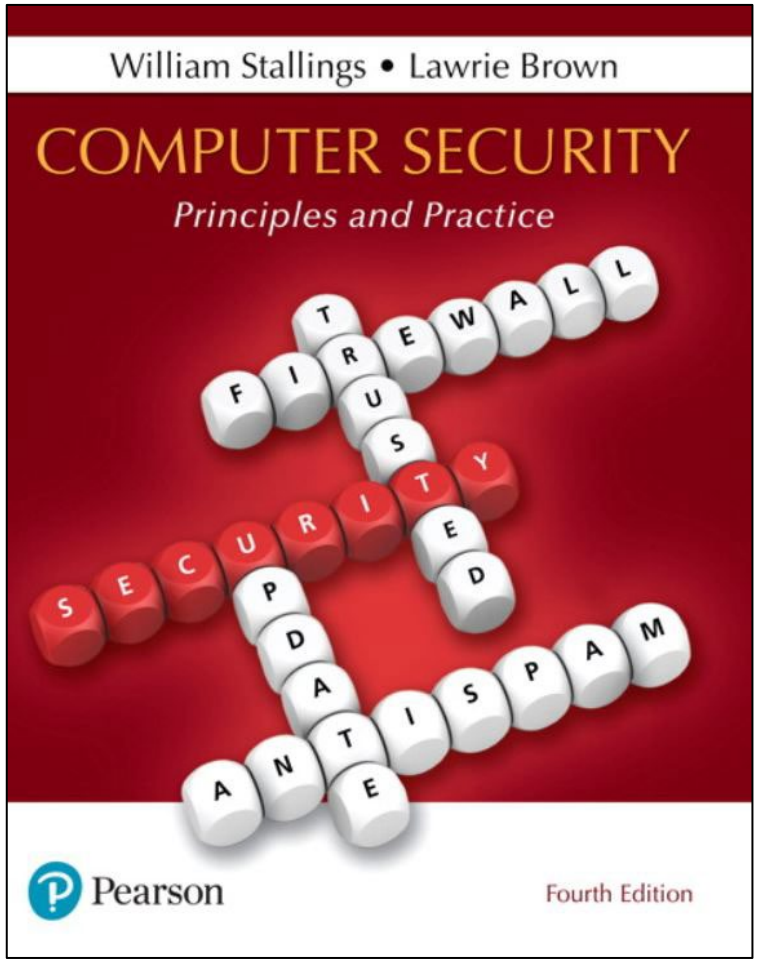
Intrusion Detection

CalPolyPomona

Computer Security: Principles and Practice



- **Chapter 8**

. Intrusion Detection

Intrusion and Intrusion Detection

- Security Intrusion:

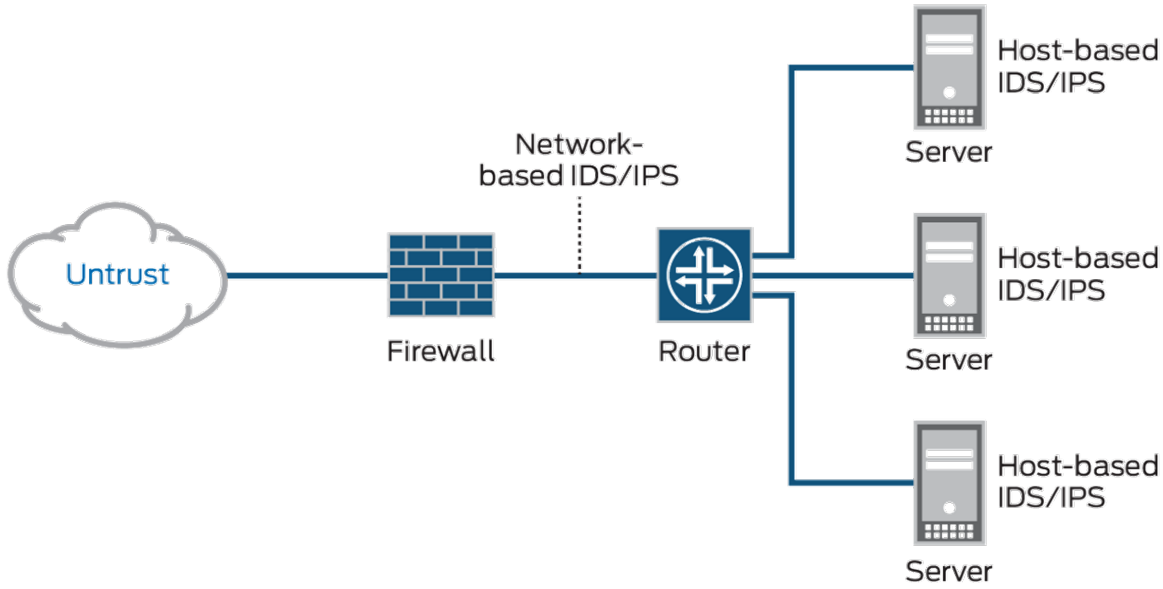
  Unauthorized act of bypassing the security mechanisms of a system

- Intrusion Detection:

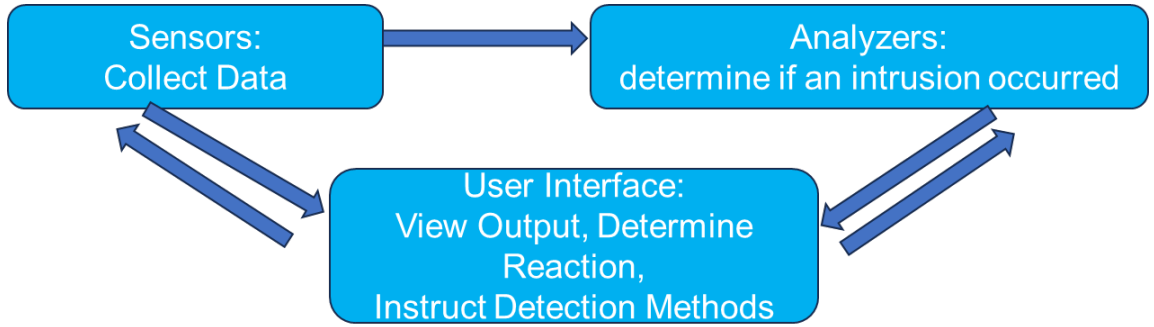
  A hardware or software function that gathers and analyzes information from various areas within a computer or a network to identify possible security intrusions

Intrusion Detection System (IDS)

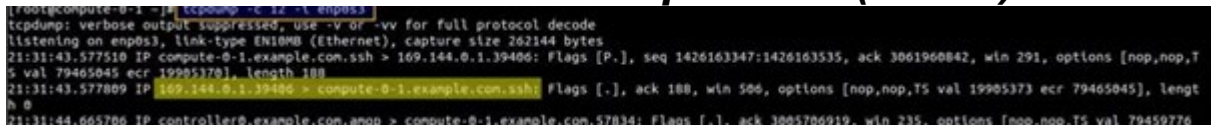
- **Host-based IDS (HIDS)**
  - Monitors the characteristics of a single host for suspicious activity
- **Network-based IDS (NIDS)**
  - Monitors network traffic and analyzes network, transport, and application protocols to identify suspicious activity
- **Distributed or hybrid IDS**
  - Combines information from a number of sensors, often both host and network based, in a central analyzer that is able to better identify and respond to intrusion activity



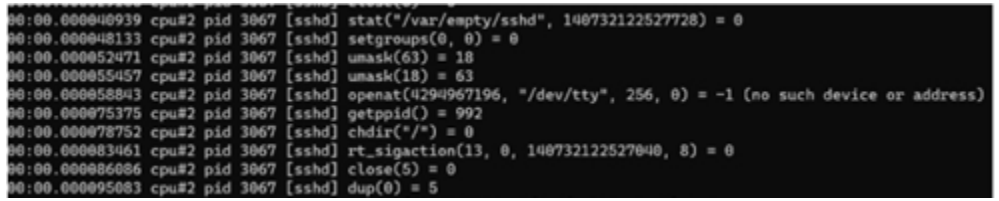
Components of an IDS and data sources



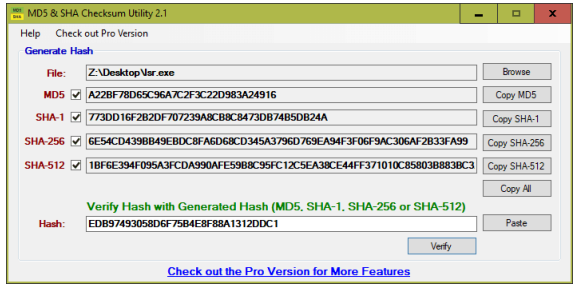
Sensor data: Network packets (N-IDS)



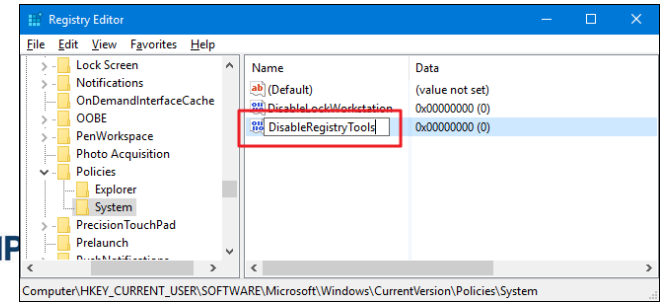
Sensor data: System call trace (H-IDS)



Sensor Data: File Checksum (H-IDS)

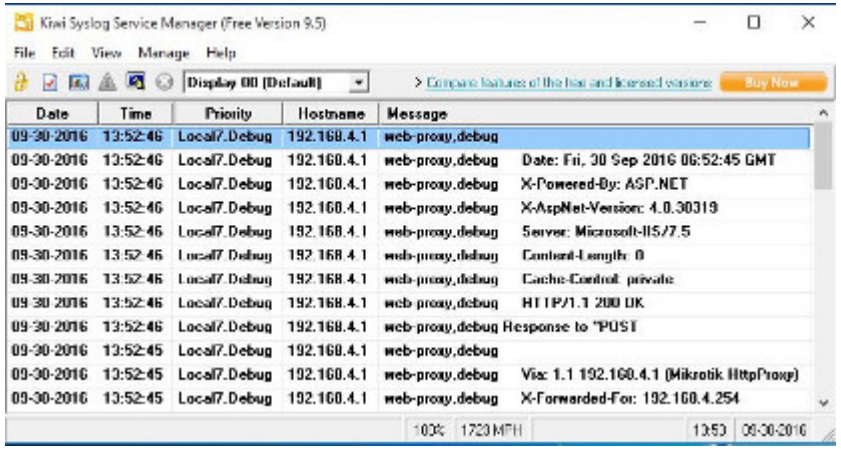


Sensor Data: Win. Registry access events (H-IDS)



Each type of sensor data contains whatever the administrator has configured to monitor, store and/or send to a collector/analyzer. Note: they have a cost as resources required to generate, analyze and send

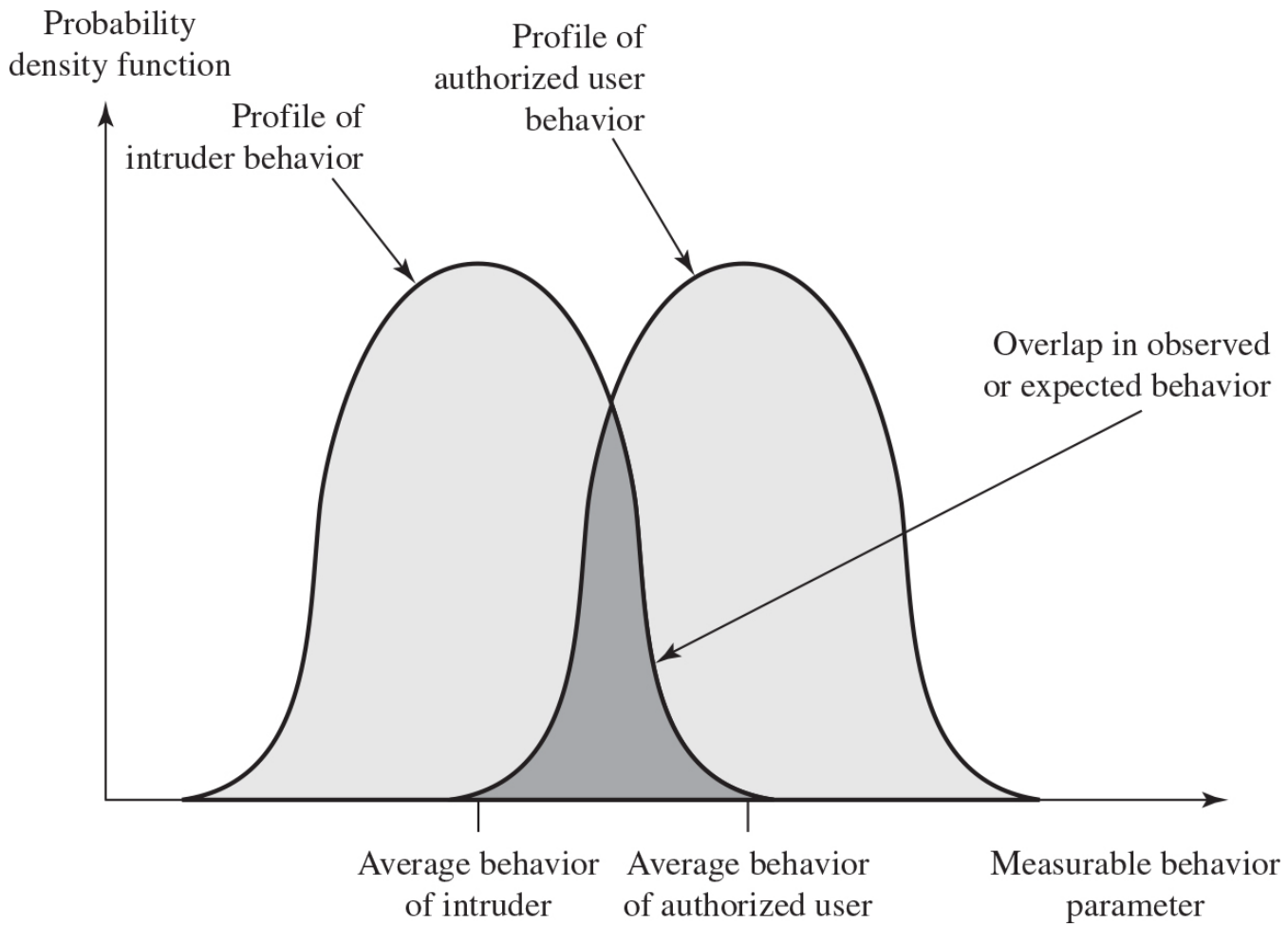
Sensor data: log messages (H-IDS)



Profiles of Behavior of Intruders and Authorized Users

Why is it so hard to detect an intrusion?

Because, except for the cases when Indicators of Compromise are known and detected (malicious known IPs, malicious domains, malware hashes, heuristic rules, etc.) The behavior of an intruder has many characteristics that correspond to the average user behavior -> it's never a sharp difference



General approaches for any IDS (host/network/hybrid)

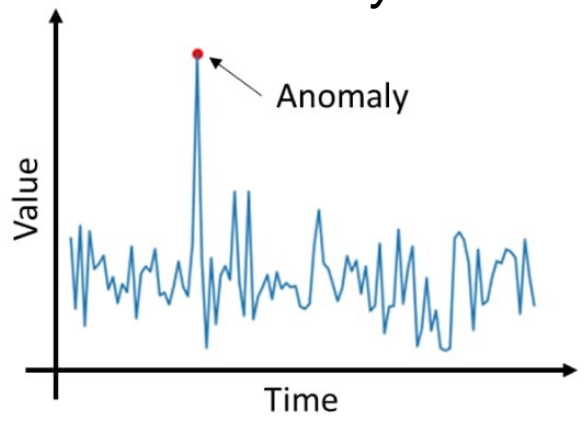
Anomaly detection

- Involves the collection of data relating to the behavior of legitimate users over a period of time
- Current observed behavior is analyzed to determine whether this behavior is that of a legitimate user or that of an intruder

Signature/Heuristic detection

- Uses a set of known malicious data patterns or attack rules that are compared with current behavior
- Also known as misuse detection
- Can only identify known attacks for which it has patterns or rules

Univariate outlier anomaly detection



Example: Rule for detection of risky command

Rule Example: Detecting Suspicious PowerShell Commands

plaintext Copy code

```
Rule Name: PowerShellSuspiciousCommands
Description: Detects suspicious PowerShell commands indicative of malicious activity.
Detection Type: Signature

Rule:
  Event Type: Command Execution
  Application: PowerShell.exe
  Condition: CommandLine matches regex "(^|\s)([A-Za-z]:[\\/][^\\/:*?\"<>|]+\.ps1)(
  Severity: High
  Action: Alert
```

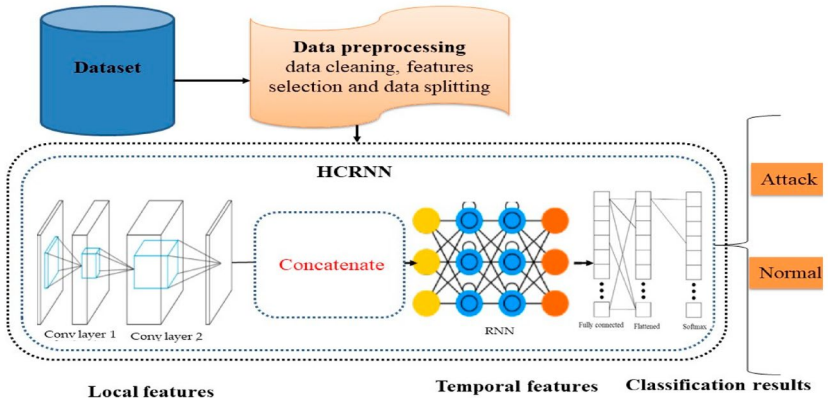
Example: Heuristic Rule for process spawning

Rule Name: Heuristic_ProcessSpawning
Description: Detects unusual patterns of process spawning.
Detection Type: Heuristic

Rule:
Event Type: Process Creation
Condition:
- Number of processes spawned by a parent process within a short time frame is significantly higher than normal.
- Executable files spawned from non-standard locations (e.g., temporary folders).
Severity: High
Action: Terminate Process and Alert

Anomaly Detection

- A variety of classification approaches are used:
  - **Statistical**
    - Analysis of the observed behavior using univariate, multivariate, or time-series models of observed metrics
  - **Knowledge-based**
    - Approaches use an expert system that classifies observed behavior according to a set of rules that model legitimate behavior
  - **Machine-learning**
    - Approaches automatically determine a suitable classification model from the training data using data mining techniques



Signature or Heuristic Detection

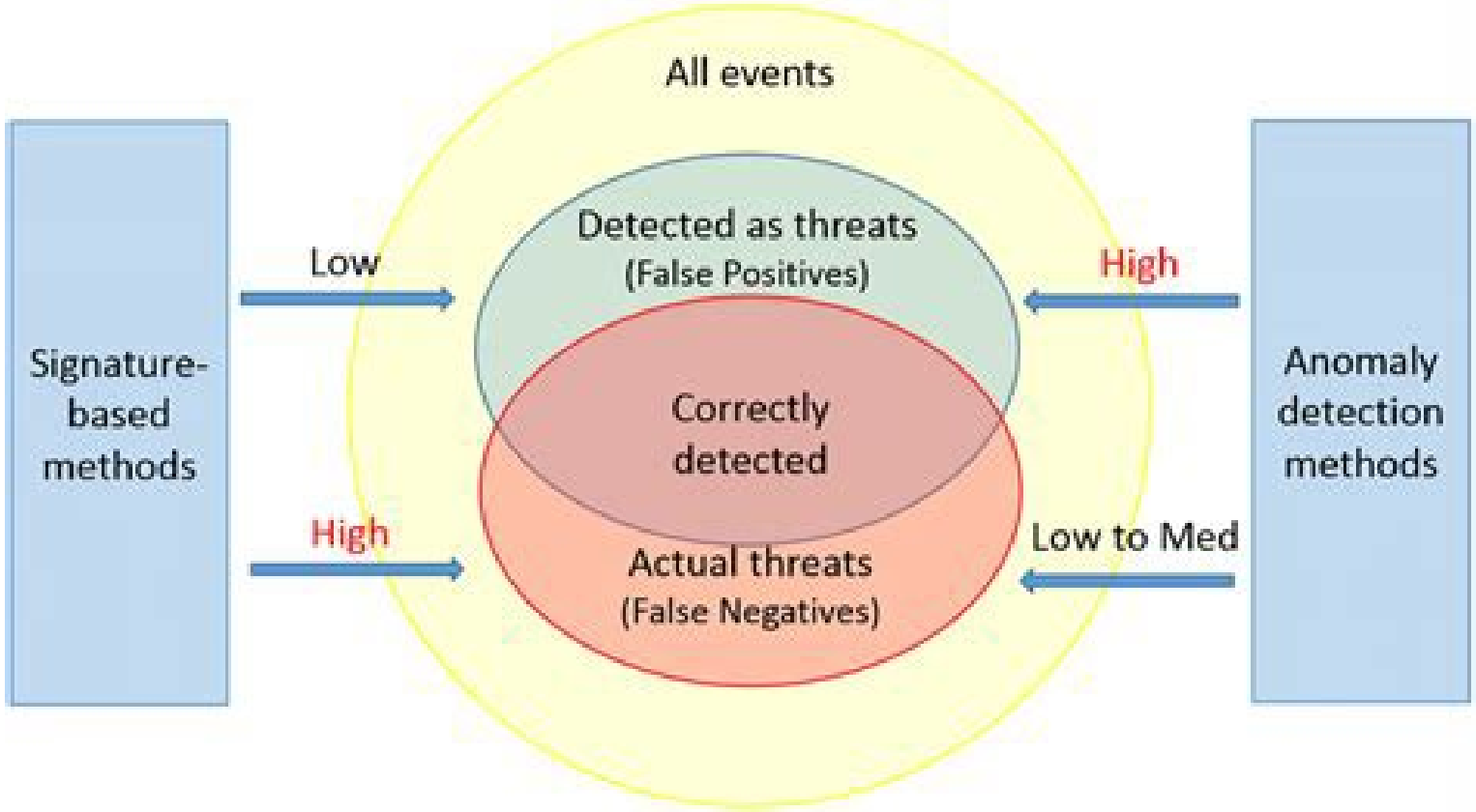
- **Signature approaches**
  - Match a large collection of known patterns of malicious data against data stored on a system or in transit over a network
  - The signatures need to be large enough to minimize the false alarm rate, while still detecting a sufficiently large fraction of malicious data
  - **Widely used in anti-virus products, network traffic scanning proxies**, and in **NIDS**
- **Rule-based heuristic identification**
  - Involves the use of rules for identifying known penetrations or penetrations that would exploit known weaknesses
  - Rules can also be defined that identify suspicious behavior, even when the behavior is within the bounds of established patterns of usage
  - Typically, rules used are specific
  - SNORT is an example of a rule-based NIDS

Snort rule with exact match of content

```
alert udp any any -> any 53 (content:"|01 00 00 01 00 00 00 00 00 01|"; offset:
2; depth: 10; content:"|00 00 29 10 00 00 00 80 00 00 00|";  \
msg: "covert iodine tunnel request"; threshold: type limit, track by_src, count
1, seconds 300; sid: 5619500; rev: 1;)

alert udp any 53 -> any any (content: "|84 00 00 01 00 01 00 00 00 00|"; offset:
2; depth: 10; content:"|00 00 0a 00 01|";  \
msg: "covert iodine tunnel response"; threshold: type limit, track by_src, count
1, seconds 300; sid: 5619501; rev: 1;)
```

Signature Vs Anomaly detection



Host-Based Intrusion Detection (HIDS)

- Adds a specialized layer of security software to vulnerable or sensitive systems
- Can use either anomaly or signature and heuristic approaches:
- Sources: System calls, File checksum, Usage heuristics, audit logs, registry access
- Monitors activity to detect suspicious behavior
  - Primary purpose is to detect intrusions, log suspicious events, and send alerts
  - Can detect both external and internal intrusions

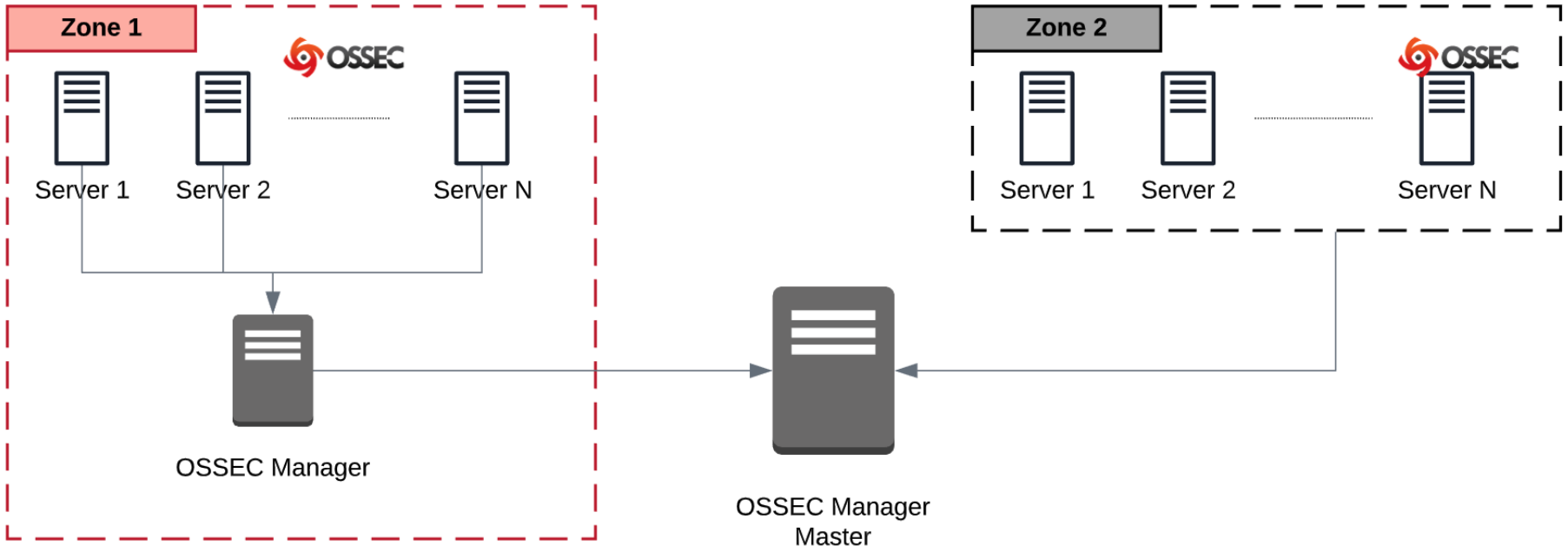
Ubuntu system calls

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

Windows system calls

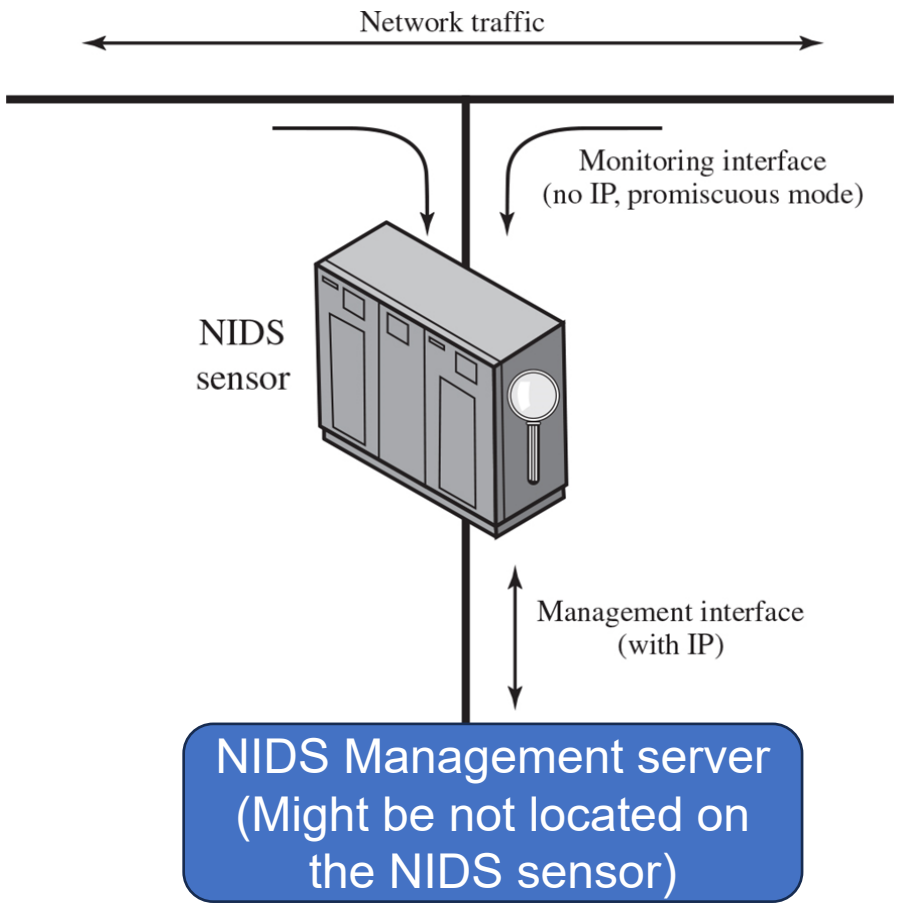
comctl32
kernel32
msvcpp
msvcrt
mswsock
ntdll
ntoskrnl
user32
ws2_32

Example of HIDS: OSSEC



Network-Based IDS (NIDS)

- Monitors traffic at selected points on a network
- Examines traffic packet by packet in real or close to real time
- May examine network, transport, and/or application-level protocol activity
- Comprised of a number of sensors, one or more servers for NIDS management functions, and one or more management consoles for the human interface
- Analysis of traffic patterns may be done at the sensor, the management server or a combination of the two



Example of NIDS Sensor Deployment

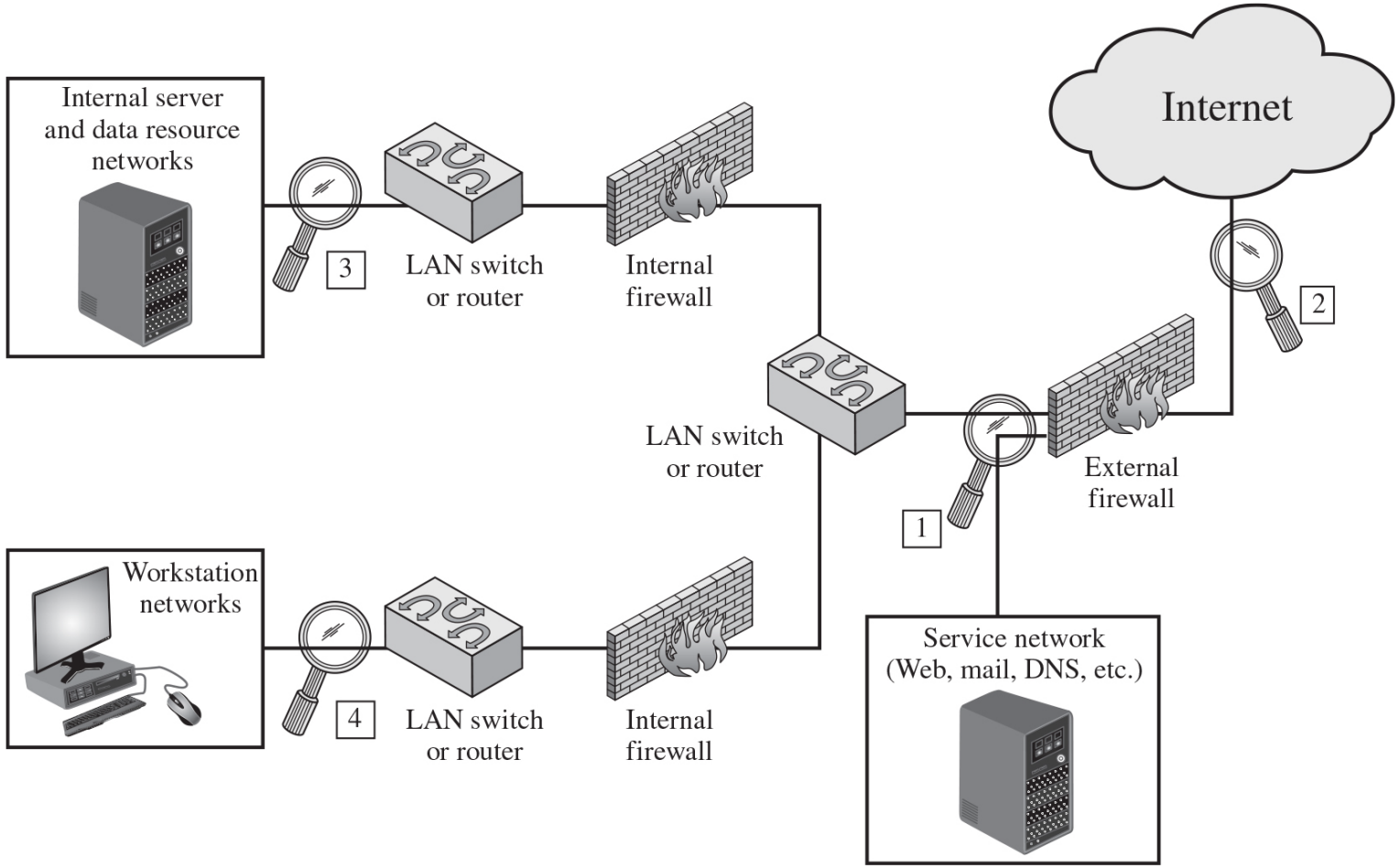


Figure 8.5

Intrusion Detection Techniques with NIDS

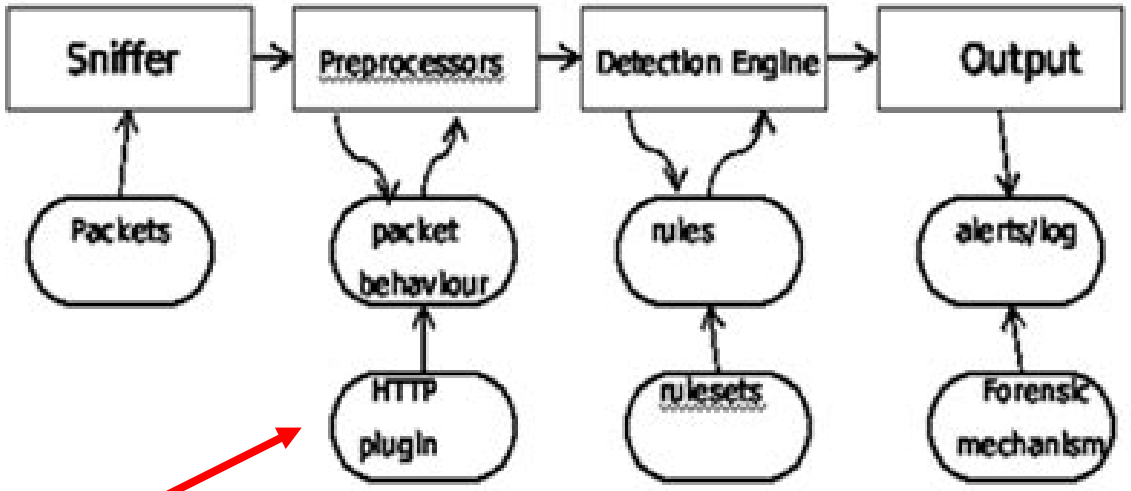
Attacks suitable for Signature detection

- Application layer reconnaissance and attacks
- Transport layer reconnaissance and attacks
- Network layer reconnaissance and attacks
- Unexpected application services
- Policy violations

Attacks suitable for Anomaly detection

- Denial-of-service (DoS) attacks
- Scanning
- Worms
- Zero-days

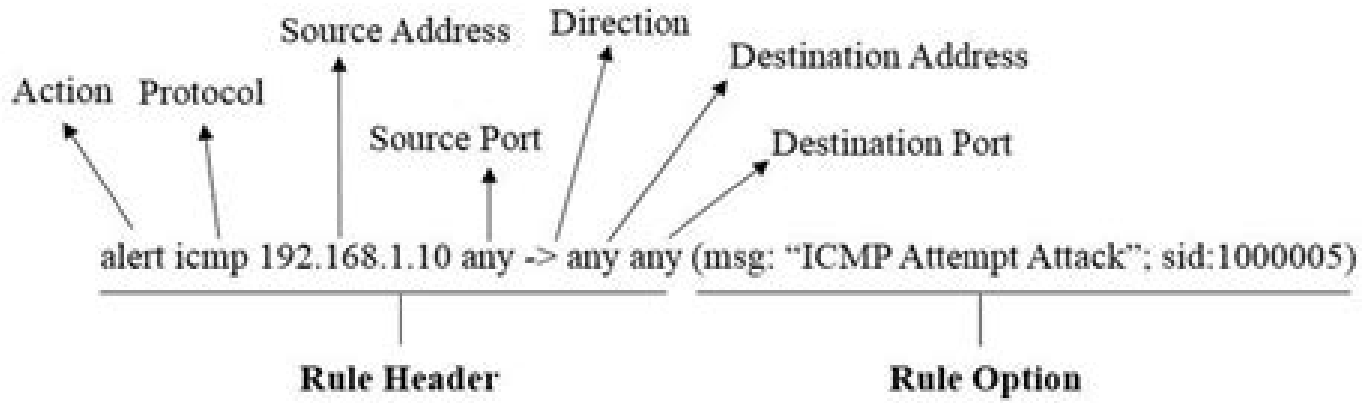
Example of NIDS: Snort



Generate logs that can be stored for forensic analysis or for real-time processing with a SIEM

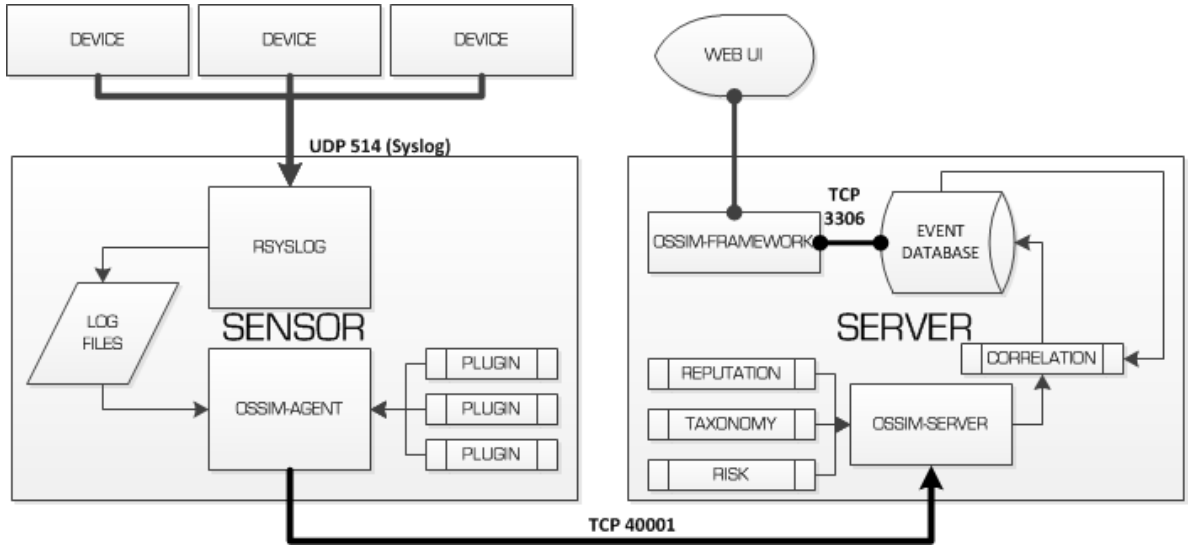
Example of plugin:
reconstruct
HTTP messages from
TCP segments to check for
known signatures of attacks

Snort Rule



Security Information and Event Management (SIEM)

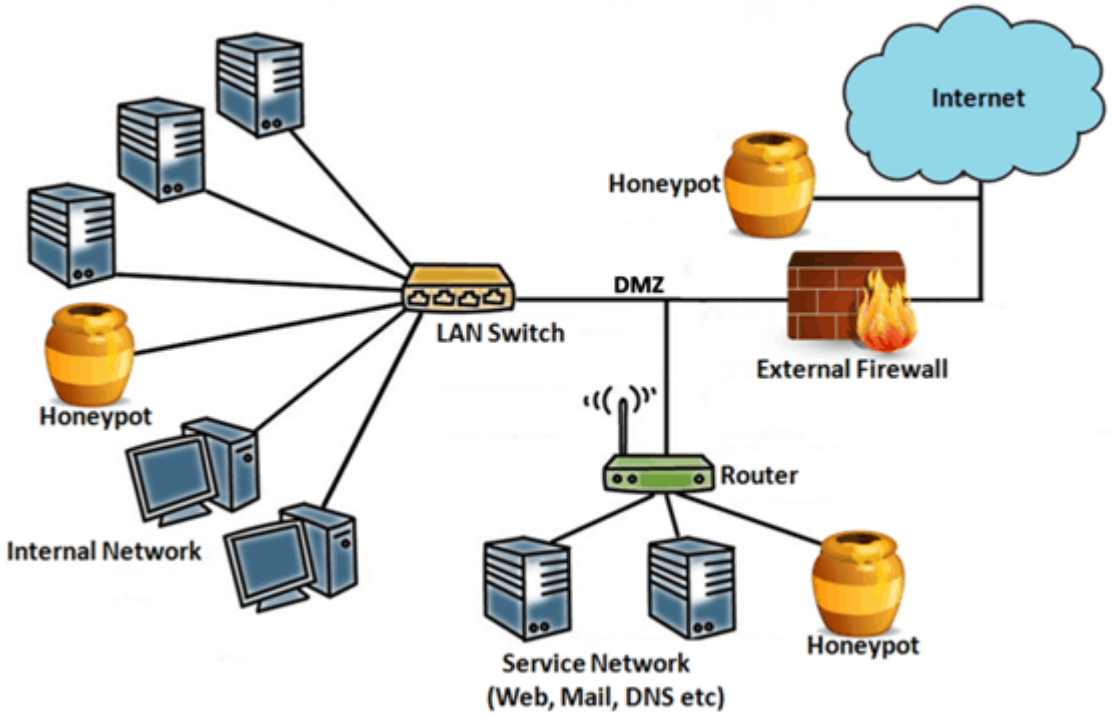
- A Security Information and Event Management (SIEM) system is a distributed system that collects, correlates, analyze and stores logs (aka, *events*) and metrics from various sensors to prevent or detect intrusions
- ***Sensors*** are sometime called ***probes***, and can be NIDS, HIDS, antivirus, firewalls, and any source of information useful to spot an attack (mostly, generating logs collected by the SIEM)



OSSIM SIEM architecture

Honeypot

- Decoy systems designed to:
  - Lure a potential attacker away from critical systems
  - Collect information about the attacker's activity
  - Encourage the attacker to stay on the system long enough for administrators to respond
- Systems are filled with fabricated information that a legitimate user of the system wouldn't access
- To be effective in the study of the attackers, a honeypot needs to be *instrumented* as much as possible with any kinds of sensors and probes
- Two categories of Honeypot:
  - **Low interaction honeypot:** Consists of a software package that emulates particular IT services or systems well enough to provide a realistic initial interaction but does not execute a full version of those services or systems.
  - **High interaction honeypot:** Is a real system, with a full operating system, services and applications, which are instrumented and deployed where they can be accessed by attackers.



Example:
https://www.dshield.org/honeypot.html
Installation on Rpi:
https://medium.com/swlh/installing-dshield-honeypot-on-a-raspberry-pi-e10d967825b2