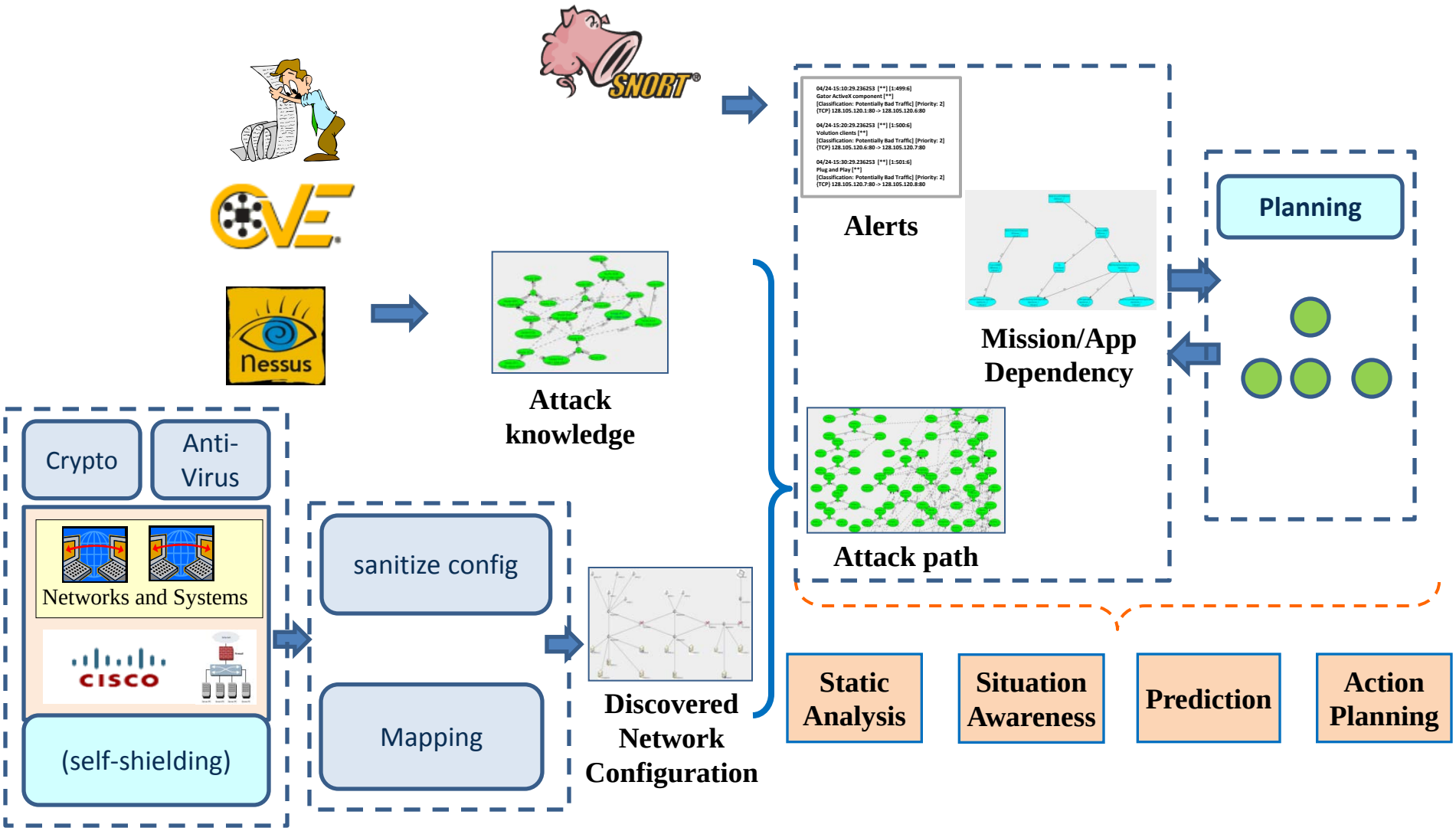
Intrusion Detection System

CSC 154

Slides adapted from Peng Liu, PSU

Comprehensive Security Analysis: Roadmap



Picture from Jason Lee, Peng Liu, MilCom Tutorial: Cyber Security and Situational Awareness: Theory and Practice

What are intrusions?

- A cyber attack may either succeed or fail
- Successful attacks are called intrusions
  - Vulnerability
  - Exploits
- Failed attacks are NOT intrusions

Intrusion Types

- Host intrusions
- Network intrusions
- Application intrusions

Host Intrusion Example

- Mallory logged into your laptop (a host) using your username and password
  - He then deletes your Phase-4 report and Lab 4 report, and
  - You are very mad!
- This intrusion is due to a successful masquerading attack
- Mallory is interested in doing bad operations on your laptop

- Other: kernel vulnerability exploited for root access

Network intrusion example

- Mallory sends a buffer overflow attack packet to a web server 1,000 miles away
- This packet includes malicious code in its payload
- The packet overflows the buffer of the server → the malicious code is executed → the server becomes a bot or zombie
- Similarly Mallory breaks into 2,000 other web servers and gets 2,000 bots
- Mallory's goal is to create a botnet, instead of a single host

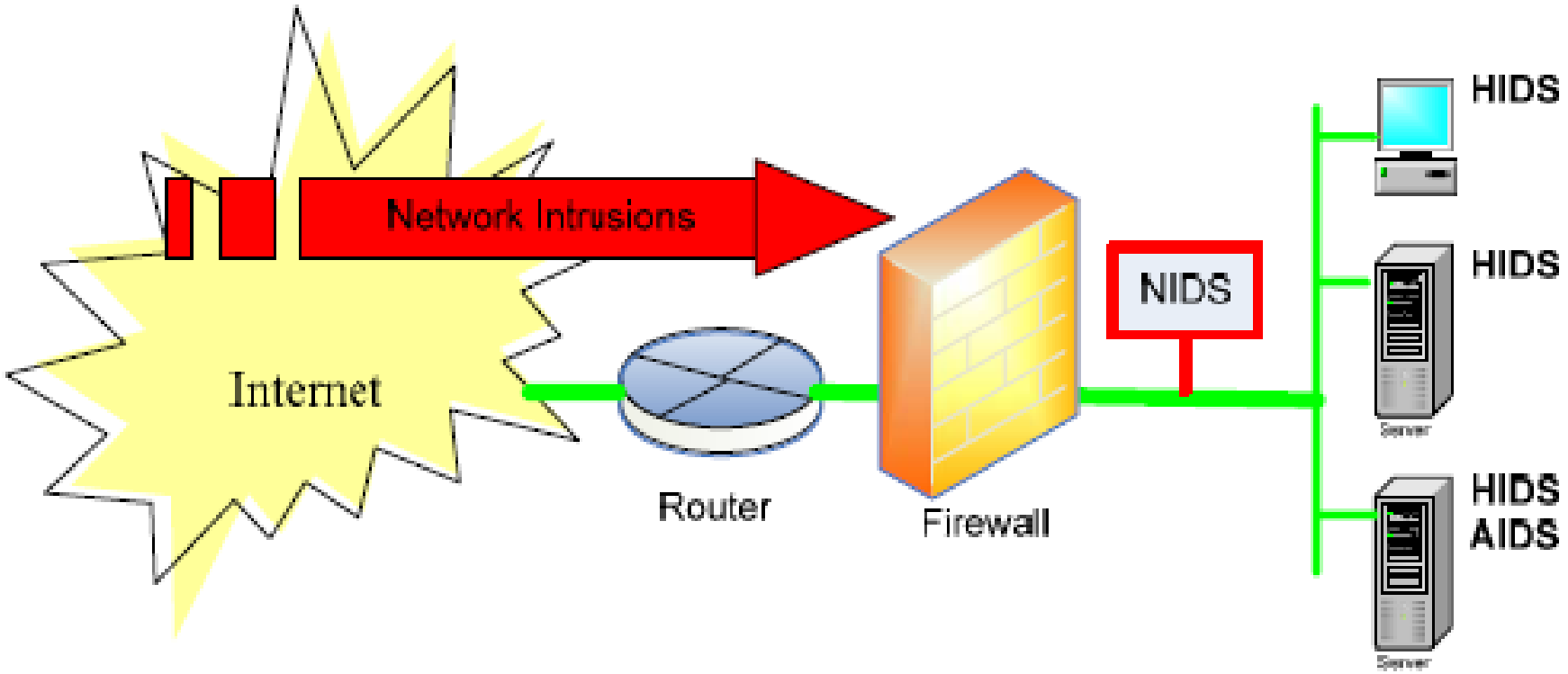
Application intrusion example

- Through a SQL injection attack, Mallory changes the interest rate of his mortgage in the bank's database from 7% to 4%
  - Malicious SQL statements injected into query
  - Ex. ' union SELECT 1, load_file('/etc/passwd') #
- Application intrusions abuse the application logic instead of the database server's OS
- Application intrusions are semantic-aware

Three types of IDS

- Intrusion Detection Systems (IDS)
  - Host-based IDS (HIDS)
  - Network-based IDS (NIDS)
  - Application-based IDS (AIDS)

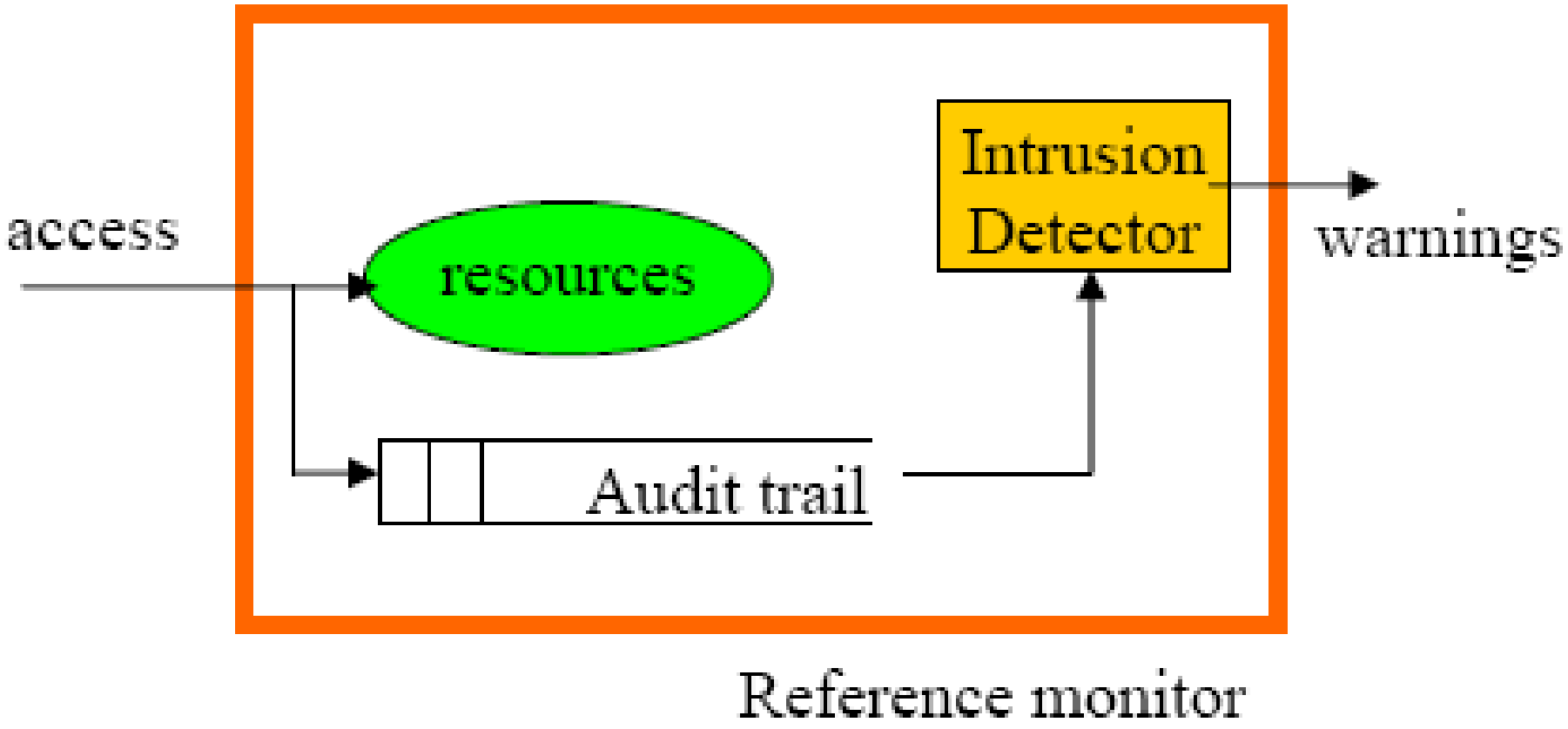
Three types of IDS



Idea of HIDS

- Job 1: Record each user's activity → you get an audit trail for each user
  - Example activities: Windows operations performed by the user
    - Read a file; Delete a file; Execute a program
- Job 2: Analyze the audit trail periodically to identify anomaly
  - What is abnormal?
    - Alice never deletes files in directory X, but she did it today
    - Alice never runs command X, but she did it today

HIDS


access
resources
Intrusion Detector
warnings
Audit trail
Reference monitor

Targets of HIDS

- L2R
- Attempted break-in
- Masquerading
- Trojan horse
- Virus
- Spyware
- Rootkit
- Bots
- Leakage by a legitimate user
- Inference by a legitimate user

Idea of NIDS – Signature-Based Detection

- Job 1: Record each incoming or outgoing packet → you get the packet trace
- Job 2: Analyze each packet in the trace to see if an attack signature is matched
  - A signature is a special byte string that only appears in attacking packets
  - Other methods: anomaly detection

Look Into a Bad Packet

The Orignal Packet of Code Red:

GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u909
0%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190
%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0\r\n

Targets of NIDS

- R2L
- Probing
- Worm
- DoS
- Botnets

Idea of AIDS

- Exploit the application's semantics to identify anomaly

- What is an application anomaly?
  - Example: No mortgage issued in 2006 should have an interest rate below 6%, but Alice's new mortgage enjoys a 5% rate → unbelievable!

HIDS Approaches

- Method 1: session level HIDS
- Method 2: anti-virus
  - Fingerprints of virus (e.g., Trojans)
- Method 3: system call level HIDS
- Other methods:
  - Buffer overflow detection techniques
  - Data flow graph based detection
  - Control flow graph based detection

Session Level Intrusion Detection

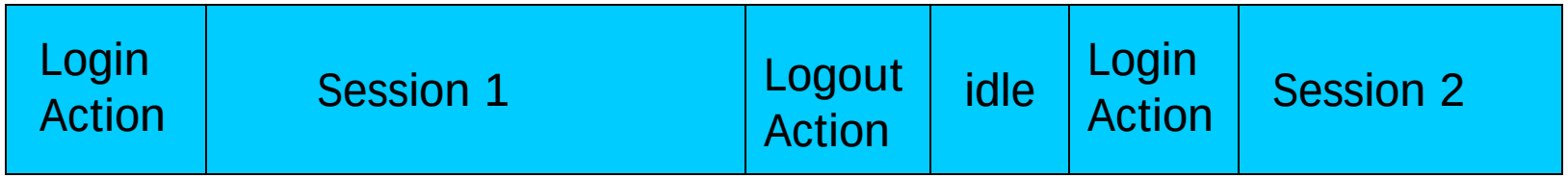
- Session level (log-in session): anomaly detection
- Measures to characterize a session: intensity measures, categorical measures, counting measures, …

Measure	*Description*
CPU usage	CPU time
Audit Record	# of audit records (for each hour)
File Usage	# of times each file was accessed
System Errors	# of times each type of error occurred
Directory Usage	Whether a directory was accessed
System Call	# of times each system call was used

HIDS

- The intrusion: Mallory logged into your host using your username and password; then he deletes your phase 4 report and lab 4 reports; then he logged out.

- The detection is done session by session
  - A session includes all the actions taken by a user between login and logout



HIDS

- Job 1: get the audit trail
  - Your audit trail is basically a sequence of sessions
- Job 2: check if any session is abnormal
  - Job 2A: How to **characterize** a session?
  - Job 2B: How to know if a session is abnormal?

HIDS

- During each session:
  - Some CPU time is consumed
  - Some files are used (read/write/delete)
  - Some commands are used (delete, etc)

- Hence, the intrusion session can be characterized by 3 numbers: <20, 2, 2>
  - 20 seconds CPU time
  - 2 files are used → phase 4 report; lab 4 report
  - 2 commands are used: delete → delete

HIDS

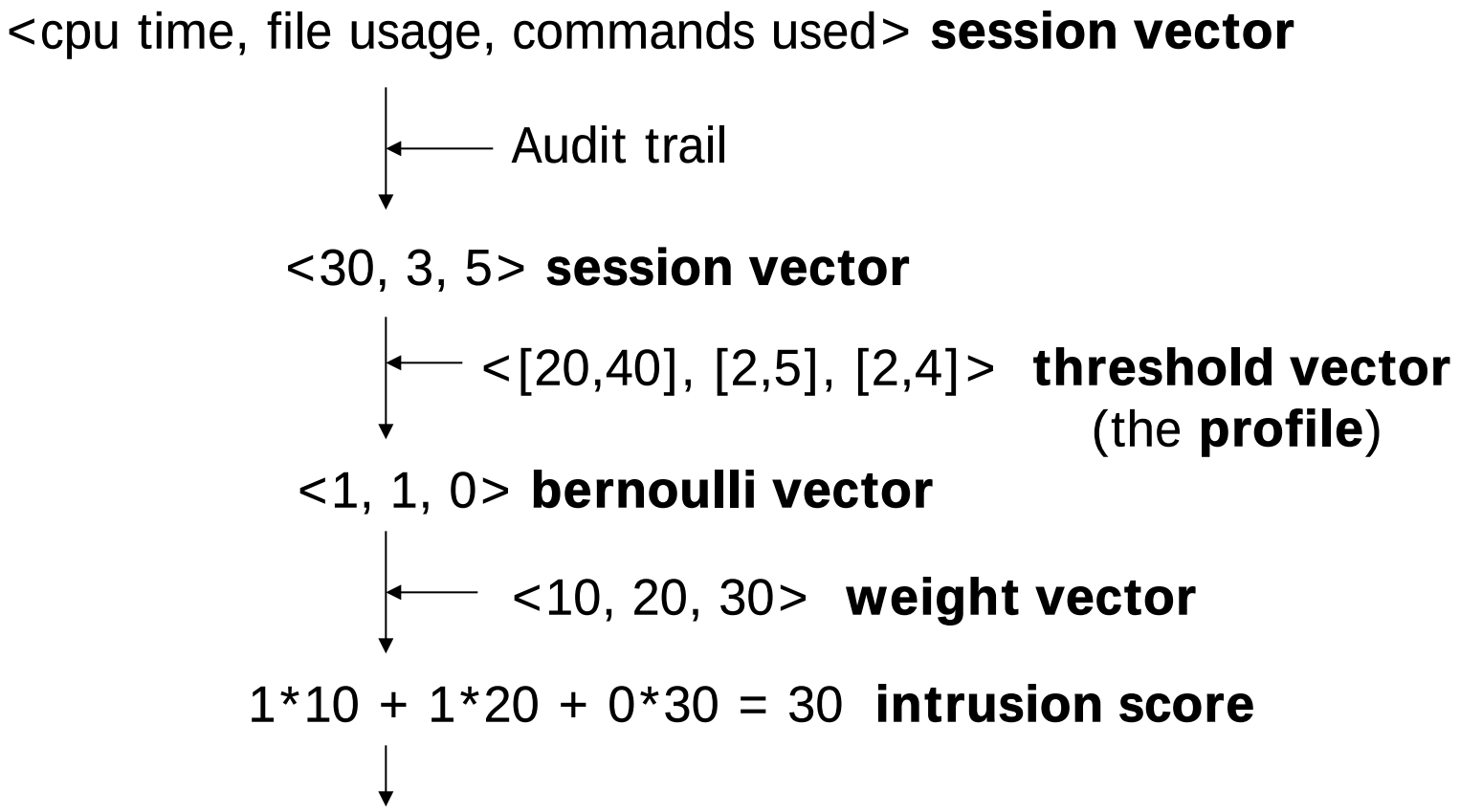
- Idea: The IDS knows the characteristics of a typical session of yours

- A typical session of yours can be characterized by 3 ranges:
  - [15, 50] CPU time
  - [5, 10] # of files used
  - [15, 25] # of commands used

- The intrusion session is normal in terms of CPU time, but abnormal in terms of range 2 and range 3

The Haystack Algorithm

- **The NIDES algorithm is more advanced.**
  - Next-Generation Intrusion-Detection Expert System



The Haystack Algorithm

suspicion quotient - the probability that a random session's intrusion score is less than or equal to the session's

Session 1 -- 50
Session 2 -- 30
Session 3 -- 60
Session 4 -- 40
Session 5 -- 10
Session 6 -- 60
Session 7 -- 20
Session 8 -- 30
Session 9 -- 40
Session 10 -- 50

suspicion quotient = 0.40

0 - very suspicious
1 - very normal

Summary of Session-based HIDS

- This approach is an **anomaly detection** approach
- This approach is profile based
- This approach is a statistical approach
- This approach can be used to detect many other kinds of attacks

System Call Level Intrusion Detection

- *S. Forrest, S. A. Hofmeyr, and A. Somayaji. A sense of self for Unix processes. IEEE S&P, 1996.*

  - **Key observation: when a process is compromised, its system call sequences often look abnormal**

    - In this sense, the process is no longer "itself"
    - "Short sequences of system calls executed by running processes are a good discriminator between normal and abnormal operating characteristics of several common UNIX programs."
    - "We look at all overlapping sequences of length *K* in the new trace and determine if they are represented in the normal database."
      - The magic value of *K* is 6

Signature-based NIDS: Snort

- Each "alert" rule contains a signature
- A signature is **a string of bytes**

```
alert tcp any any -> 195.4.12.0/24 111 (content: "|00 01 B6 a5|"; msg:
"external mountd access";)
```

A Four-Byte Signature

Tells the dest IP & port number

Two step signature matching:

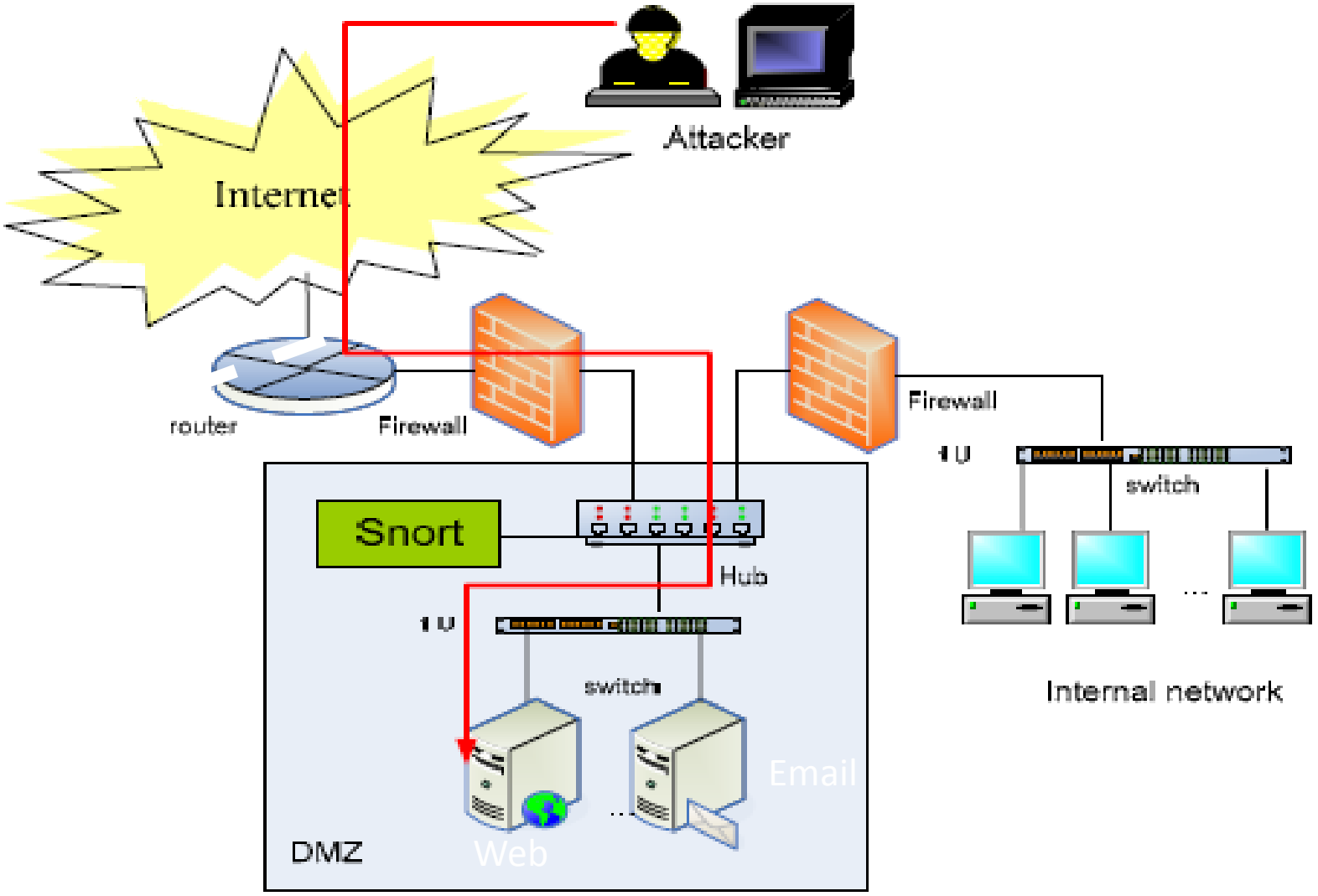
--Step 1: match the packet's *header* against the rule

--Step 2: match the *payload* against the 4-byte signature

--Principle: every attack packet is targeting the receiver program, which is identified by a **port** number

On a single PC (single IP), there are multiple programs

Snort in Scenario



Snort Log Rules

```
log udp any any -> 195.4.12.0/24 1:1024
```

Snort does not log all traffic – too much!
Log rules do NOT contain any signatures.

Snort Rules

```
alert tcp any any -> any 139 (content: !"GET";)
```

```
alert tcp any any -> any 21 (msg: "FTP ROOT"; content: "USER root";
nocase:)
```

```
Alert tcp 195.4.12.5 139 -> 195.4.12.5 139 (flags: S; msg: "possible
LAND attack";)
```

```
alert tcp any any -> any 80 (msg: "WEB-MISC Phf attempt"; content:
"cgi-bin/phf"; classtype: attempted-admin; offset: 4; depth: 20;)
```

Intrusion Detection (steps)

- 1. Monitor the targeted system and collect the audit trails
- 2. Analyze the gathered information for signs reflecting unusual activity and misuse
- 3. Ideally, automatically respond to detected activity, mitigating damages
- 4. Generate a report about suspicious activity and notify security people
- 5. Further investigate the nature of the discovered problem, document the cause
- 6. Diagnose the problem, providing a more in-depth understanding of the network's vulnerability
- 7. Rectify the problem to stop exploitation by future intruders

Effectiveness of HIDS, NIDS, and AIDS

- False rate
  - False positives (false alarm rate)
    - Non-intrusions detected as intrusions
  - False negatives (detection rate)
    - Intrusions undetected
- Detection latency

Anomaly Detection vs. Signature-based detection

Anomaly detection	*Signature-based*
Good for unknown attacks	Unable to detect unknown attacks
Limited for known attacks	Good for known attacks