CSE508 Network Security

Stony Brook University

2021-04-15

Intrusion Detection

Michalis Polychronakis

Stony Brook University


* Welcome to CityPower Grid Rerouting *
Authorised Users only!
New users MUST notify Sys/Ops.
login:
EDITV1 sshnuke
rcr ebx, 1
bsr ecx, ecx
shrd ebx, edi, CL
[mobile]
nmap -v -sS -O 10.2.2.2
Starting nmap V. 2.54BETA25
Insufficient responses for TCP sequencing (3), OS detection may be less
accurate
Interesting ports on 10.2.2.2:
(The 1539 ports scanned but not shown below are in state: closed)
Port State Service
22/tcp open ssh
No exact OS matches for host
Nmap run completed -- 1 IP address (1 host up) scanned
sshnuke 10.2.2.2 -rootpw="Z1ON0101"
Connecting to 10.2.2.2:ssh ... successful.
Attempting to exploit SSHv1 CRC32 ... successful.
Reseting root password to "Z1ON0101".
System open: Access Level <9>
ssh 10.2.2.2 -l root
root@10.2.2.2's password:
RIF CONTROL
ACCESS GRANTED

Intrusions

"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 system

Attack Source

Local

- Unprivileged access ➔ privilege escalation
- Physical access: I/O ports (launch exploits), memory (cold boot attacks), storage (just remove it), shoulder surfing (steal credentials), dumpster diving (steal information), bugging (e.g., keylogger, antennas/cameras/sensors, HW parts), …

Remote

- Internet
- Local network (Ethernet, WiFi, cellular, bluetooth, …)
- Phone (social engineering, SMS, …)
- Infected media (~~disks~~, ~~CD-ROMs~~, USB sticks, …)
- Pre-infected components/devices/software (BIOS, NIC, router, updates, …)

Intrusion Method

Social engineering (phishing, spam, scareware, …)

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

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

Password guessing/leakage (brute force, root:12345678, …)

Physical access (reboot, keylogger, screwdriver, …)

Supply chain compromise (backdoor, infected update, …)

Software vulnerability exploitation

…

Attack Outcome

Arbitrary code execution

Privilege escalation

Disclosure of confidential information

Unauthorized access

DoS

Erroneous output

Destruction

…

Intrusion Detection

Intrusion detection systems (IDS) monitor networks or hosts for signs of malicious activity or policy violations

Detection (IDS)

just generate alerts and log any identified events

Prevention (IPS)

in addition, react by blocking the detected activity

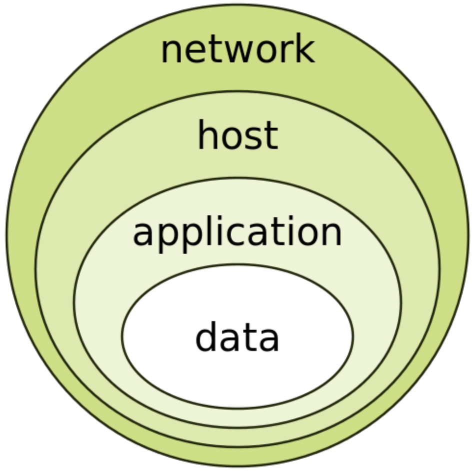
Defense in Depth

An IDS is not a silver bullet solution

- Just an additional layer of defense
- Complements existing protections, detectors, and policy enforcement mechanisms
- Requires continuous maintenance: fine-tune configuration, adapt to network changes, update rules, triage alerts, minimize false positives, …

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, cyber-physical systems, businesses/enterprises with inadequate resources, …

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

- Immediate response vs. long-term treatment
- Patches for 0-day exploits take time to develop and deploy

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

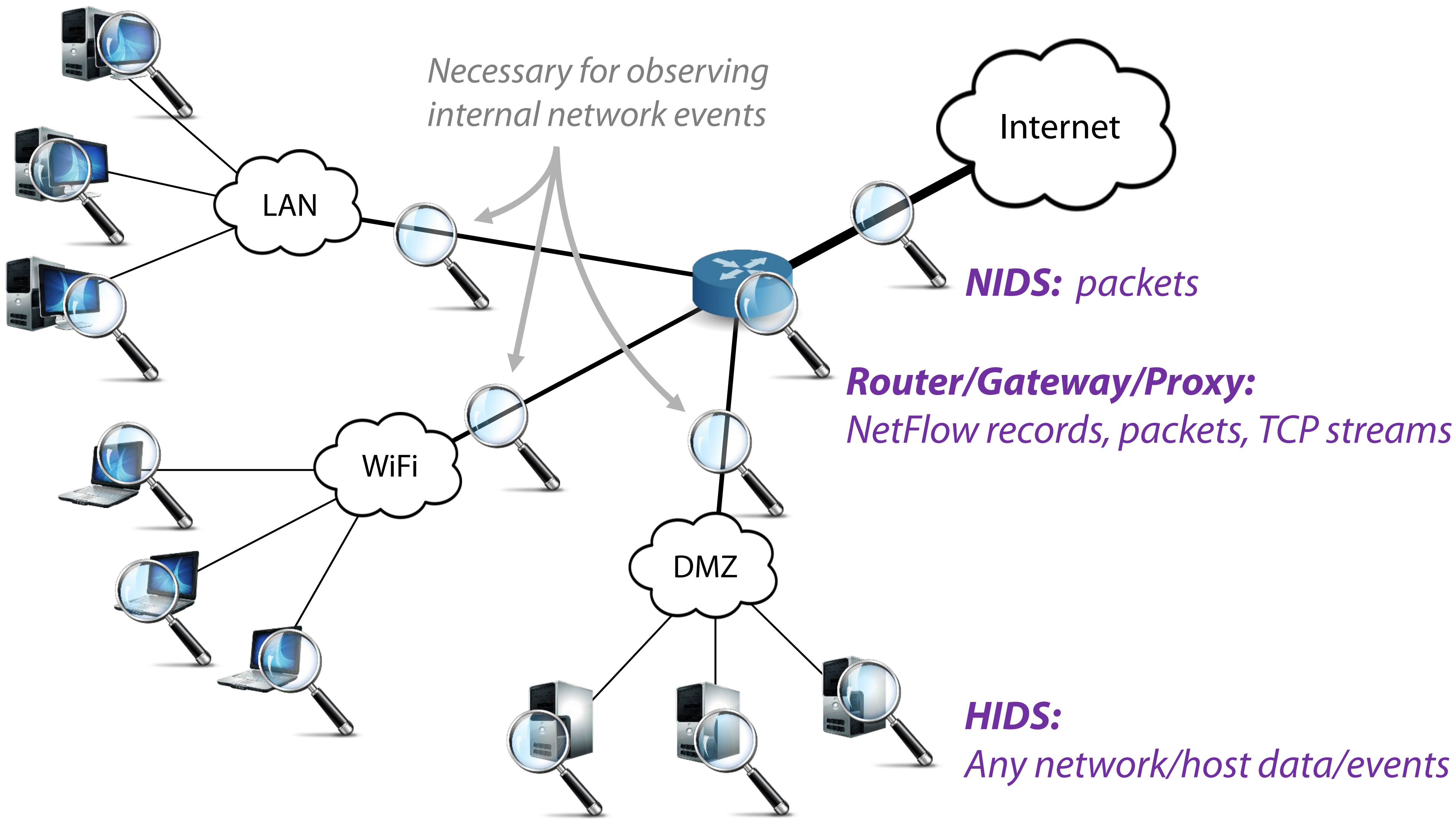
- Operates on captured *audit data*
- Off-line (e.g., periodic) vs. real-time processing

Network (NIDS)

- NetFlow records, raw packets, reassembled streams, DNS messages, …
- Passive (IDS) vs. in-line (IPS) operation
- Examples: Snort, Zeek, 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: OSquery, OSSEC, SysDig, El Jefe, AVs, registry/process/etc. monitors, network content scanners, …


Necessary for observing internal network events
Internet
LAN
NIDS: packets
Router/Gateway/Proxy:
NetFlow records, packets, TCP streams
WiFi
DMZ
HIDS:
Any network/host data/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 level

Context

NIDS: packets, unencrypted streams (unless proxy-level TLS interception is used)

HIDS: full picture (e.g., API-level monitoring to inspect data before it is encrypted)

Overhead

NIDS: none (passive)

NIPS/Proxy: adds some latency

HIDS: eats up CPU/memory (varies from negligible to complete hogging)

Subversion

NIDS: invisible in the network (passive component)

NIPS/Proxy: failure may lead to network reachability issues (in-line component)

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" (and "malicious") behavior
- Requires (re)training with an adequate amount of data
- Can potentially detect previously unknown attacks
- Prone to false positives

IDS Challenges

Conflicting goals: zero-day attack detection vs. zero false positives

Resilience to evasion

- Usually it is easy for adversaries to morph the attack vector and evade detection

Detection of targeted and stealthy attacks

- No prior knowledge of how the attack may look like

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

- Log/event aggregation tools (e.g., Splunk)

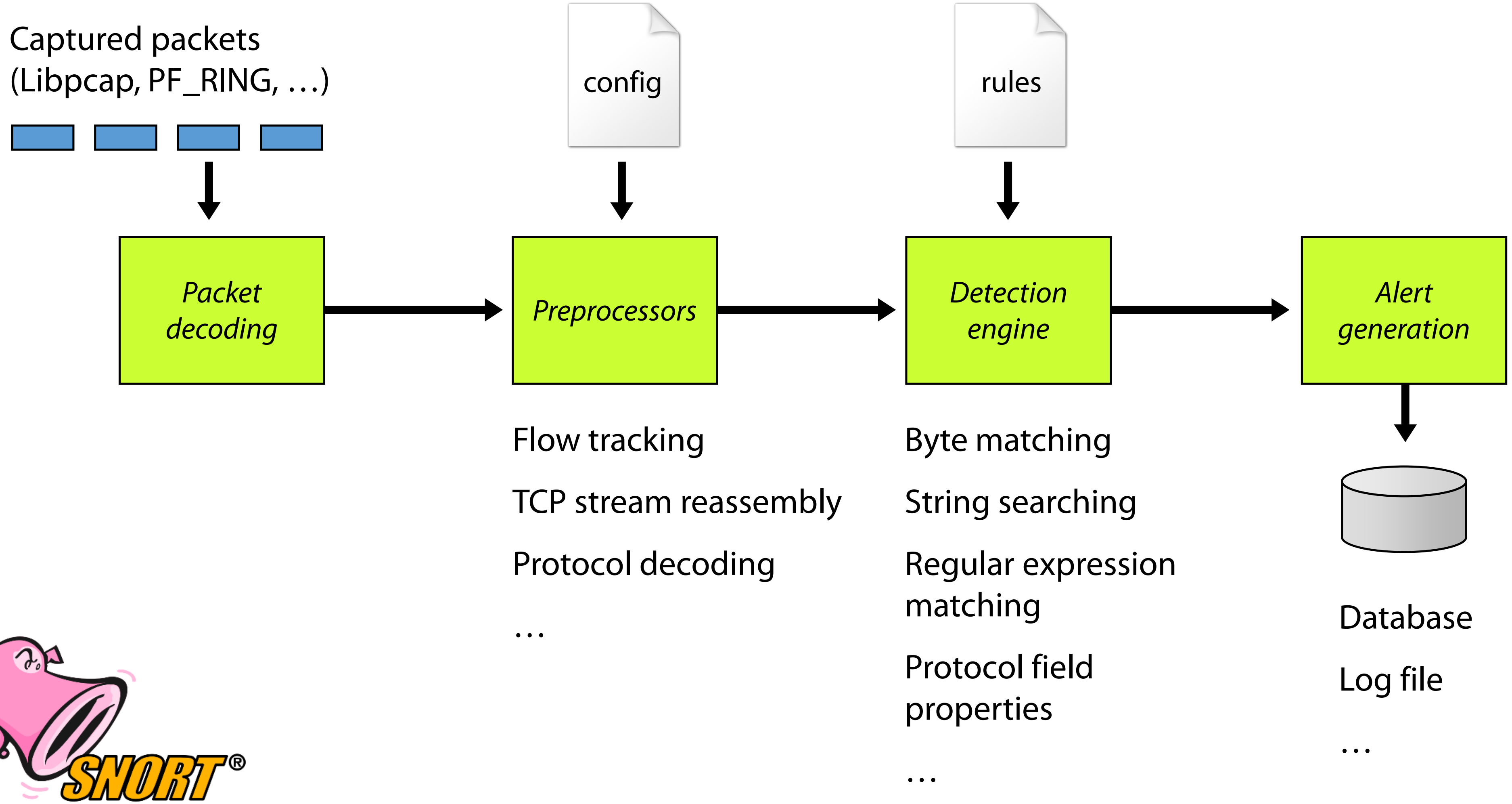
Popular Open-source Signature-based NIDS

Snort

Zeek

Suricata

Use Case: Snort



SNORT®

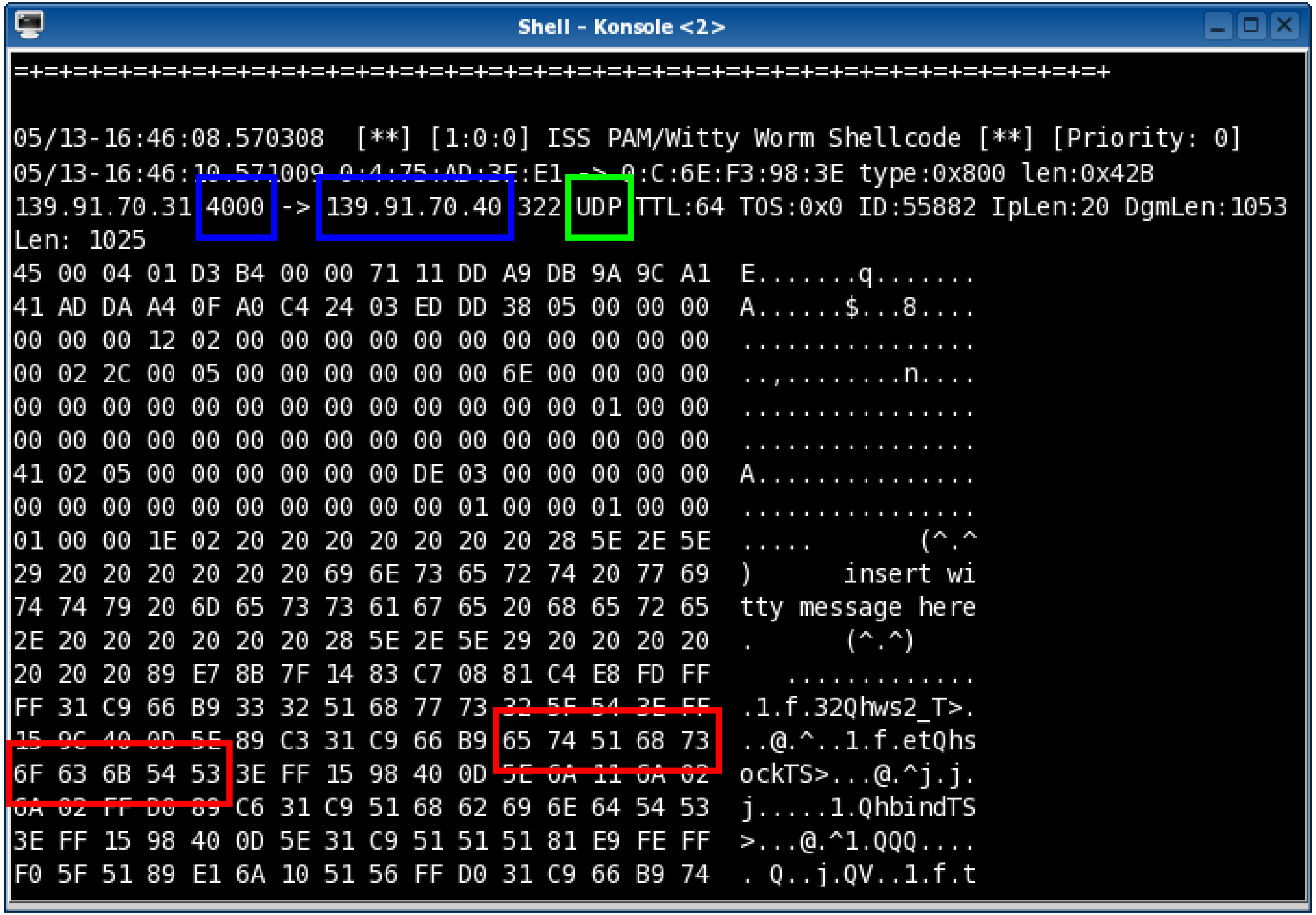
What is a Signature?

An attack description as seen at Layer 2-7

Example Snort signature for Witty worm:

action *protocol* *source/destination* *content*

```
alert udp any 4000 -> 193.92.123.0/24 any (msg:"ISS
PAM/Witty Worm Shellcode"; content:"|65 74 51 68 73
6f 63 6b 54 53|"; depth:246; sid:1000078; rev:1;)
```


Shell - Konsole <2>
=+
05/13-16:46:08.570308 [**] [1:0:0] ISS PAM/Witty Worm Shellcode [**] [Priority: 0]
05/13-16:46:10.571009 0:4:75:AD:3E:E1 -> 0:C:6E:F3:98:3E type:0x800 len:0x42B
139.91.70.31:4000 -> 139.91.70.40:322 UDP TTL:64 TOS:0x0 ID:55882 IpLen:20 DgmLen:1053
Len: 1025
45 00 04 01 D3 B4 00 00 71 11 DD A9 DB 9A 9C A1 E.......q.......
41 AD DA A4 0F A0 C4 24 03 ED DD 38 05 00 00 00 A......$...8....
00 00 00 12 02 00 00 00 00 00 00 00 00 00 00 00 ....
00 02 2C 00 05 00 00 00 00 00 00 6E 00 00 00 00 ..,........n....
00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 ....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
41 02 05 00 00 00 00 00 00 DE 03 00 00 00 00 00 A...............
00 00 00 00 00 00 00 00 00 00 01 00 00 01 00 00 ....
01 00 00 1E 02 20 20 20 20 20 20 20 28 5E 2E 5E .....   (^.^
29 20 20 20 20 20 20 69 6E 73 65 72 74 20 77 69)   insert wi
74 74 79 20 6D 65 73 73 61 67 65 20 68 65 72 65 tty message here
2E 20 20 20 20 20 20 28 5E 2E 5E 29 20 20 20 20 .   (^.^)
20 20 20 89 E7 8B 7F 14 83 C7 08 81 C4 E8 FD FF   ....
FF 31 C9 66 B9 33 32 51 68 77 73 32 5F 54 3E FF .1.f.32Qhws2_T>.
15 9C 40 0D 5E 89 C3 31 C9 66 B9 65 74 51 68 73 ..@.^..1.f.etQhs
6F 63 6B 54 53 3E FF 15 98 40 0D 5E 6A 11 6A 02 ockTS>...@.^j.j.
6A 02 FF D0 89 C6 31 C9 51 68 62 69 6E 64 54 53 j.....1.QhbindTS
3E FF 15 98 40 0D 5E 31 C9 51 51 51 81 E9 FE FF >...@.^1.QQQ....
F0 5F 51 89 E1 6A 10 51 56 FF D0 31 C9 66 B9 74 ._Q..j.QV..1.f.t

More Examples

String searching

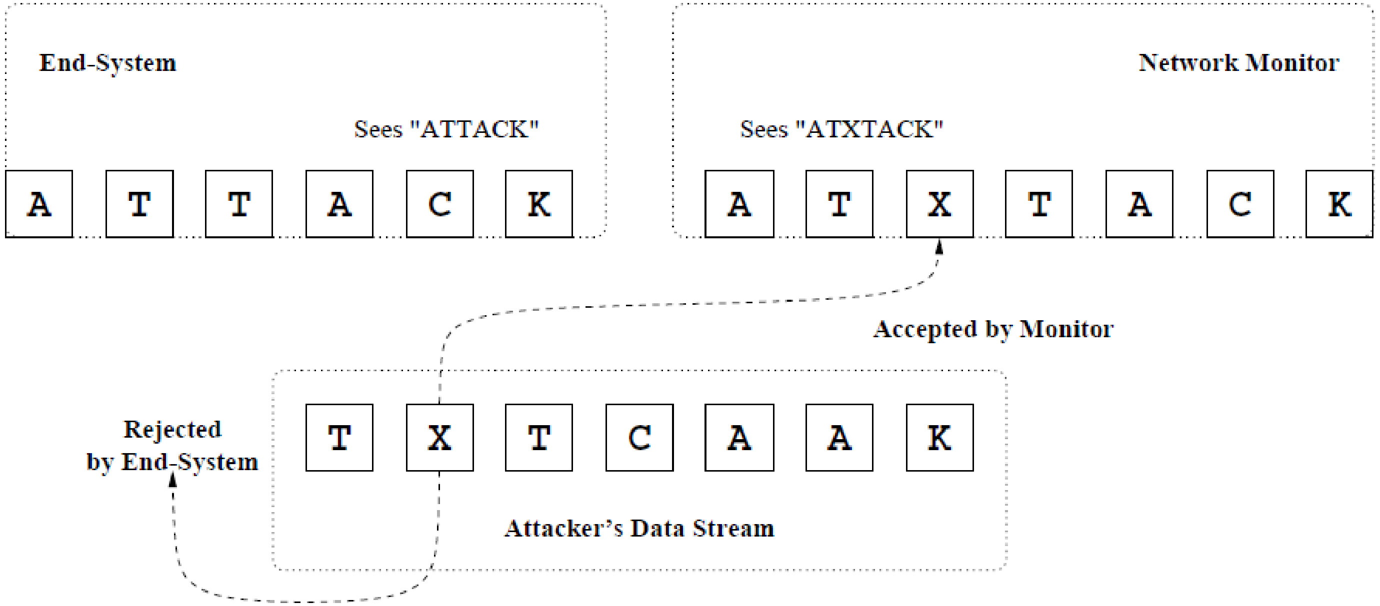
```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any
(msg:"SHELLCODE Linux shellcode"; content:"|90 90 90 E8
C0 FF FF FF|/bin/sh"; classtype:shellcode-detect;
sid:652; rev:9;)
```

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+\S{65}|\S+\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***

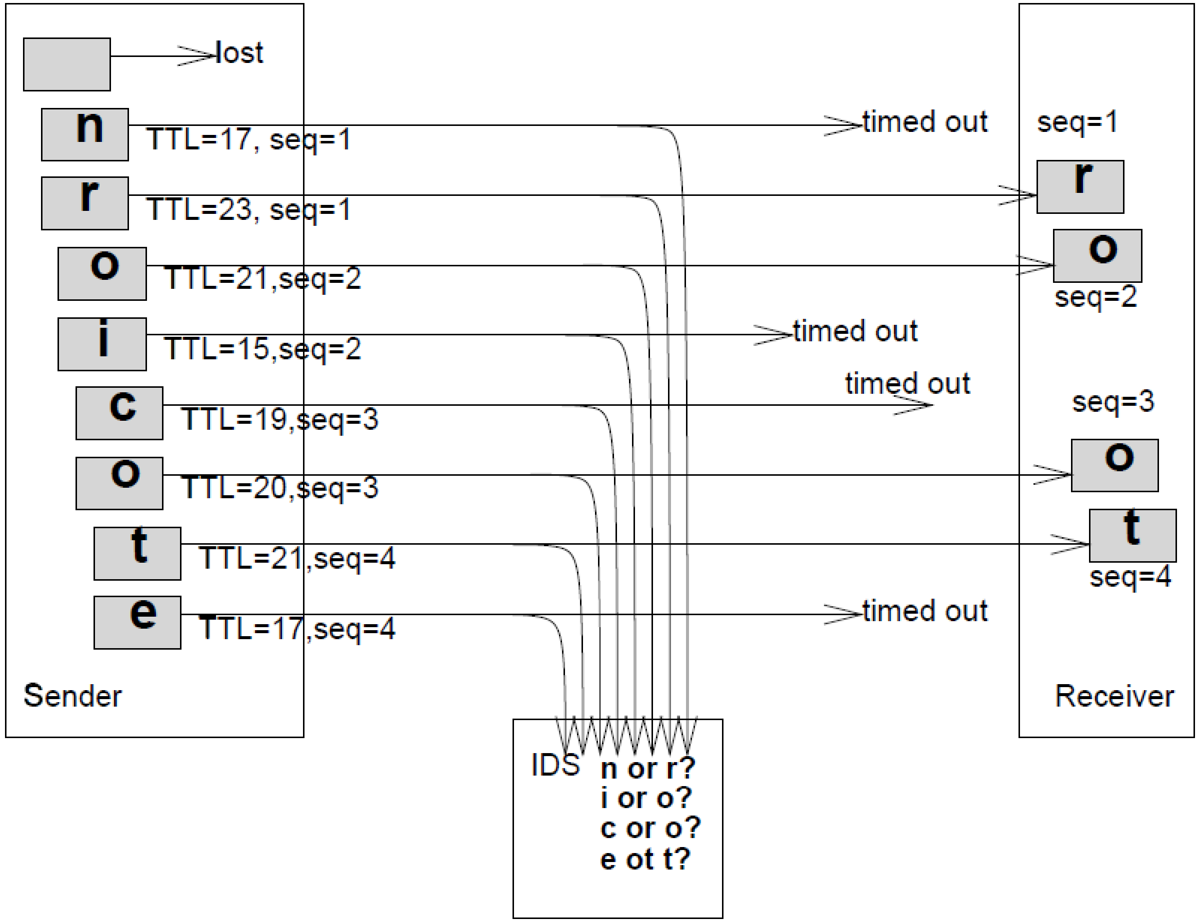


Solution: IP defragmentation, TCP stream reassembly

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

- Example: 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
- A dummy detector that marks everything as benign would have 99.9% accuracy…

False positive: legitimate behavior that was deemed malicious

False negative: an actual attack that was not detected

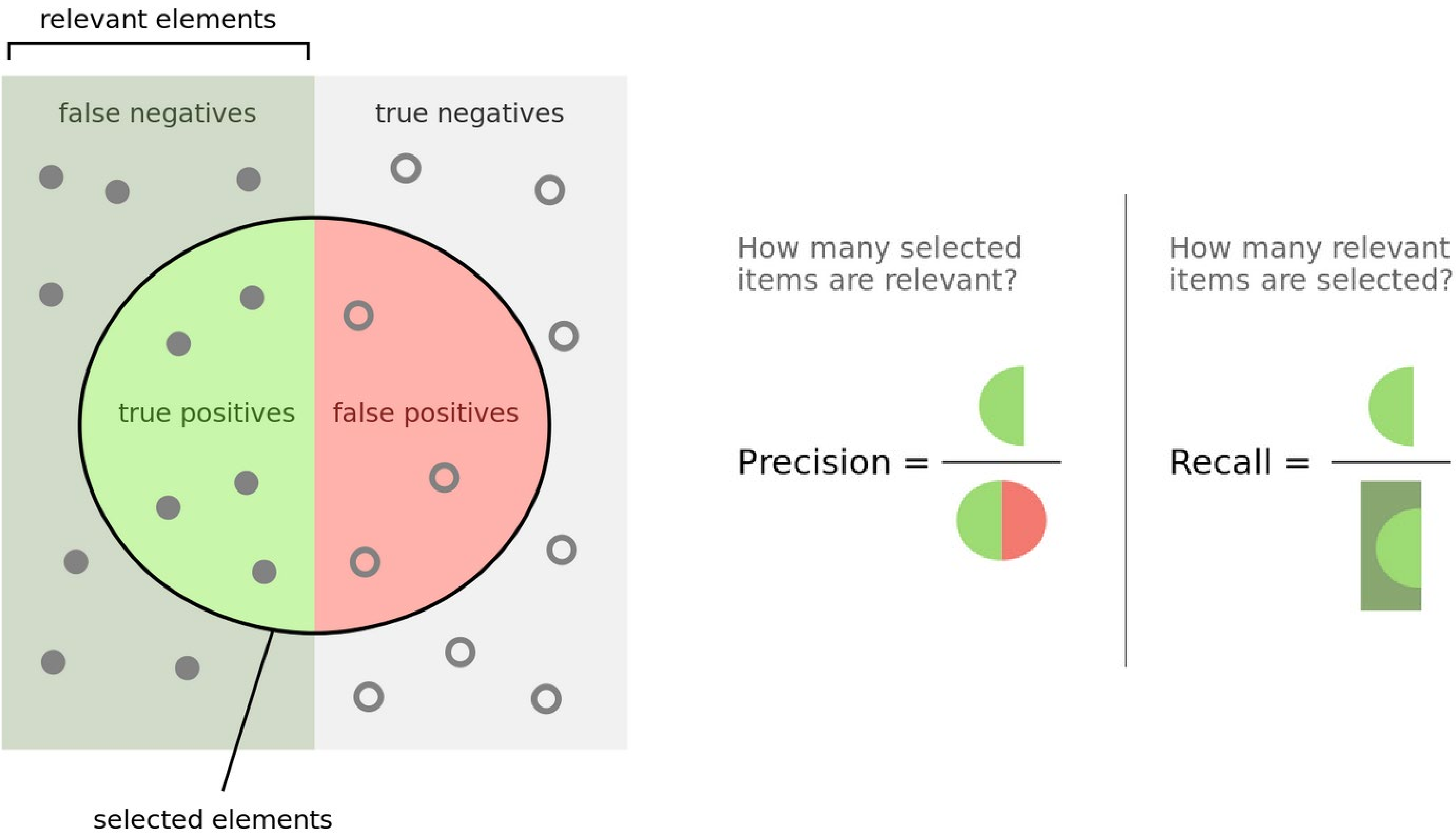
Detection Result

Actual Event	Positive (alert)	Negative (silence)
Positive (malicious)	**TP**	**FN**
Negative (benign)	**FP**	**TN**

Precision = $TP / (TP+FP)$

Recall (sensitivity) = $TP / (TP + FN)$

FP rate = $FP / (FP + TN)$



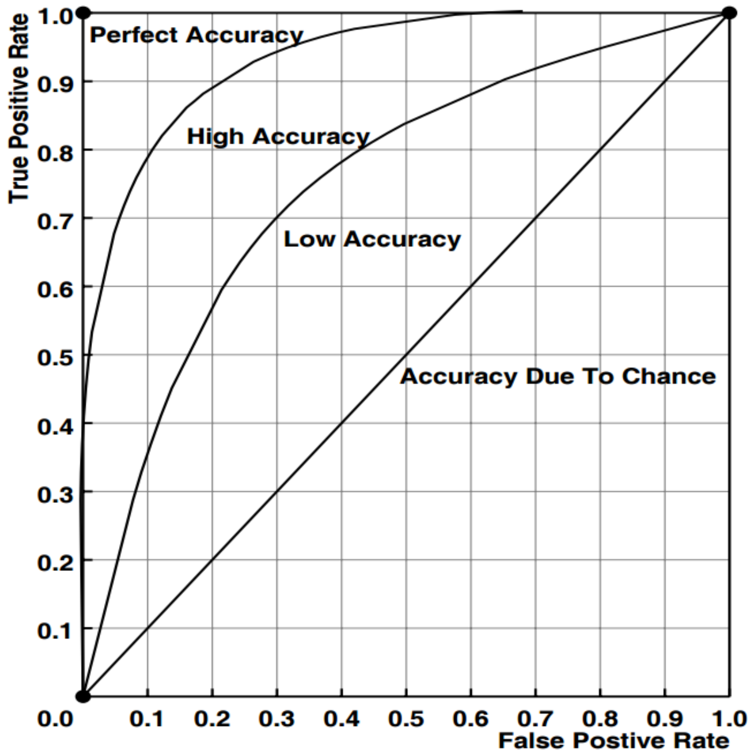
https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

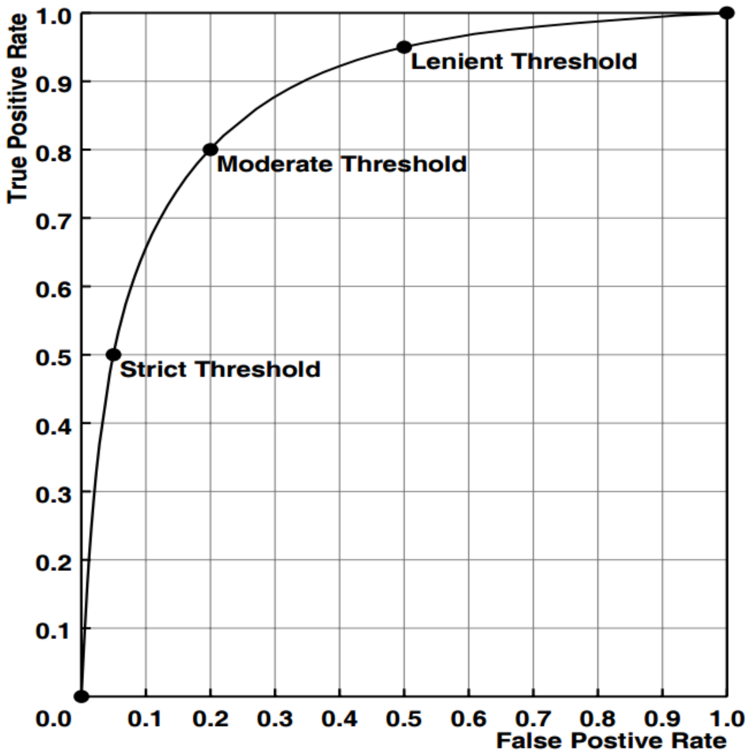
Receiver Operating Characteristic (ROC) Curve

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
- …

Passive DNS Monitoring

Store DNS resolution data (indefinitely) to detect potential threats or malicious C&C communication

- Can aid in forensic analysis after an incident has been detected
- Can be combined with allow/deny/reputation lists

DNS data can be captured at various locations

- Directly in the network's recursive server
- Sniffing raw network traffic
- On each endpoint (especially if DoT/DoH is used)

Related service: Protective DNS

- The resolver checks all queries/responses against threat intelligence data and prevents connections to known or suspected malicious sites