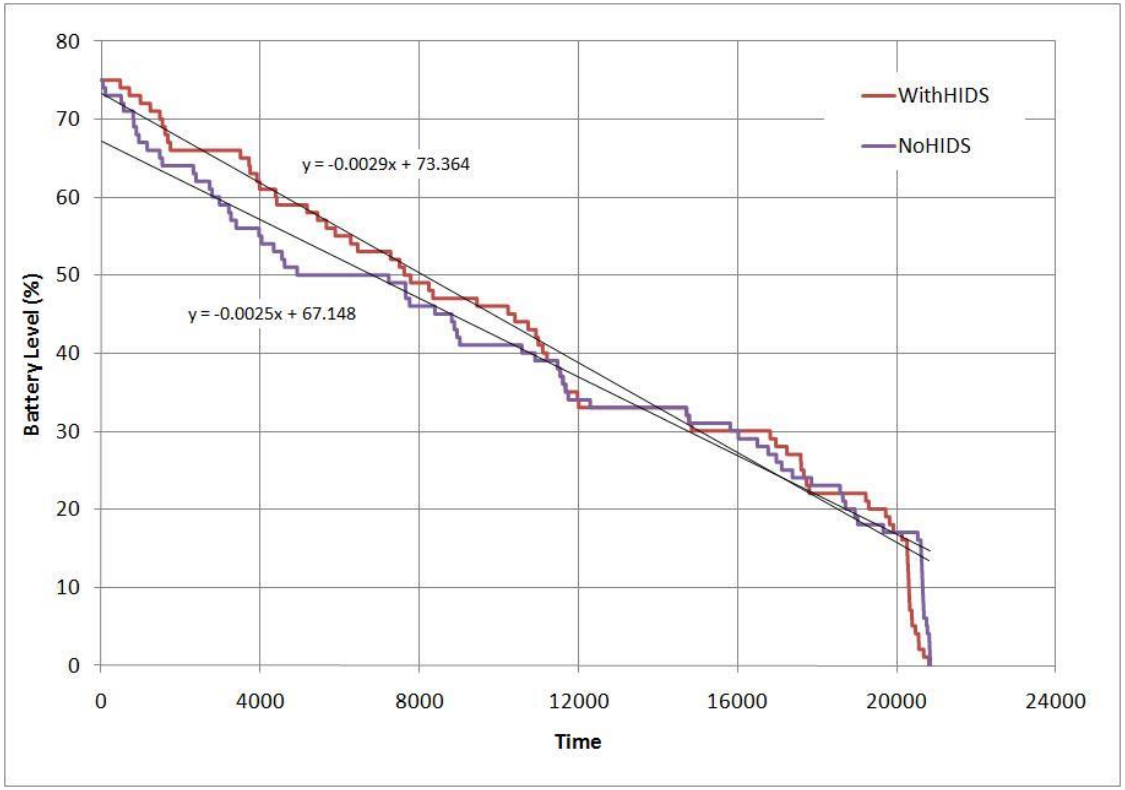
Figure 3. The average relative frequency of each byte, and the standard deviation of the frequency of each byte, for payload length 185 of port 80.

Monitoring smartphones



Monitoring smartphones

- Cost-sensitive feature selection



Anomaly-Based NIDS

- Traditional Approach (passive)
- Analyze features of the network's production traffic (background traffic)
  - Packets payload information
  - Network events
  - Network flows
  - Management Information Base (MIB)

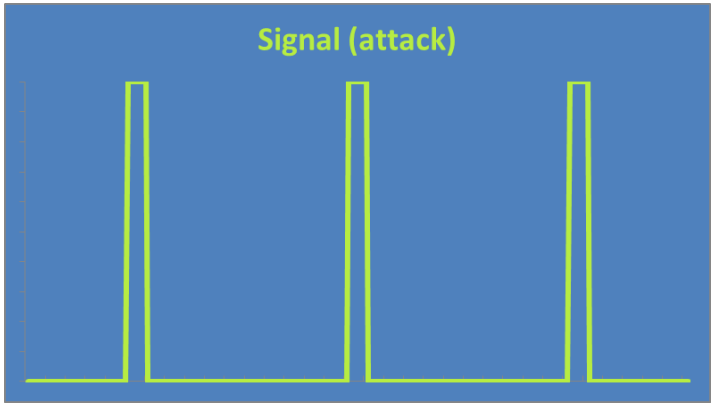
Passive NIDS usually end up having

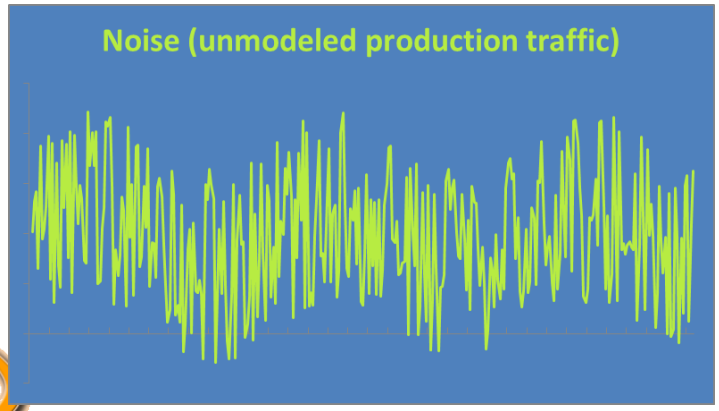
Low attack detection rate

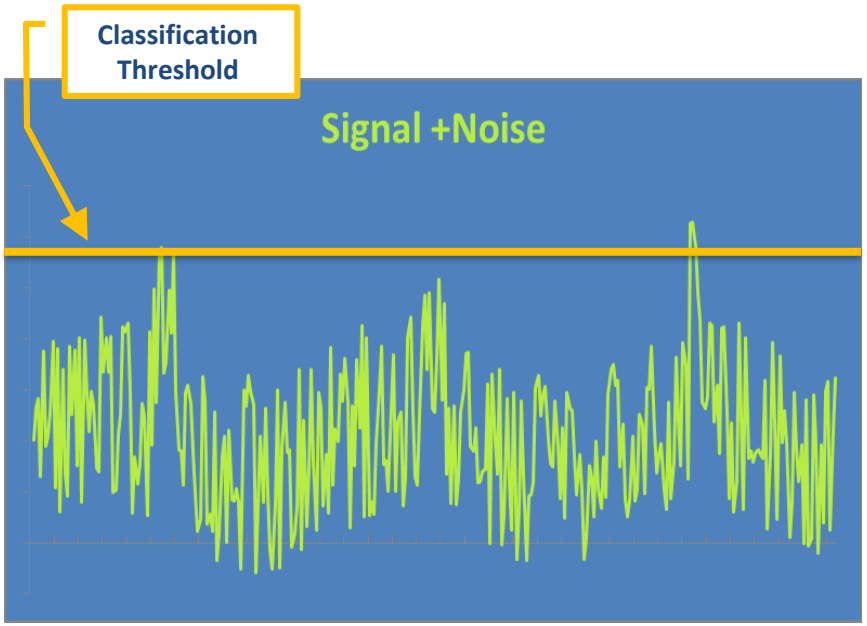
High false alarm rate

- Main disadvantages
- The production traffic varies immensely, even when the network is not under attack
- The NIDS has to process all the production traffic flowing through it

But Why? Answer: SNR!



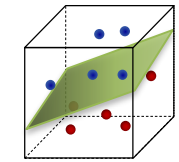




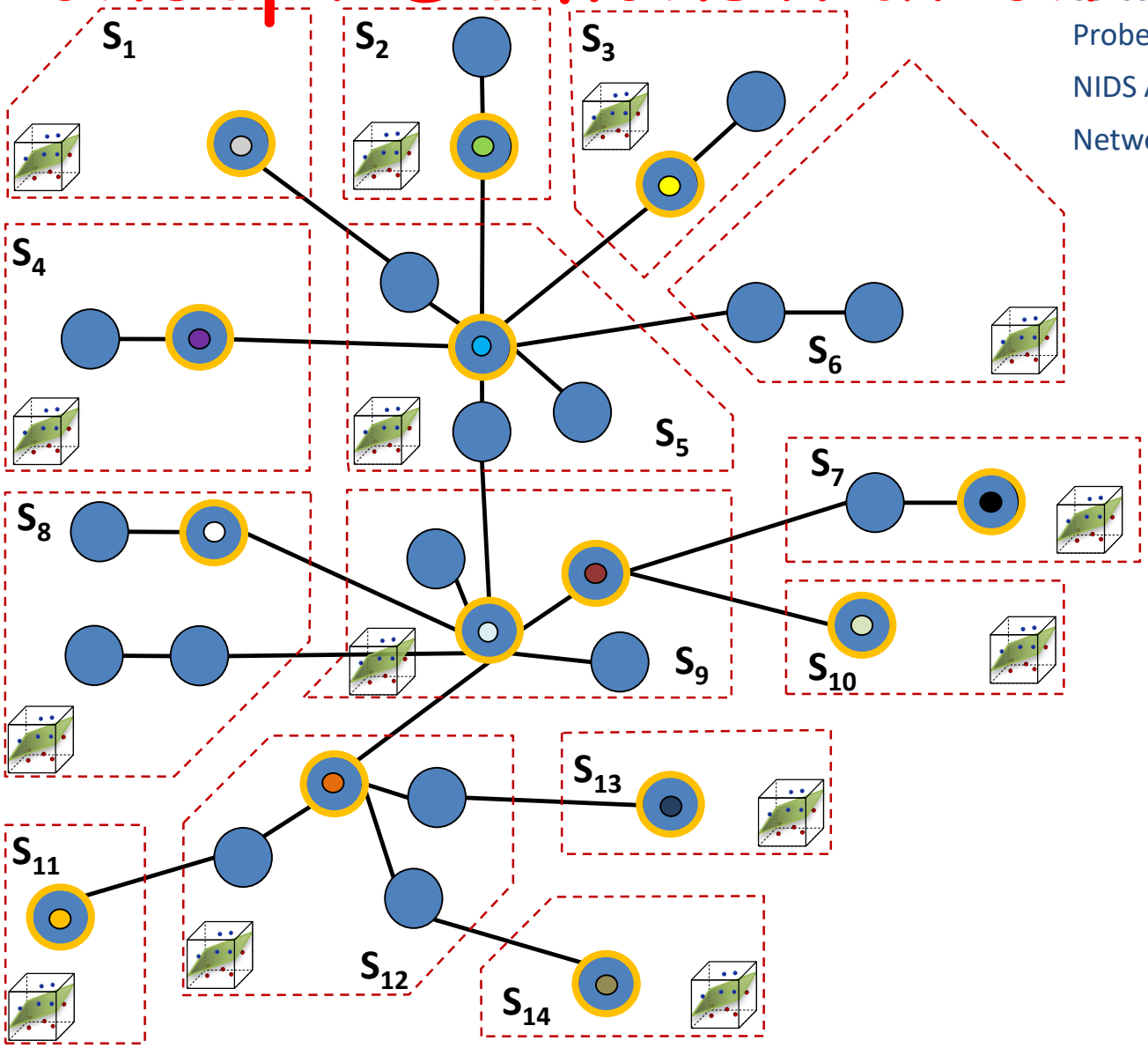
Active NIDS

- A system that detects anomalies using a self-traffic
  - The self traffic can be generated according to a known probability
- Main advantage over passive NIDS
- Self-traffic is characterized by a very low variability
  - Self-traffic features' have a very high S.N.R. (good)
  - Crucial for training a sound anomaly detection classifier

Concept Demonstration



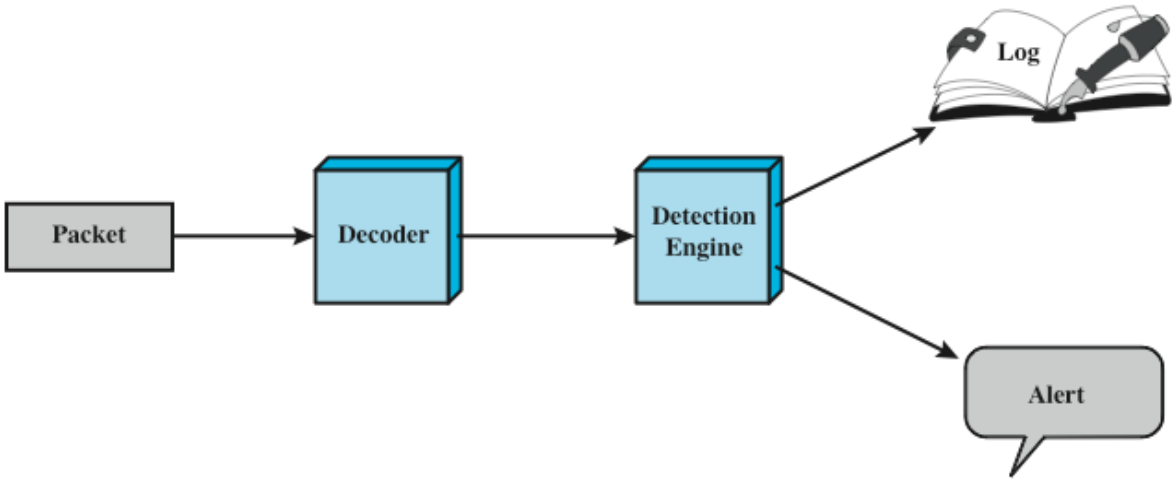
Central Detection Server



Classifier
Sensor
Probe Packet
NIDS Agent
Network Node

What's Snort?

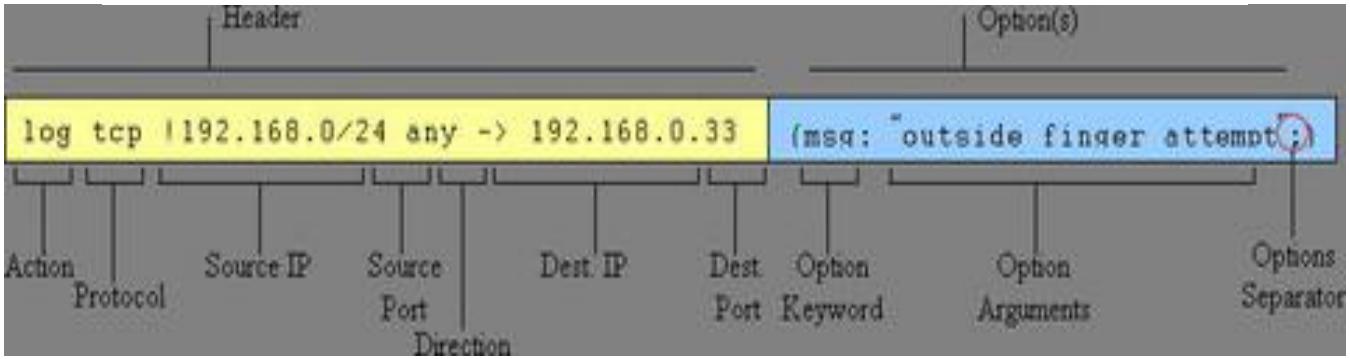
- Lightweight (in terms of memory and CPU) open source network intrusion detection system (NIDS)
- Real-time packet capture and rule analysis
- Easily deployed on nodes
- Uses a simple/flexible rule-based language
- Each rule consists of a fixed header and zero or more options



Snort rules

Action	Protocol	Address	Port	Direction	Address	Port

Action	Description
alert	Generate an alert using the selected alert method, and then log the packet.
log	Log the packet.
pass	Ignore the packet.
activate	Alert and then turn on another dynamic rule.
dynamic	Remain idle until activated by an activate rule , then act as a log rule.
drop	Make iptables drop the packet and log the packet.
reject	Make iptables drop the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.
sdrop	Make iptables drop the packet but does not log it.



Snort rule options

meta-data
msg Defines the message to be sent when a packet generates an event. **reference** Defines a link to an external attack identification system, which provides additional information. **classtype** Indicates what type of attack the packet attempted.
payload
content Enables Snort to perform a case-sensitive search for specific content (text and/or binary) in the packet payload. **depth** Specifies how far into a packet Snort should search for the specified pattern. Depth modifies the previous content keyword in the rule. **offset** Specifies where to start searching for a pattern within a packet. Offset modifies the previous content keyword in the rule. **nocase** Snort should look for the specific pattern, ignoring case. Nocase modifies the previous content keyword in the rule.
non-payload
ttl Check the IP time-to-live value. This option was intended for use in the detection of traceroute attempts. **id** Check the IP ID field for a specific value. Some tools (exploits, scanners and other odd programs) set this field specifically for various purposes, for example, the value 31337 is very popular with some hackers. **dsize** Test the packet payload size. This may be used to check for abnormally sized packets. In many cases, it is useful for detecting buffer overflows. **flags** Test the TCP flags for specified settings. **seq** Look for a specific TCP header sequence number. **icmp-id** Check for a specific ICMP ID value. This is useful because some covert channel programs use static ICMP fields when they communicate. This option was developed to detect the stacheldraht DDoS agent.
post-detection
logto Log packets matching the rule to the specified filename. **session** Extract user data from TCP Sessions. There are many cases where seeing what users are typing in telnet, rlogin, ftp, or even web sessions is very useful.

Rule examples

```
alert ip any any -> any any (msg: "IP Packet detected";)
```

```
alert tcp $TELNET_SERVERS 23 -> $EXTERNAL_NET any (msg:"TELNET
  Attempted SU from wrong group"; flow:
from_server,established; content:"to su root"; nocase;
  classtype:attempted-admin; sid:715; rev:6;)
```

One of the Main Challenges in Security – The "Base-Rate Fallacy" (The Base-Rate Fallacy and the Difficulty of Intrusion Detection, Axellson, 2000)

- In order for security applications to be "effective", they must maintain a very low false-positive rate
- In real-world scenarios, malicious code\packets are less than a fraction of 1% of the population
- In these cases, even a 99% classification rate might prove unacceptable…
- For example – a medical test which is 99% accurate for a disease that happens for 1 out of 100,000 people

One of the Main Challenges in Security – The "Base-Rate Fallacy" (The Base-Rate Fallacy and the Difficulty of Intrusion Detection, Axellson, 2000)

- In order for security applications to be "effective", they must maintain a very low false-positive rate
- In real-world scenarios, malicious code\packets are less than a fraction of 1% of the population
- In these cases, even a 99% classification rate might prove unacceptable…
- For example – a medical test which is 99% accurate for a disease that happens for 1 out of 100,000 people
  - Even if you tested positive, your chances of a TP < 1%
- This has serious implications for experiment design and algorithm performance