

# **ECE560**

# **Computer and Information Security**

## **Fall 2024**

Intrusion Detection and Prevention

Tyler Bletsch  
Duke University



# Outline

Understanding intruders

Intrusion detection system (IDS)

Intrusion prevention systems (IPS)

Detection theory

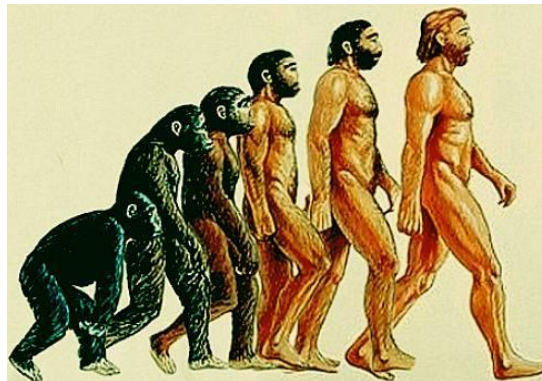
Firewalls

# Two ways to categorize intruders

- **Class of intruder:** What are they after?



- **Intruder skill level:** How smart are they?

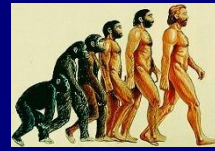


# Classes of intruder

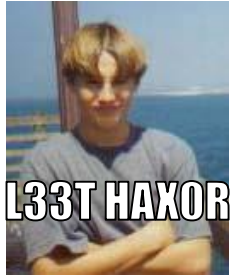


- **Class of intruder:** What are they after?
  - **Criminal** want to **monetize**: Turn attacks into money
    - Methods: Identity theft, corporate espionage, data theft, ransomware
    - Often Eastern European or southeast Asian (but *every* country has them)
    - Collaborate on dark web forums, conduct business on illicit sales sites
  - **Activists** want to **achieve political ends**
    - Methods: Deface websites, conduct DoS attacks, steal and leak data
  - **State-sponsored actors** want to **really achieve political ends**
    - Sponsored by governments. Also known as **Advanced Persistent Threats (APTs)** – covert, professional, long-term
    - Recent trends: Russia, China, and Iran attacking western powers; covert western counterattacks and overt western revelations
  - **Explorers**: motivated by learning or prestige
  - **Script kiddies**: using published tools to cause mischief

# Two ways to categorize intruders



- **Intruder skill level:** How smart are they?



- Apprentice

- Minimal technical skills, use existing tools
- Largest group, includes most criminals
- Easiest to defend against



All-of-you.jpg

- Journeyman

- Can modify existing tools and exploit newly published vulnerabilities
- Can discover some vulnerabilities

- Master

- Highly skilled, can discover new vulnerabilities broadly
- Writes their own tools
- Common in APT crews and at the top of criminal organizations
- Hardest to defend against



# **Intruders will want you to misapprehend their skill and motivation!**

- Criminals may want to seem like political activists to cover their true activities.
  - Apprentices want to appear like Masters.
  - Masters want to appear like Apprentices.
  - Etc.
- 
- During forensics, be hesitant to jump to conclusions...

# Intruder Behavior

1. Target acquisition and information gathering
2. Initial access
3. Privilege escalation
4. Information gathering or system exploit
5. Maintaining access
6. Covering tracks

### **(a) Target Acquisition and Information Gathering**

- Explore corporate website for information on corporate structure, personnel, key systems, as well as details of specific web server and OS used.
- Gather information on target network using DNS lookup tools such as dig, host, and others; and query WHOIS database.
- Map network for accessible services using tools such as NMAP.
- Send query email to customer service contact, review response for information on mail client, server, and OS used, and also details of person responding.
- Identify potentially vulnerable services, eg vulnerable web CMS.

### **(b) Initial Access**

- Brute force (guess) a user's web content management system (CMS) password.
- Exploit vulnerability in web CMS plugin to gain system access.
- Send spear-phishing email with link to web browser exploit to key people.

### **(c) Privilege Escalation**

- Scan system for applications with local exploit.
- Exploit any vulnerable application to gain elevated privileges.
- Install sniffers to capture administrator passwords.
- Use captured administrator password to access privileged information.

### **(d) Information Gathering or System Exploit**

- Scan files for desired information.
- Transfer large numbers of documents to external repository.
- Use guessed or captured passwords to access other servers on network.

### **(e) Maintaining Access**

- Install remote administration tool or rootkit with backdoor for later access.
- Use administrator password to later access network.
- Modify or disable anti-virus or IDS programs running on system.

### **(f) Covering Tracks**

- Use rootkit to hide files installed on system.
- Edit logfiles to remove entries generated during the intrusion.

## Table 8.1

# Examples of Intruder Behavior

(Table can be found on pages 271-272 in textbook.)





# Outline

Understanding intruders

Intrusion detection system (IDS)

Intrusion prevention systems (IPS)

Detection theory

Firewalls

# 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

**Comprises three logical components:**

- **Sensors - collect data**
- **Analyzers - determine if intrusion has occurred**
- **User interface - view output or control system behavior**

# Analysis Approaches

## Anomaly detection

- Collect data relating to the behavior of legitimate users
- Current observed behavior is compared to baseline
- Detect:
  - Denial-of-service (DoS) attacks
  - Scanning
  - Worms

## Signature/Heuristic detection

- Scan for known malicious data patterns via signature (e.g. antivirus) or rules (e.g. 'snort')
- Can only identify known attacks
- Detect:
  - Reconnaissance and attacks
  - Unexpected application services
  - Policy violations

# 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

# Host-Based Intrusion Detection (HIDS)

- Primary purpose is to detect intrusions, log suspicious events, and send alerts
  - Can detect both external and internal intrusions
- Data sources:
  - System call traces
  - Audit (log file) records
  - File integrity checksums
  - Registry access

# Distributed HIDS deployment

- Can put HIDS agents on many systems, manage centrally

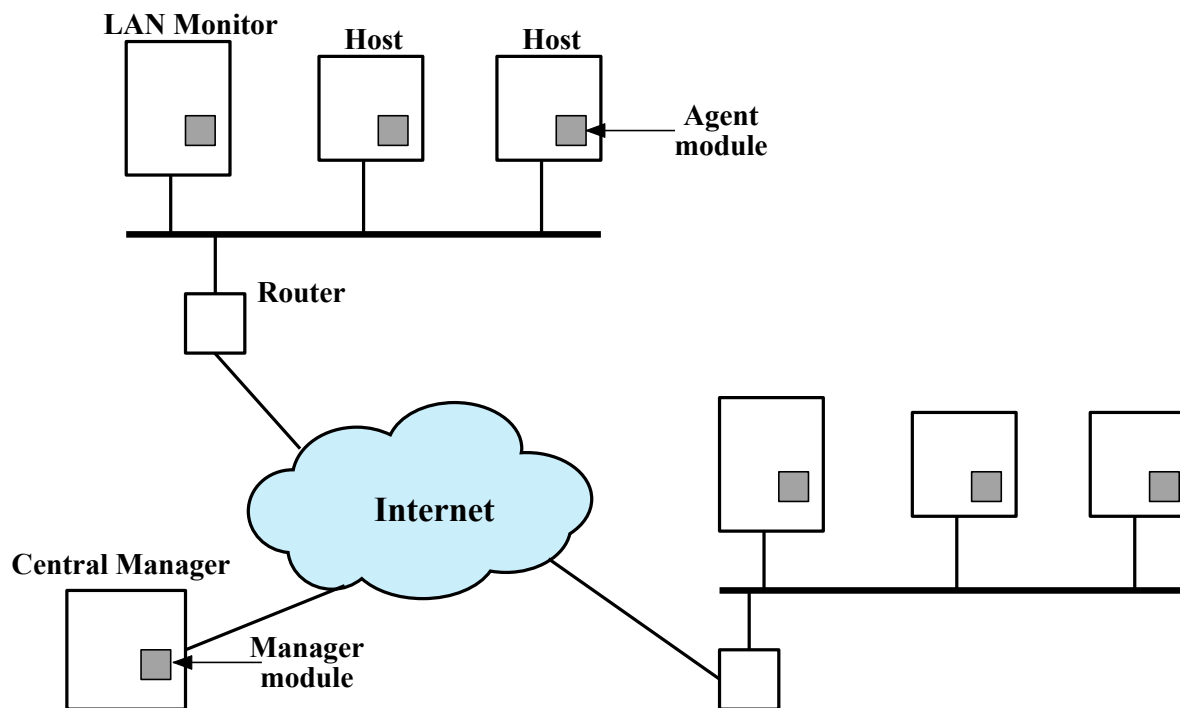
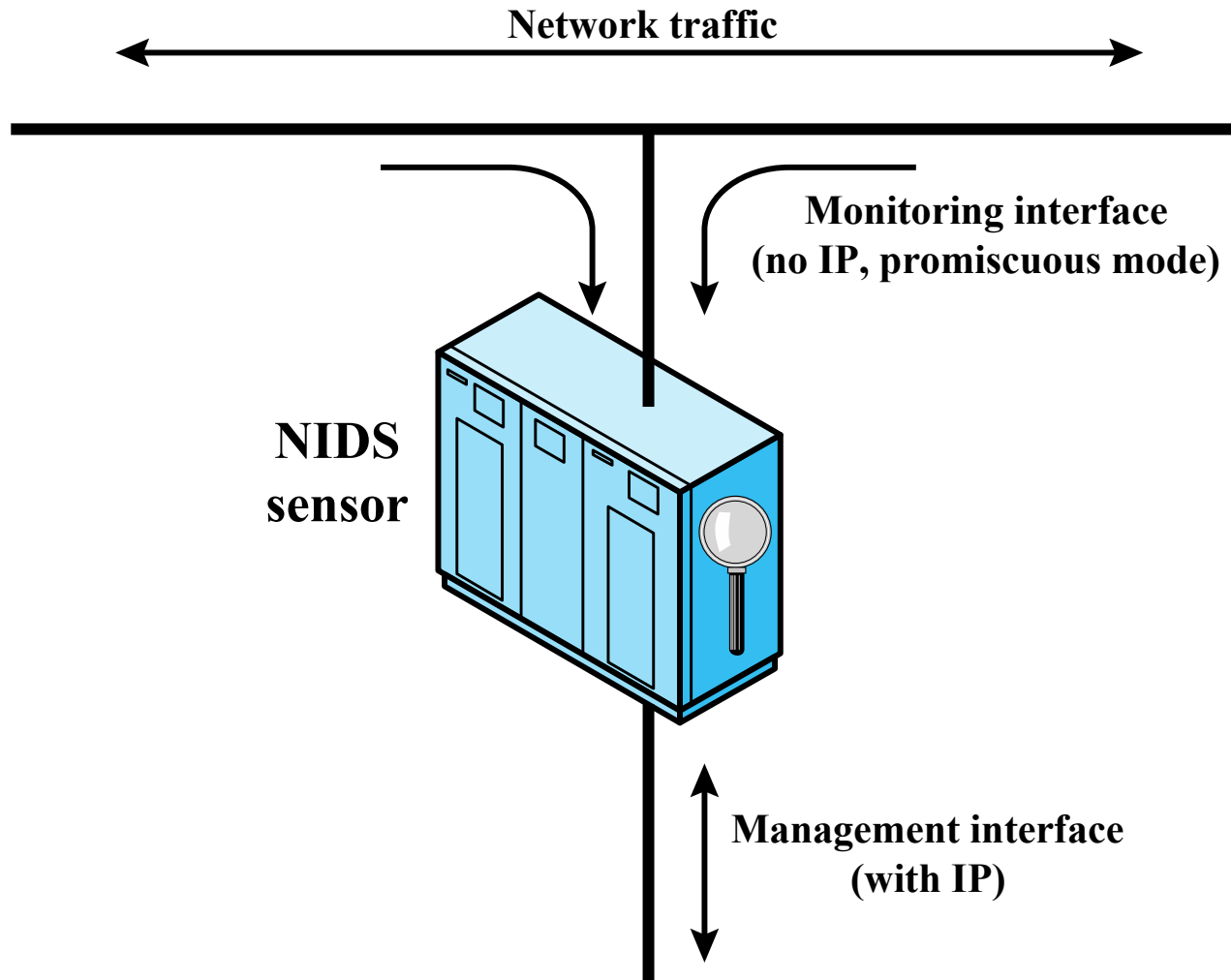


Figure 8.2 Architecture for Distributed Intrusion Detection



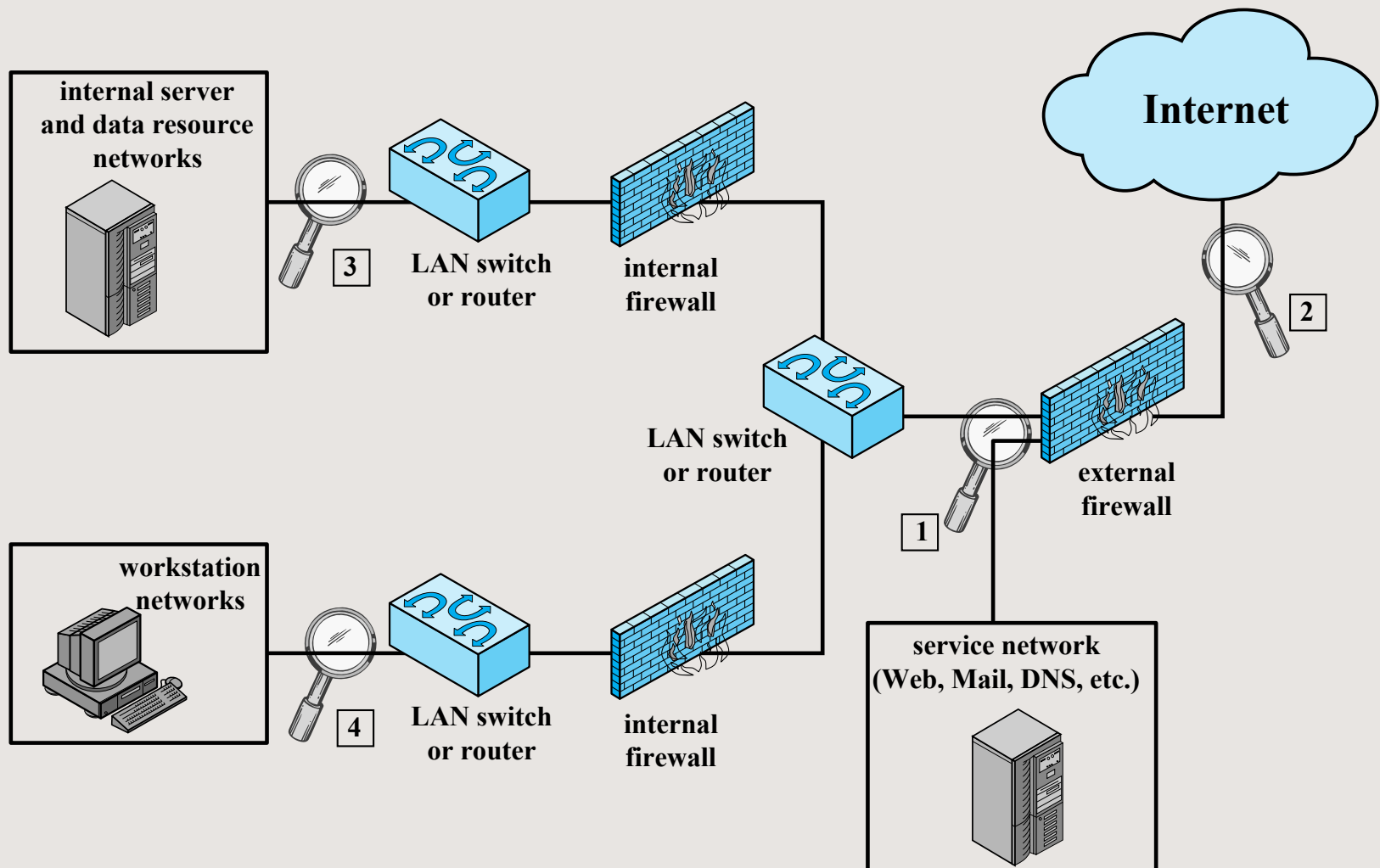
# Network-Based IDS (NIDS)

- **Monitors traffic** at selected points on a network
- **Examines traffic** packet by packet in real time
  - May examine network, transport, and/or application-level protocol activity
- **Comprised of:**
  - A number of sensors
  - One or more management servers
- Analysis of traffic patterns may be done at the sensor, the management server or a combination of the two



**Figure 8.4 Passive NIDS Sensor**





**Figure 8.5 Example of NIDS Sensor Deployment**

# Stateful Protocol Analysis

- Understands and tracks network, transport, and application protocol **states** to ensure they progress as expected
- Higher resource use than stateless systems

# Logging of Alerts

- Typical information logged by a NIDS sensor includes:
  - Timestamp
  - Connection or session ID
  - Event or alert type
  - Rating
  - 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

# Flow records

- Modern IDS will often keep **flow records**: info on every TCP connection and UDP flow.
  - Data usually not kept (too big + privacy reasons)
  - Know the connect time, source IP+port, destination IP+port, duration
- Motivation: Historical tracking of suspicious activity
  - “I now know this malware talks to 24.1.2.3, so which of my machines have been talking to that IP?”
  - “I learned that someone at IP address 34.2.3.4 used stolen credentials, where have they been connecting, and have those machines been doing anything weird since then?”
  - “The server became infected at 2:23am, what connections were going on around then?”
  - “Let me scan the flow records and find stuff that looks like portscans so I can investigate!”

# Honeypots



- 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
- Resources that have no production value
  - Therefore incoming communication is most likely a probe, scan, or attack
  - Initiated outbound communication suggests that the system has probably been compromised
- Classified as being either low or high interaction
  - 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



# Outline

Understanding intruders

Intrusion detection system (IDS)

Intrusion prevention systems (IPS)

Detection theory

Firewalls

# Single slide coverage of (almost) all IPS

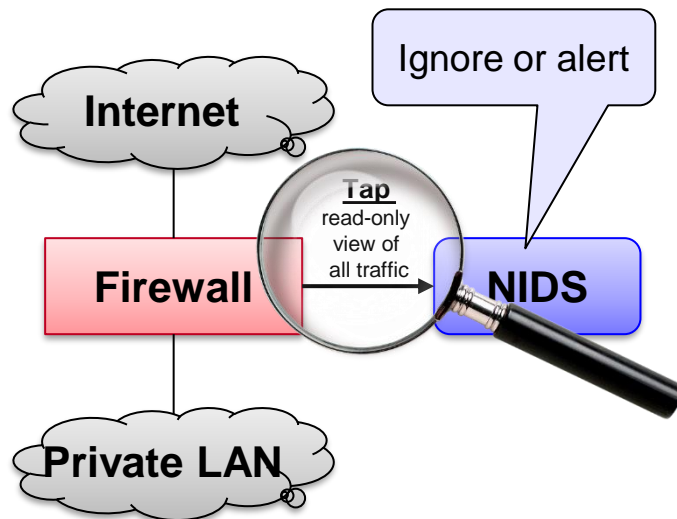
## **Intrusion Prevention System (IPS):**

It's IDS that can do something about stuff it sees

# Example: NIDS vs NIPS

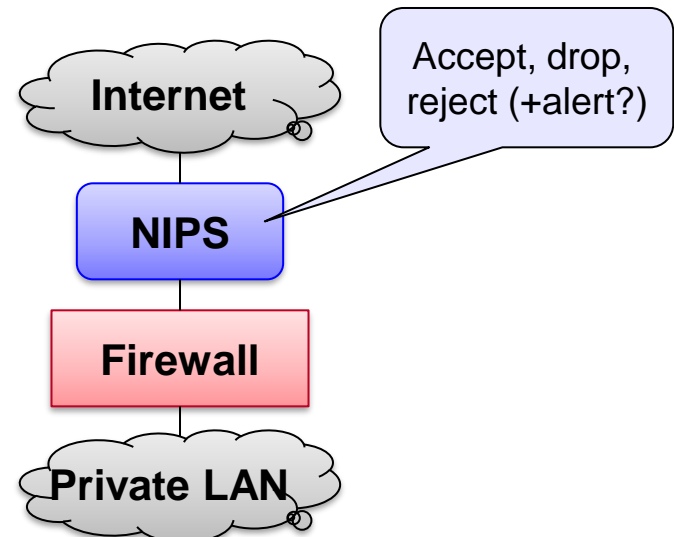
## NIDS (mirrored port)

- Can only comment passively on traffic it sees
- False positive: Spurious alert



## NIPS (inline)

- Can **drop** (ignore) or **reject** (drop with ICMP notice sent to sender) any packet it doesn't like; can also alert.
- False positive: Breaks stuff

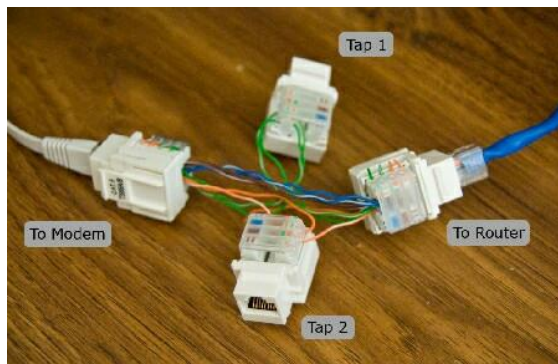




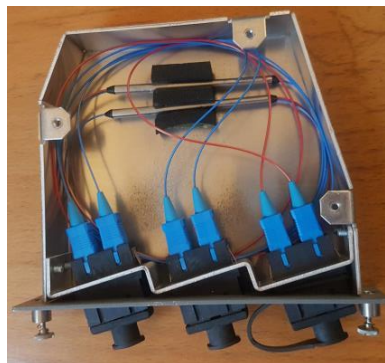


# Wait, how do you get all the traffic like that?

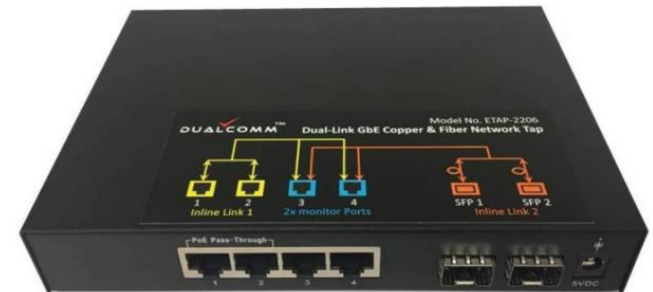
- Network **passive taps**:
  - Classic bidirectional copper (e.g. 100Mb Ethernet): **passive tap** has separate transmit and receive **wires** – literally splice them off
  - Modern optical fiber (e.g. fiber Ethernet): **passive tap again!** separate transmit and receive **fibers** – can use a passive light splitter!
- Network **active taps** (AKA “**port span**”):
  - Can always have hardware that replicates packets to another port
  - Can be done by dedicated hardware or by many modern network switches
    - When done on a switch, it’s often called a **port span**



*Passive tap for copper Ethernet*



*Passive tap for fiber Ethernet*



*Active tap for copper+fiber Ethernet*

# NIPS at Duke

- All the “Is this your student?” emails I’ve gotten from OIT were from Duke’s IDS/IPS system, which is comprised of several components
- Examples:
  - Portscans are detected using a homespun python script that looks at flow data from a network logger and triggers if unique targets for a given service exceeds a threshold – threshold is configurable per service.
    - Example alert data:

The alert condition for 'Duke Scanners by IP' was triggered.

This alert triggers when the argus scanner detect processes detects an IP on our networks that appears the be scanning. The behavior should be investigated to make sure that it was intentional and not malicious. If so and is likely to reoccur, we should see if the IP is static and possibly exclude it from this alert.


-----

ip,port,hosts\_touched,threshold,firstseen,lastseen,host

152.3.53.133,22,256,50,2018-10-25\_20:30:20,2018-10-25\_20:55:27,kali-vcn-28.vm.duke.edu
  - Auto-blocking of VictimCo incoming IP address: Caused because the unencrypted reverse shell content contained info about an .htaccess and/or .htpasswd file (one of many rules that this flow would eventually violate)
    - “Solved” by whitelisting VictimCo with OIT’s IDS/IPS systems

# Examples of free modern IDS/IPS

- OSSEC: Open source, cross platform HIDS



OSSEC WebUI Version 0.8

Main Search Integrity checking Stats About

---

November 07th, 2018 08:56:58 AM

### Available agents:

- +ossec-server (127.0.0.1)
- +Node2 (192.168.43.193)

### Latest modified files:

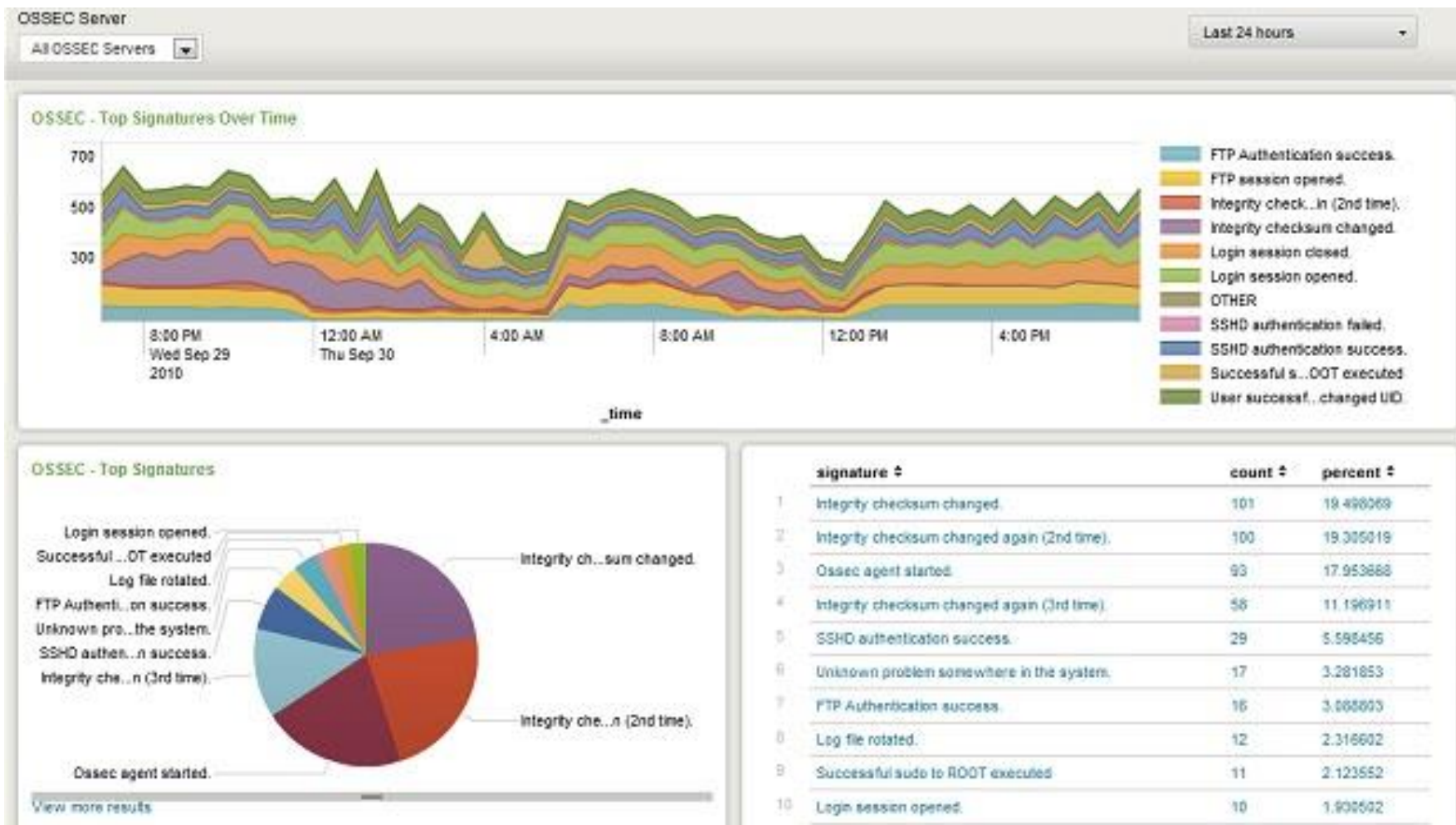
- +/etc/resolv.conf
- +/etc/mail/aliases.db
- +/etc/rc.local
- +/etc/ld.so.cache
- +/etc/group

### Latest events

Level:	3 - Ossec server started.	2018 Nov 07 08:55:39
Rule Id:	502	
Location:	Node1->ossec-monitord	
ossec: Ossec started.		
Level:	3 - New ossec agent connected.	2018 Nov 07 08:55:34
Rule Id:	501	
Location:	(Node2) 192.168.43.193->ossec	
ossec: Agent started: 'Node2->192.168.43.193'.		
Level:	7 - Integrity checksum changed.	2018 Nov 07 07:36:36
Rule Id:	550	
Location:	Node1->syscheck	
Integrity checksum changed for: '/etc/resolv.conf' Old md5sum was: '359e8b08f3de686150fb76121b185f3' New md5sum is: 'ffa171aba012e63354e9956f91541eae' Old sha1sum was: '1fb3d5b2f0bc4b5f81101ec934d7481b2f7c7465' New sha1sum is: 'd03743d2090c9a9b4099eb13124ce876eaf9ac10'		

# Examples of free modern IDS/IPS

- **Splunk:** Free and premium versions available; covers HIDS+NIDS



# Examples of free modern IDS/IPS

- **Snort:** Open-source NIDS, old and common, single-threaded

Services / Snort / Alerts ?

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Clear all interface log files

### Alert Log View Settings

Interface to Inspect:  ☐ Auto-refresh view    
Choose interface.. Alert lines to display.

Alert Log Actions:

### Alert Log View Filter

+

### Last 1000 Alert Log Entries

Date	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	SID	Description
2017-07-23 20:49:52	1	UDP	A Network Trojan was Detected	66.240.205.34 Q ⊕	1066	 Q ⊕	16464	1:31136 ⊕ ✖	MALWARE-CNC Win.Trojan.ZeroAccess inbound connection
2017-07-22 06:15:49	2	UDP	Potentially Bad Traffic	163.172.17.76 Q ⊕	54465	 Q ⊕	5060	140:26 ⊕ ✖	(spp_sip) Method is unknown
2017-07-21 09:26:30	2	UDP	Potentially Bad Traffic	163.172.22.169 Q ⊕	52428	 Q ⊕	5060	140:26 ⊕ ✖	(spp_sip) Method is unknown
2017-07-21 01:03:28	2	UDP	Potentially Bad Traffic	163.172.17.76 Q ⊕	46834	 Q ⊕	5060	140:26 ⊕ ✖	(spp_sip) Method is unknown
2017-07-20 20:36:37	2	UDP	Potentially Bad Traffic	163.172.22.169 Q ⊕	54788	 Q ⊕	5060	140:26 ⊕ ✖	(spp_sip) Method is unknown
2017-07-20 08:31:30	2	UDP	Potentially Bad Traffic	163.172.17.76 Q ⊕	59571	 Q ⊕	5060	140:26 ⊕ ✖	(spp_sip) Method is unknown



# Examples of free modern IDS/IPS

- **Suricata:** Open-source NIDS, multi-threaded, bit fancier

The screenshot displays the EveBox web interface in a Mozilla Firefox browser. The address bar shows the URL `10.44.100.50:5636/#/inbox`. The interface includes a navigation menu with 'EveBox', 'Inbox', 'Escalated', 'Alerts', 'Events', and 'Reports'. A 'Last 24 hours' filter is selected, and the page shows 'Showing 1-78 of 78' alerts. A table of alerts is displayed with columns for '#', 'Timestamp', 'Source / Dest', and 'Signature'. Each alert row has an 'Archive' button. The alerts include various signatures such as 'ET CINS Active Threat Intelligence Poor Reputation IP group 81', 'ET SCAN Potential SSH Scan', 'ET P2P BitTorrent DHT ping request', 'ET DROP Dshield Block Listed Source group 1', 'ET SCAN Suspicious inbound to MSSQL port 1433', 'ET CINS Active Threat Intelligence Poor Reputation IP group 98', 'ET SCAN Suspicious inbound to MySQL port 3306', 'ET COMPROMISED Known Compromised or Hostile Host Traffic group 10', and 'ET SCAN Suspicious inbound to MSSQL port 1433'.

#	Timestamp	Source / Dest	Signature
4	2018-06-30 16:03:49 a minute ago	S: 92.63.197.18 D: 10.44.100.50	ET CINS Active Threat Intelligence Poor Reputation IP group 81
46	2018-06-30 16:02:55 2 minutes ago	S: 112.85.42.148 D: 10.44.100.50	ET SCAN Potential SSH Scan
2	2018-06-30 15:59:03 6 minutes ago	S: 10.44.100.235 D: 234.51.34.227	ET P2P BitTorrent DHT ping request
2	2018-06-30 15:59:02 6 minutes ago	S: 181.214.87.225 D: 10.44.100.50	ET DROP Dshield Block Listed Source group 1
2	2018-06-30 15:58:19 6 minutes ago	S: 14.157.159.75 D: 10.44.100.50	ET SCAN Suspicious inbound to MSSQL port 1433
2	2018-06-30 15:54:49 10 minutes ago	S: 107.170.255.53 D: 10.44.100.50	ET CINS Active Threat Intelligence Poor Reputation IP group 98
2	2018-06-30 15:54:25 10 minutes ago	S: 218.60.67.79 D: 10.44.100.50	ET SCAN Suspicious inbound to MySQL port 3306
2	2018-06-30 15:53:23 11 minutes ago	S: 185.8.49.228 D: 10.44.100.50	ET COMPROMISED Known Compromised or Hostile Host Traffic group 10
2	2018-06-30 15:53:19 11 minutes ago	S: 211.239.113.60 D: 10.44.100.50	ET SCAN Suspicious inbound to MSSQL port 1433
2	2018-06-30 15:51:02 14 minutes ago	S: 10.44.100.235 D: 233.23.34.71	ET P2P BitTorrent DHT ping request



# Outline

Understanding intruders

Intrusion detection system (IDS)

Intrusion prevention systems (IPS)

Detection theory

Firewalls

# Problem: We're not sure

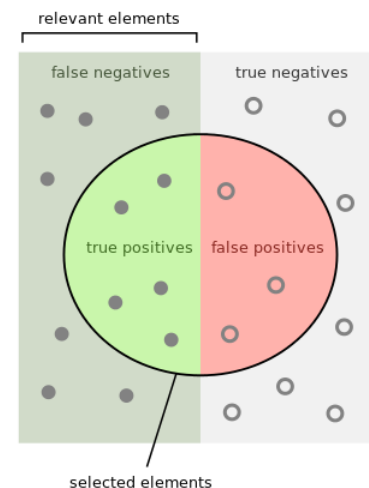
- We might say it's **malicious** and we're **right** (**True positive**)  
We detected bad stuff and did something about it! Yay! 😊
- We might say it's **malicious** but we're **wrong** (**False positive**)  
We blocked legitimate stuff! People are mad at us! ☹️
- We might say it's **benign** and we're **right** (**True negative**)  
That traffic is cool and good, let it through! Yeah! 😊
- We might say it's **benign** and we're **wrong** (**False negative**)  
We missed an attack! Oh no, danger! ☹️



# Confusion Matrix

- A Confusion matrix is a table describing the performance of some detection algorithm
  - True positives (TP): number of correct classifications of malware
  - True negatives (TN): number of correct classifications of non-malware
  - False positives (FP): number of incorrect classifications of non-malware as malware
  - False negatives (FN): number of incorrect classifications of malware as non-malware

<i>Detection Result</i>		T	F
<i>Reality</i>	T	True Positive	False Negative
	F	False Positive	True Negative



[https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall)

# Metrics

(from perspective of detector)

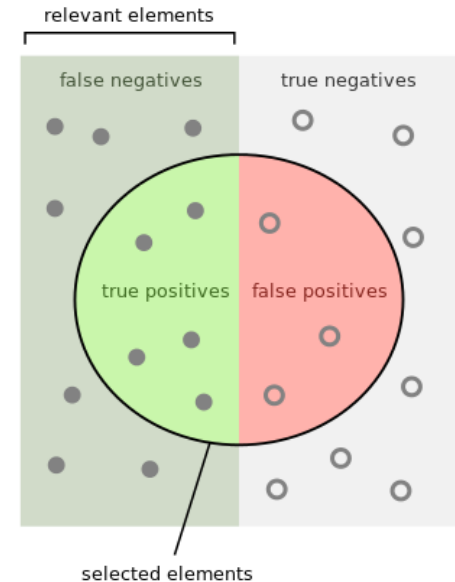
- **False positive rate:**  $FPR = \frac{FP}{FP + TN} = \frac{\# \text{ benign marked as malicious}}{\text{total benign}}$
- **True negative rate:**  $TNR = 1 - FPR = \frac{TN}{FP + TN} = \frac{\# \text{ benign unmarked}}{\text{total benign}}$
- **False negative rate:**  $FNR = \frac{FN}{FN + TP} = \frac{\# \text{ malicious not marked}}{\text{total malicious}}$
- **True positive rate:**  $TPR = 1 - FNR = \frac{TP}{FN + TP} = \frac{\# \text{ malicious correctly marked}}{\text{total malicious}}$

Shorthand for this part  
“**Alert**” = mark as malicious  
“**Malware**” = packet is malicious

		<i>Detection Result</i>	
		T	F
<i>Reality</i>	T	True Positive	False Negative
	F	False Positive	True Negative

# Precision and Recall

- **Recall** (also known as sensitivity)
  - fraction of correct instances among all instances that actually are positive (malware)
  - $TP / (TP + FN)$ 
    - ^ Note: This is also the TPR
- **Precision**
  - fraction of correct instances (malware) that algorithm believes are positive (malware)
  - $TP / (TP + FP)$



How many selected items are relevant?

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

[https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall)

**Recall:** percent of malware you alert on  
**Precision:** percent alerts that are right

# Bayes Rule

- $\Pr(x)$  function, probability of event  $x$ 
  - $\Pr(\text{sunny}) = 0.8$  (80% of sunny day)
- Conditional probability
  - $\Pr(x|y)$ , probability of  $x$  *given*  $y$
  - $\Pr(\text{cavity}|\text{toothache}) = 0.6$
  - 60% chance of cavity given you have a toothache
- Bayes' Rule (of conditional probability)

Bayes rule of conditional probability

$$\Pr(B|A) = \frac{\Pr(A|B) \cdot \Pr(B)}{\Pr(A)}$$

Example:

- Assume:  $\Pr(\text{cavity}) = 0.5$ ,  $\Pr(\text{toothache}) = 0.1$
- What is  $\Pr(\text{toothache}|\text{cavity})$ ?
  - $= \Pr(\text{cavity}|\text{toothache}) * \Pr(\text{toothache}) / \Pr(\text{cavity})$   
 $= 0.6 * 0.1 / 0.5$   
 $= 0.12$

# Base Rate Fallacy

- Occurs when assessing  $P(X|Y)$  without considering probability of  $X$  and the total probability of  $Y$

Example:

- Base rate of malware is 1 packet in a 10,000  $\rightarrow \text{Pr}(\text{Malware}) = 0.00001$
- Intrusion detection system is 99% accurate  $\rightarrow \text{Pr}(\text{Alert} | \text{Malware}) = 0.99$
- 1% false positive rate (alert on benign)  $\rightarrow \text{Pr}(\text{Alert} | \text{!Malware}) = 0.01$
- 1% false negative rate (fail to alert on malicious)  $\rightarrow \text{Pr}(\text{!Alert} | \text{Malware}) = 0.01$
- A packet is marked by the NIDS as malware.  
What is the probability that packet  $X$  actually is malware?  
 $\rightarrow \text{Pr}(\text{Malware} | \text{Alert})$
- This is the **precision**: the rate at which an alert is actually true.  
("How often was alerting someone actually justified?")

# Base Rate Fallacy

- Our goal is to find the true alert rate (i.e.,  $\Pr(\text{Malware}|\text{Alert})$ ) using Bayes rule:

$$\Pr(\text{Malware}|\text{Alert}) = \frac{\Pr(\text{Alert}|\text{Malware}) * \Pr(\text{Malware})}{\Pr(\text{Alert})}$$

- We know:
  - 1% false positive rate (benign marked as malicious 1% of the time); TNR= 99%
  - 1% false negative rate (malicious marked as benign 1% of the time); TPR= 99%
  - Base rate of malware is 1 packet in 10,000

- Let's figure the ingredients to this equation...

- $\Pr(\text{Alert}|\text{Malware}) = ?$  **TPR = 0.99**
- $\Pr(\text{Malware}) = ?$  **Base rate = 0.0001**
- $\Pr(\text{Alert}) = ?$  **0.01**

$$\begin{aligned}\Pr(\text{Alert}|\text{Malware}) &= \frac{\# \text{ malicious correctly marked}}{\text{total malicious}} \\ &= \frac{TP}{FN + TP} = TPR\end{aligned}$$

$$\begin{aligned}\Pr(\text{Alert}) &= \Pr(\text{Alert}|\text{Malware}) * \Pr(\text{Malware}) + \Pr(\text{Alert}|\neg \text{Malware}) * \Pr(\neg \text{Malware}) \\ &= (0.99 * 0.0001) + (0.01 * 0.9999) = 0.01\end{aligned}$$

# Base rate fallacy ...

- Now let's plug into the Bayes rule formula:

$$\Pr(\text{Malware}|\text{Alert}) = \frac{\Pr(\text{Alert}|\text{Malware}) * \Pr(\text{Malware})}{\Pr(\text{Alert})}$$

- Using these ingredients:

- $\Pr(\text{Alert}|\text{Malware}) = 0.99$
- $\Pr(\text{Malware}) = 0.0001$
- $\Pr(\text{Alert}) = 0.01$

$$= \frac{0.99 \cdot 0.0001}{0.01} = 0.0099$$



- A little less than 1% of alarms are actually malware!
- What does this mean for network administrators?

Almost all the stupid  
alerts are LIES!!!!



# All the math in one place

Name:	Base rate			Recall		Precision		
Prob:	$P(M)$	$P(A !M)$	$P(!A M)$	$P(A M)$	$P(!A !M)$	$P(M A)$	$P(A)$	$P(!M)$
Rate:	BR	FPR	FNR	TPR	TNR			
Eqn:		$FP/(FP+TN)$	$FN/(FN+TP)$	$\frac{1-FNR}{TP/(FN+TP)} =$	$\frac{1-FPR}{TN/(FP+TN)} =$	$\frac{TP/(TP+FP)}{P(A M)*P(M)/P(A)} =$	$P(A M)*P(M) + P(A !M)*P(!M)$	$1-BR$
<b>A</b>	0.0001	0.01	0.01	0.9900	0.9900	0.0098	0.0101	0.9999
<b>B</b>	0.0001	0.01	0.0001	0.9999	0.9900	0.0099	0.0101	0.9999
<b>C</b>	0.0001	0.0001	0.01	0.9900	0.9999	0.4975	0.0002	0.9999
<b>D</b>	0.0001	0.0001	0.0001	0.9999	0.9999	0.5000	0.0002	0.9999

Four possible situations

Note: You can access this spreadsheet – [it's here](#).



# Which variable matters most? (1)

Name:	Base rate			Recall		Precision		
Prob:	$P(M)$	$P(A !M)$	$P(!A M)$	$P(A M)$	$P(!A !M)$	$P(M A)$	$P(A)$	$P(!M)$
Rate:	BR	FPR	FNR	TPR	TNR			
Eqn:		$FP/(FP+TN)$	$FN/(FN+TP)$	$\frac{1-FNR}{TP/(FN+TP)}$	$\frac{1-FPR}{TN/(FP+TN)}$	$\frac{TP/(TP+FP)}{P(A M)*P(M)/P(A)}$	$P(A M)*P(M) + P(A !M)*P(!M)$	$1-BR$
A	0.0001	0.01	0.01	0.9900	0.9900	0.0098	0.0101	0.9999
B	0.0001	0.01	0.0001	0.9999	0.9900	0.0099	0.0101	0.9999
C	0.0001	0.0001	0.01	0.9900	0.9999	0.4975	0.0002	0.9999
D	0.0001	0.0001	0.0001	0.9999	0.9999	0.5000	0.0002	0.9999

Making this better...

...didn't help much ☹️

# Which variable matters most? (2)

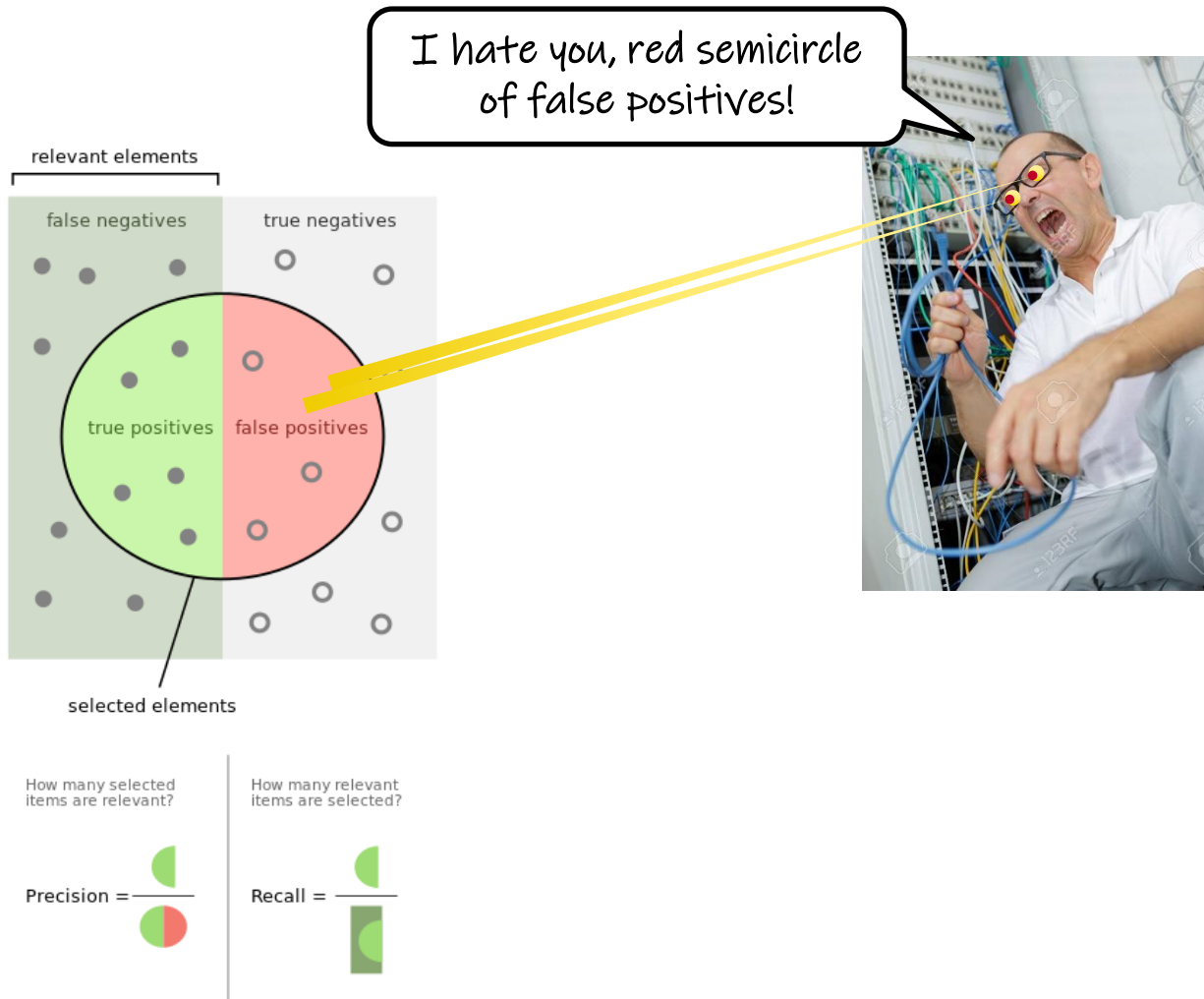
Name:	Base rate			Recall		Precision		
Prob:	$P(M)$	$P(A !M)$	$P(!A M)$	$P(A M)$	$P(!A !M)$	$P(M A)$	$P(A)$	$P(!M)$
Rate:	BR	FPR	FNR	TPR	TNR			
Eqn:		$FP/(FP+TN)$	$FN/(FN+TP)$	$1-FNR = TP/(FN+TP)$	$1-FPR = TN/(FP+TN)$	$TP/(TP+FP) = P(A M)*P(M)/P(A)$	$P(A M)*P(M) + P(A !M)*P(!M)$	$1-BR$
A	0.0001	0.01	0.01	0.9900	0.9900	0.0098	0.0101	0.9999
B	0.0001	0.01	0.0001	0.9999	0.9900	0.0099	0.0101	0.9999
C	0.0001	0.0001	0.01	0.9900	0.9999	0.4975	0.0002	0.9999
D	0.0001	0.0001	0.0001	0.9999	0.9999	0.5000	0.0002	0.9999

But making this better...

...helped a lot!! 😊

# Base Rate Fallacy conclusion

- In any detection system, you need a false positive rate as low or lower than the base rate, otherwise most alarms are incorrect!





# Outline

Understanding intruders

Intrusion detection system (IDS)

Intrusion prevention systems (IPS)

Detection theory

Firewalls

# Firewall Characteristics

## Design goals

All traffic from inside to outside, and vice versa, must pass through the firewall

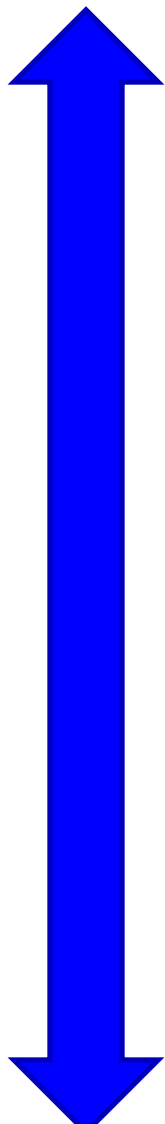
Only authorized traffic as defined by the local security policy will be allowed to pass

The firewall itself is immune to penetration



# Types of firewalls

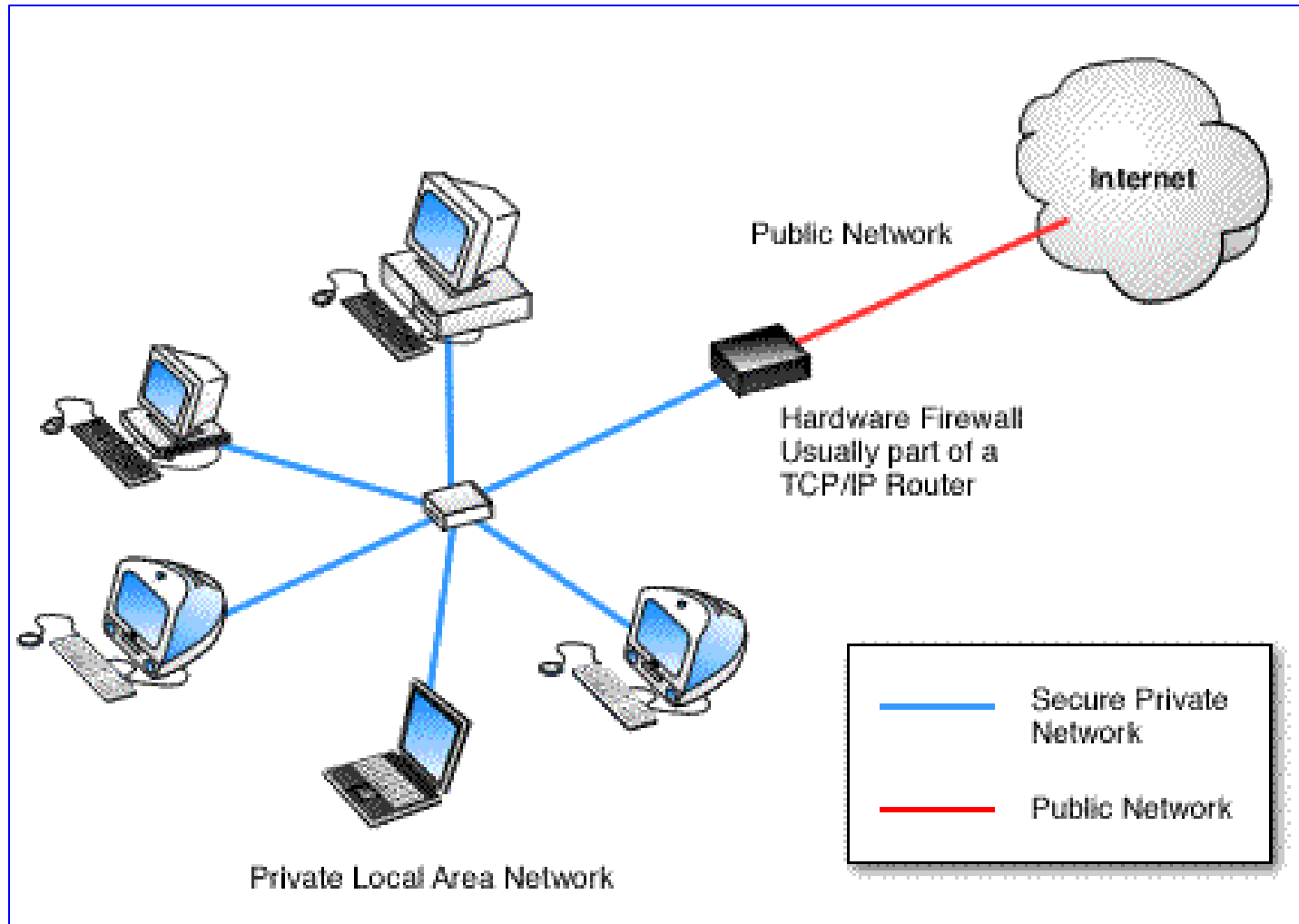
Simpler, less expressive, less resource-intensive



Type	Logic	Pros	Cons
<b>Packet filter</b>	Decide on per-packet basis	<ul style="list-style-type: none"><li>• Simple</li><li>• Fast</li><li>• Easy to configure</li></ul>	<ul style="list-style-type: none"><li>• Dumb</li><li>• Not very expressive</li></ul>
<b>Stateful packet inspection</b>	Decide on stream or higher level basis	<ul style="list-style-type: none"><li>• More expressive</li></ul>	<ul style="list-style-type: none"><li>• More resource intensive</li><li>• More configuration</li></ul>
<b>Application-level proxy</b>	Understands app-level traffic	<ul style="list-style-type: none"><li>• Can enforce app-relevant restrictions</li></ul>	<ul style="list-style-type: none"><li>• Need one customized for each app</li></ul>

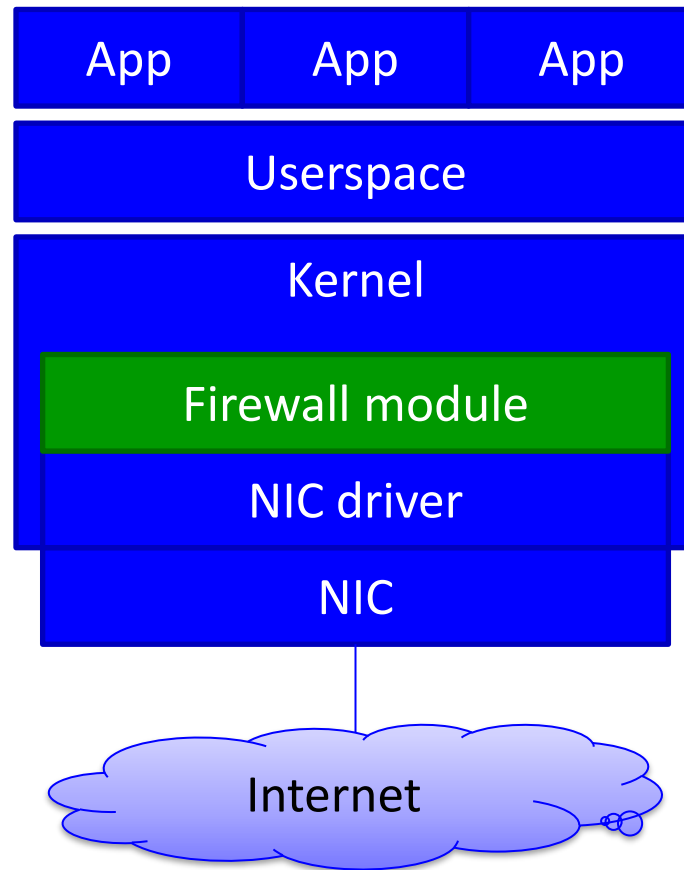
More complex, more expressive, more resource-intensive

# Placement of firewalls (1)



LAN firewall

# Placement of firewalls (2)



Host-based firewall



# Firewall Filter Characteristics

- Characteristics that a firewall access policy could use to filter traffic include:

**IP address  
and protocol  
values**

**This type of  
filtering is used by  
packet filter and  
stateful inspection  
firewalls**

**Typically used to  
limit access to  
specific services**

**Application  
protocol**

**This type of  
filtering is used by  
an application-  
level gateway that  
relays and  
monitors the  
exchange of  
information for  
specific  
application  
protocols**

**User  
identity**

**Typically for  
inside users who  
identify  
themselves using  
some form of  
secure  
authentication  
technology**

**Network  
activity**

**Controls access  
based on  
considerations  
such as the time or  
request, rate of  
requests, or other  
activity patterns**

# Limitations of firewalls

- Book spends a long time on this, but it's simple: **firewalls have human-built rules *and* can only deal with packets that go through them.**
- Two scenarios they don't help:
  - HTTP service has a vulnerability and firewall allows HTTP  
(Firewall set to allow the bad thing)
  - Firewall is at ISP uplink but rogue cell phone gets inside of LAN via WiFi  
(Firewall not traversed to do the bad thing)

# Packet Filtering Firewall

- Applies rules to each incoming and outgoing IP packet
  - Typically a list of rules based on matches in the IP or TCP header:
    - Source IP address
    - Destination IP address
    - Port numbers
    - Source and destination transport-level address
    - IP protocol field
    - Interface
- Two default policies:
  - DROP - prohibit unless expressly permitted
    - More conservative, controlled, visible to users
  - ACCEPT - permit unless expressly prohibited
    - Easier to manage and use but less secure

# Table 9.1

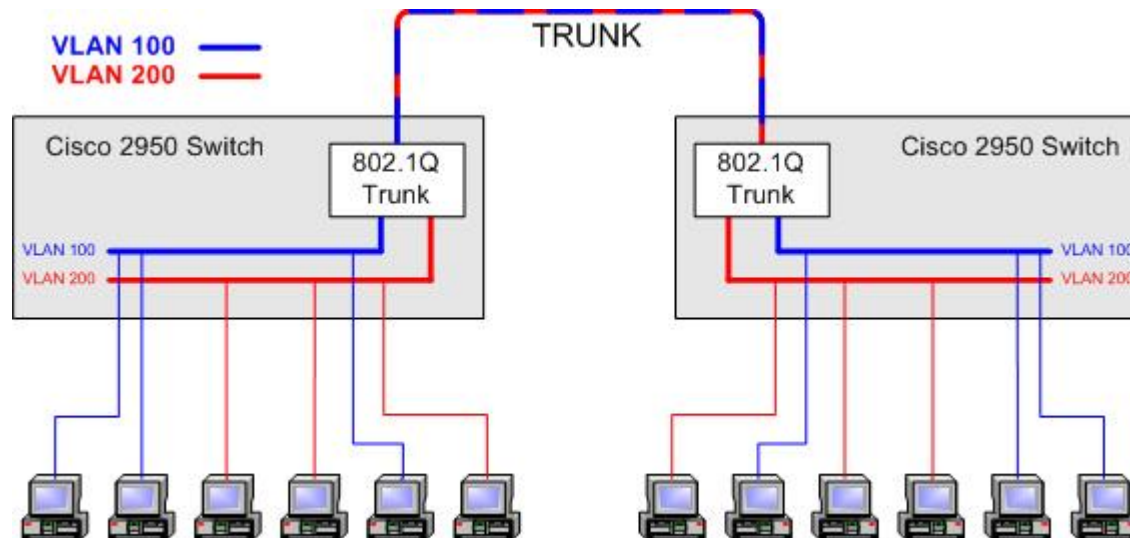
## Packet-Filtering Examples

Rule	Direction	Src address	Dest addresss	Protocol	Dest port	Action
1	In	External	Internal	TCP	25	Permit
2	Out	Internal	External	TCP	>1023	Permit
3	Out	Internal	External	TCP	25	Permit
4	In	External	Internal	TCP	>1023	Permit
5	Either	Any	Any	Any	Any	Deny

# Reminder: VLANs exist

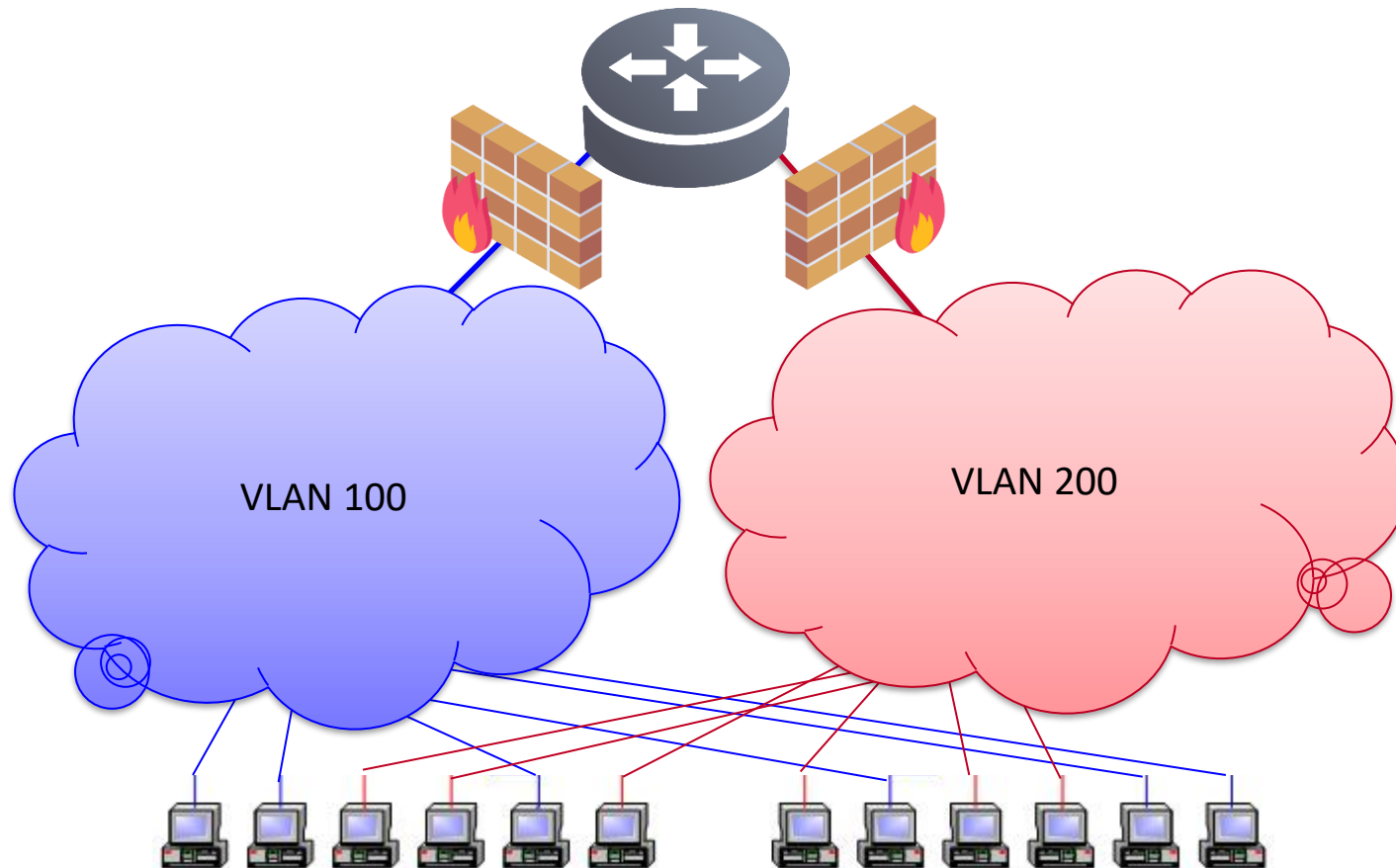
This slide is from way back in the network lecture

- Logically separate layer 2 networks
- Switch ports can be:
  - **Access ports:** can only see one VLAN, aren't aware of VLAN concept
  - **Trunk ports:** end point includes a VLAN tag in packet header to indicate which VLAN it wants to talk to; interprets such headers on incoming packets



# VLANs make it convenient to have firewall/NIDS/NIPS boundaries

- If two VLANs want to talk, it's via a router; that's a great place to put a firewall!





# Conclusion

## Understanding intruders

- Criminal/activist/state/other
- Skill level

## Intrusion detection systems (IDS)

- Look for anomalies or signatures, log/alert accordingly
- Either host-based or network-based

## Intrusion prevention system (IPS)

- It's an IDS but it takes action

## Detection theory

- Need false positive rate  $\leq$  base rate, otherwise most alerts are wrong

## Firewalls

- Block traffic based on rules