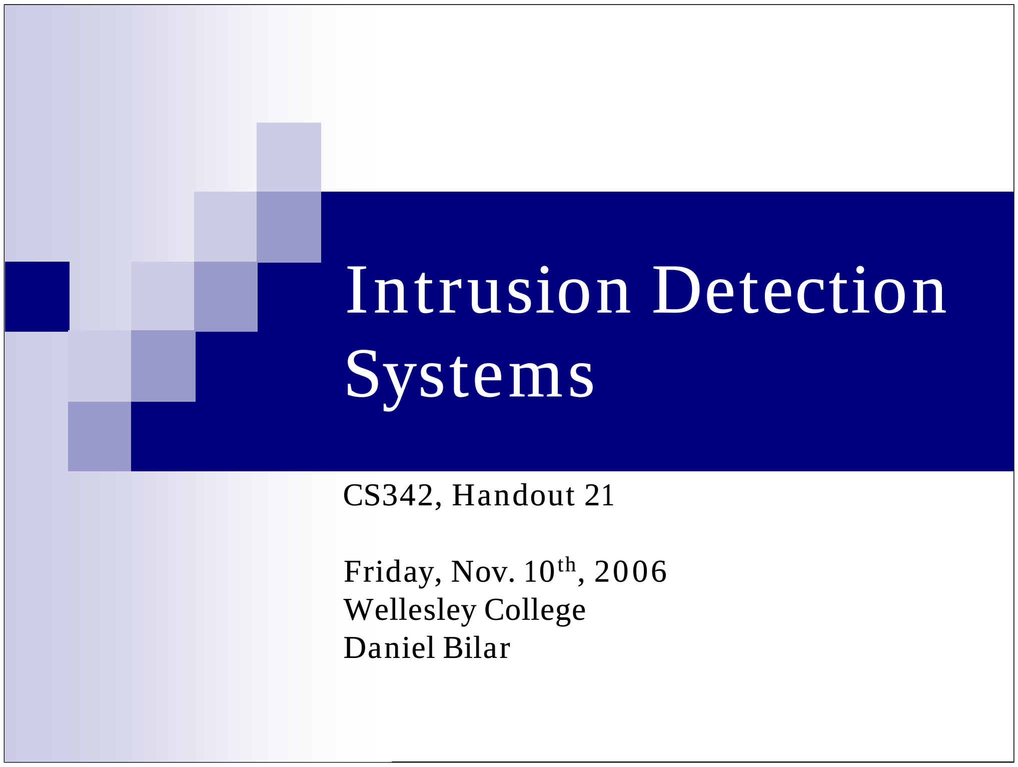
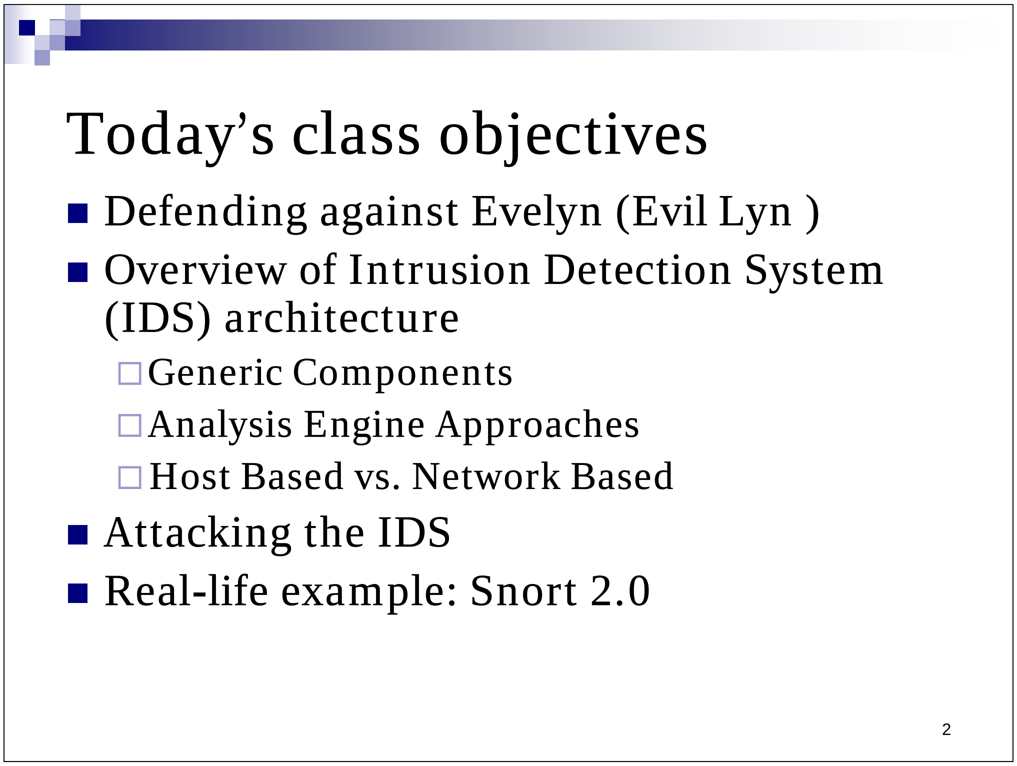


Intrusion Detection Systems

CS342, Handout 21

Friday, Nov. 10th, 2006
Wellesley College
Daniel Bilar



Today’s class objectives

- Defending against Evelyn (Evil Lyn)
- Overview of Intrusion Detection System (IDS) architecture
  - □ Generic Components
  - □ Analysis Engine Approaches
  - □ Host Based vs. Network Based
- Attacking the IDS
- Real-life example: Snort 2.0

Perspective switch

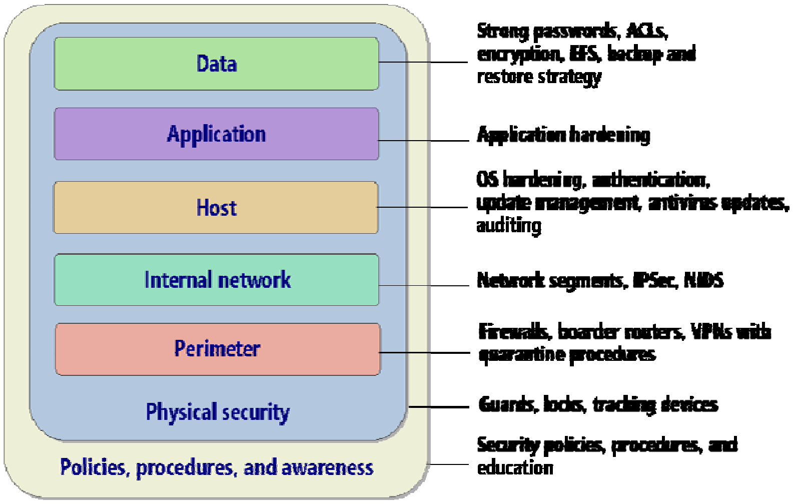
- In the past couple of lectures and problem sets, we were wearing our ‘attacker hats’

- We will switch to defense for the next coming lectures
  - □ Intrusion Detection Systems
  - □ Firewalls
  - □ The Law

Guiding principle for defense is “Defense-in-Depth”

Using a **layered approach** to increase an attacker’s risk of detection and reduce an attacker’s chance of success

Defemse-in-Depth



Data — Strong passwords, ACLs, encryption, EFS, backup and restore strategy

Application — Application hardening

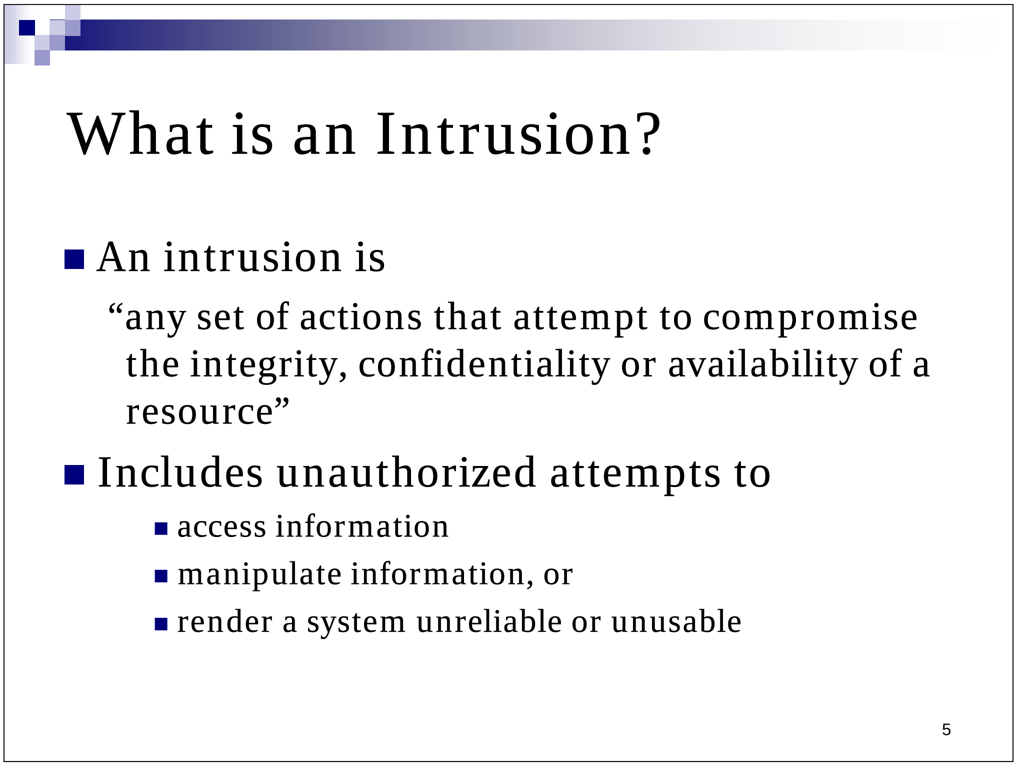
Host — OS hardening, authentication, update management, antivirus updates, auditing

Internal network — Network segments, IPSec, NIDS

Perimeter — Firewalls, boarder routers, VPNs with quarantine procedures

Physical security — Guards, locks, tracking devices

Policies, procedures, and awareness — Security policies, procedures, and education

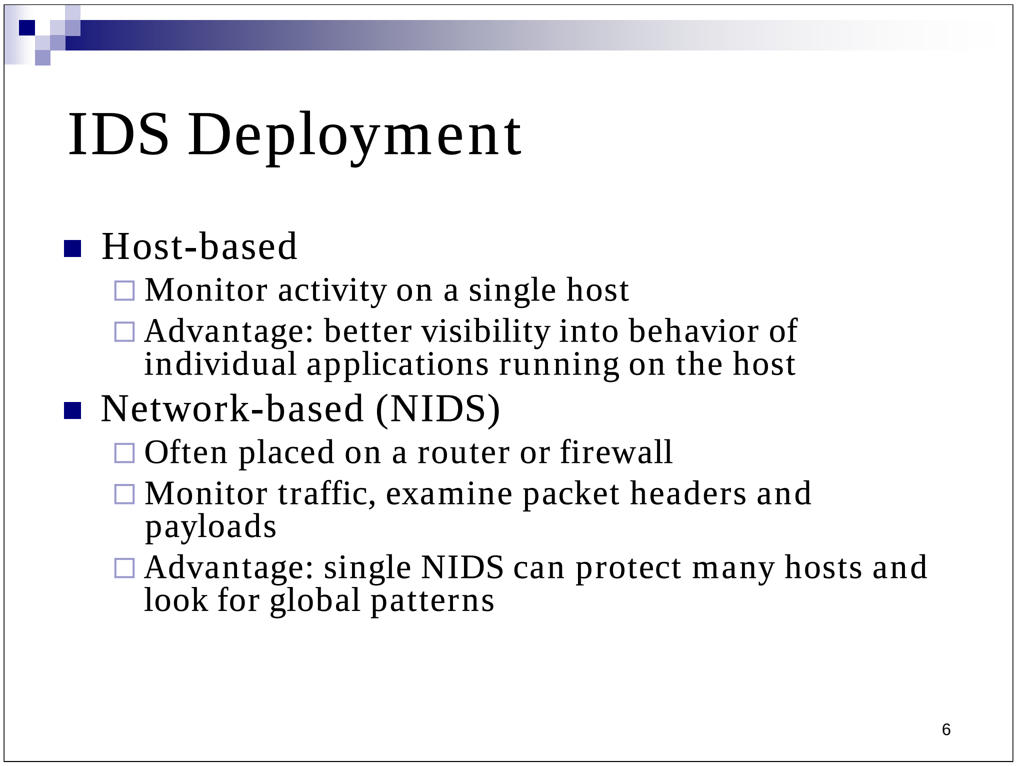


What is an Intrusion?

- An intrusion is

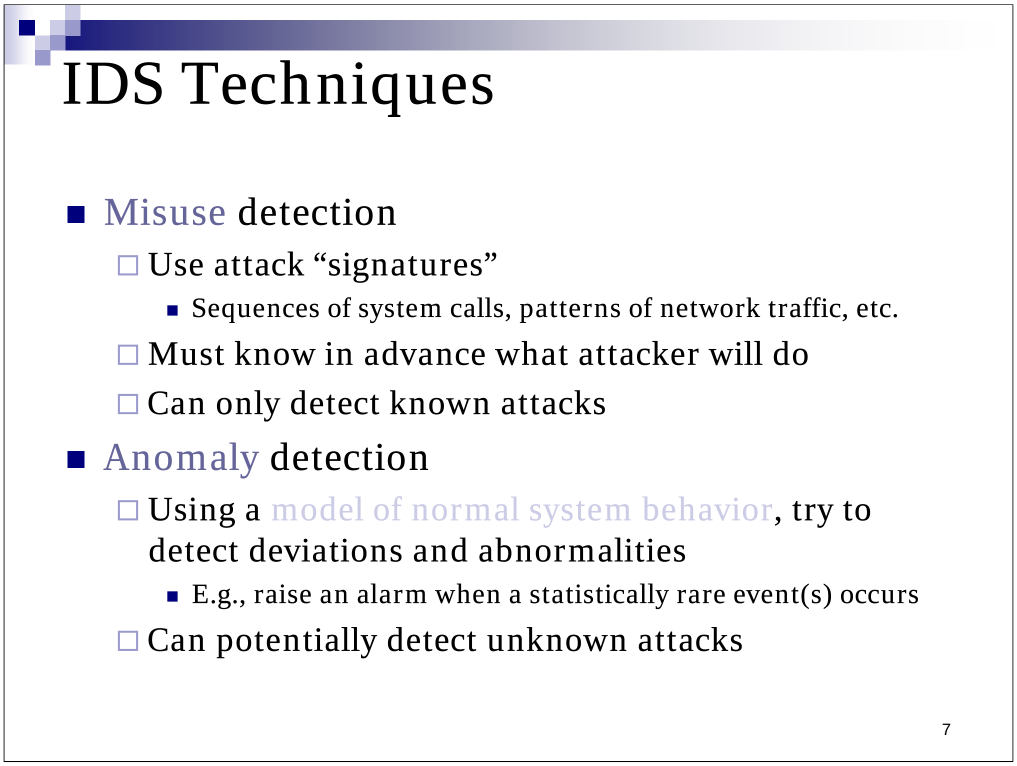
  “any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource”

- Includes unauthorized attempts to
  - access information
  - manipulate information, or
  - render a system unreliable or unusable



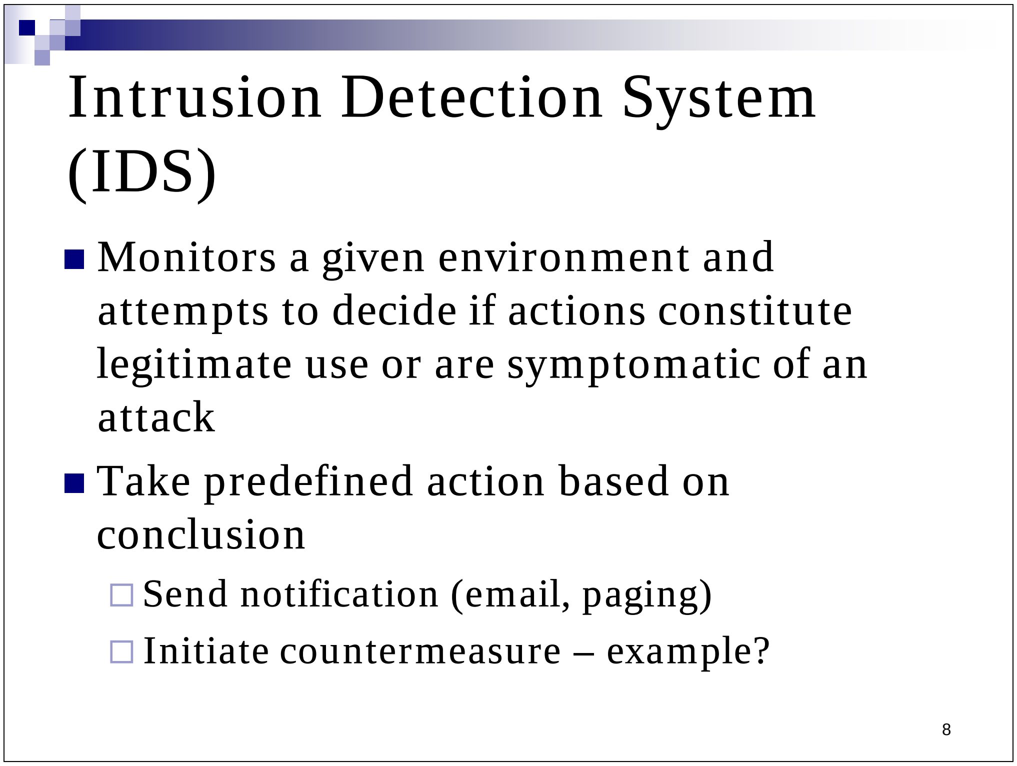
IDS Deployment

- Host-based
  - □ Monitor activity on a single host
  - □ Advantage: better visibility into behavior of individual applications running on the host
- Network-based (NIDS)
  - □ Often placed on a router or firewall
  - □ Monitor traffic, examine packet headers and payloads
  - □ Advantage: single NIDS can protect many hosts and look for global patterns



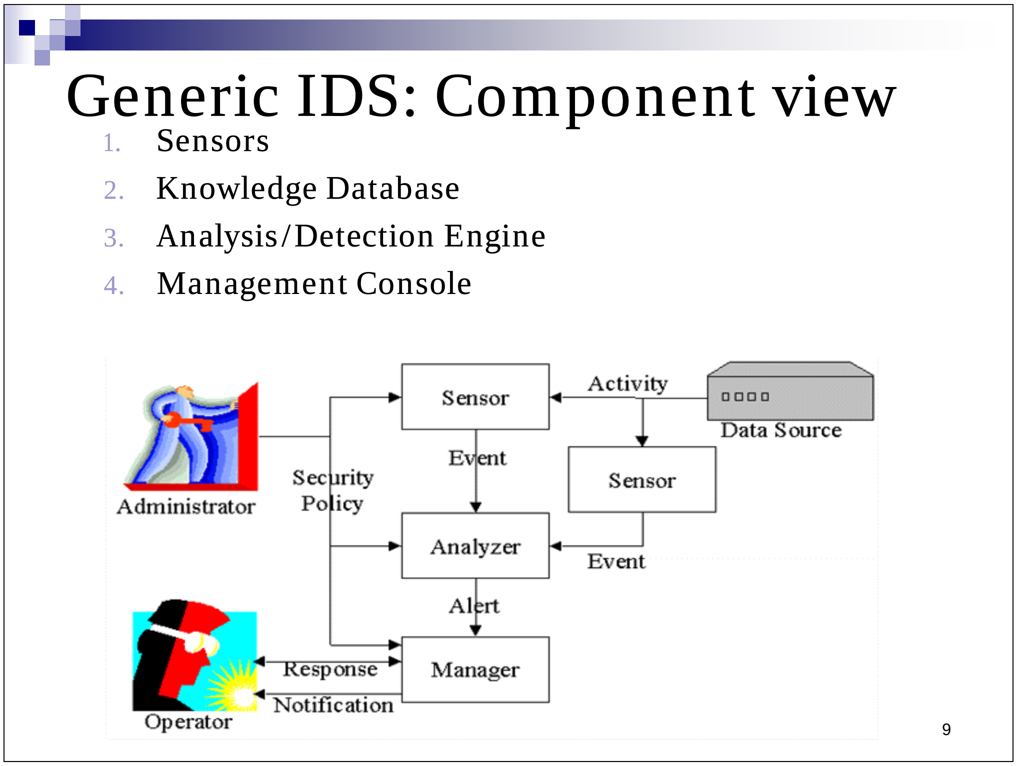
IDS Techniques

- Misuse detection
  - □ Use attack “signatures”
    - Sequences of system calls, patterns of network traffic, etc.
  - □ Must know in advance what attacker will do
  - □ Can only detect known attacks
- Anomaly detection
  - □ Using a model of normal system behavior, try to detect deviations and abnormalities
    - E.g., raise an alarm when a statistically rare event(s) occurs
  - □ Can potentially detect unknown attacks



Intrusion Detection System (IDS)

- Monitors a given environment and attempts to decide if actions constitute legitimate use or are symptomatic of an attack
- Take predefined action based on conclusion
  - □ Send notification (email, paging)
  - □ Initiate countermeasure – example?



Generic IDS: Component view

1. Sensors
2. Knowledge Database
3. Analysis/Detection Engine
4. Management Console

From http://www.sans.org/rr/intrusion/interop.php

The diagram above illustrates the terms described below and their relationships. Not every IDS will have all of these separate components exactly as shown. Some IDSs will combine these components into a single module; some will have multiple instances of these modules.

Sensor: The ID component that collects data about activity from data sources, detects events, and forwards them to the analyzer.

Activities: Activities are elements of the data source that are identified by the sensor or analyzer as being of interest to the operator. Examples of this include network session showing unexpected telnet activity, operating system log file entries showing a user attempting to access files to which he or she is not authorized to have access, etc.

Event: Activity that is detected by the sensor and which may result in an IDMEF alert being transmitted. For example, 'N' failed logins in 'T' seconds might indicate a brute-force login attack.

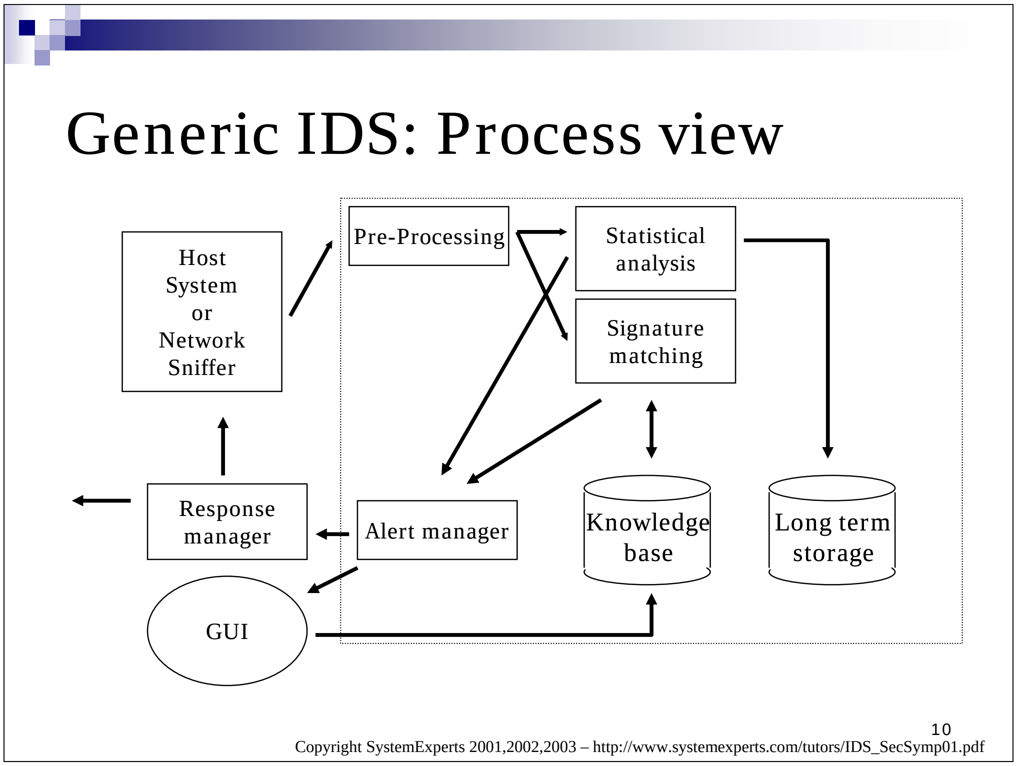
Analyzer: The ID component that analyses the events and according to the security policy possibly generates alerts based on these events. Alerts are formatted and transferred to managers using the IDMEF format over IDXP (optional) transfer protocol. In many existing IDSs, the sensor and the analyzer are part of the same component.

Alert: A message from an analyzer to a manager that an event of interest has been detected. An alert typically contains information about the unusual activity that was detected, as well as the specifics of the occurrence.

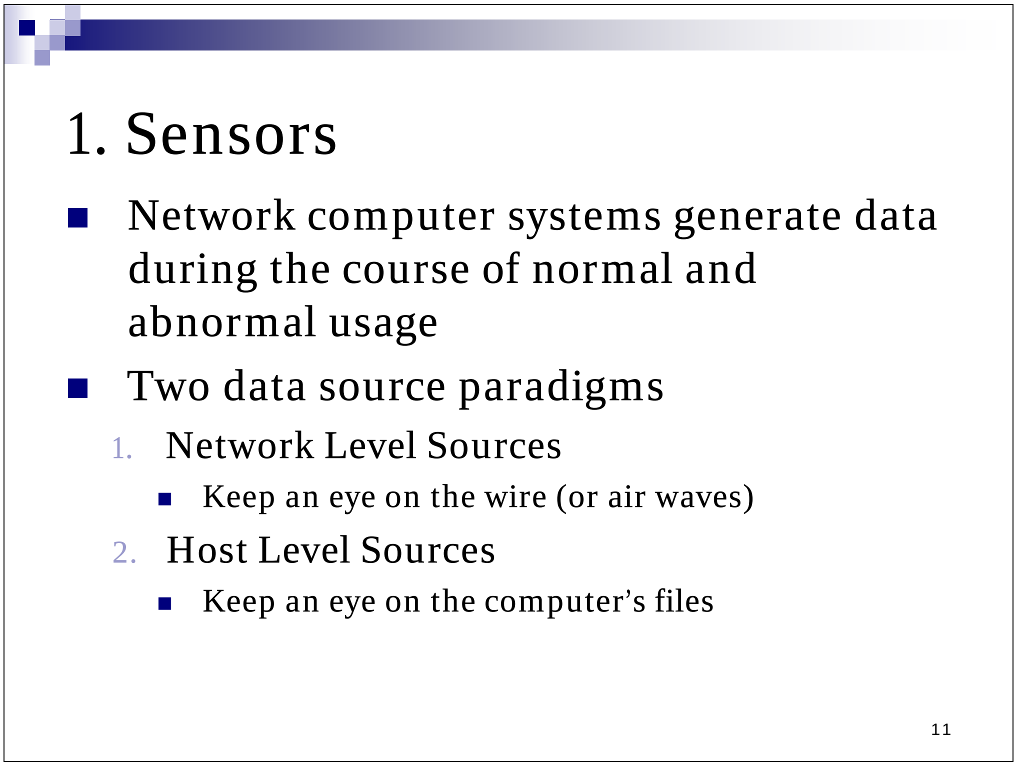
Manager: The ID component or process from which the operator manages the various components of the ID system. Management functions typically include (but are not limited to) sensor configuration, analyzer configuration, event notification management, data consolidation, and reporting. Managers inform the operator through different types of notification that alerts have occurred, as per the security policy.

Administrator: The human with overall responsibility for setting the security policy of the organization including decisions about deploying and configuring the IDSs.

Operator: The human that is primary user of the IDS manager for initiating responses to alerts and notifications

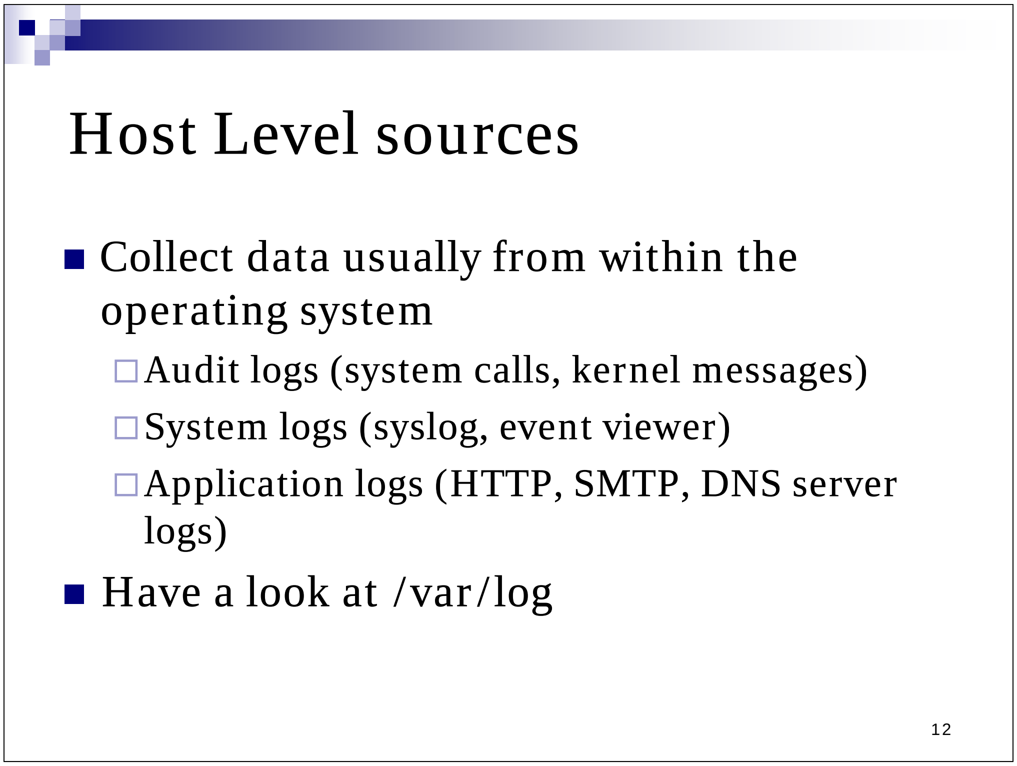


Generic IDS: Process view



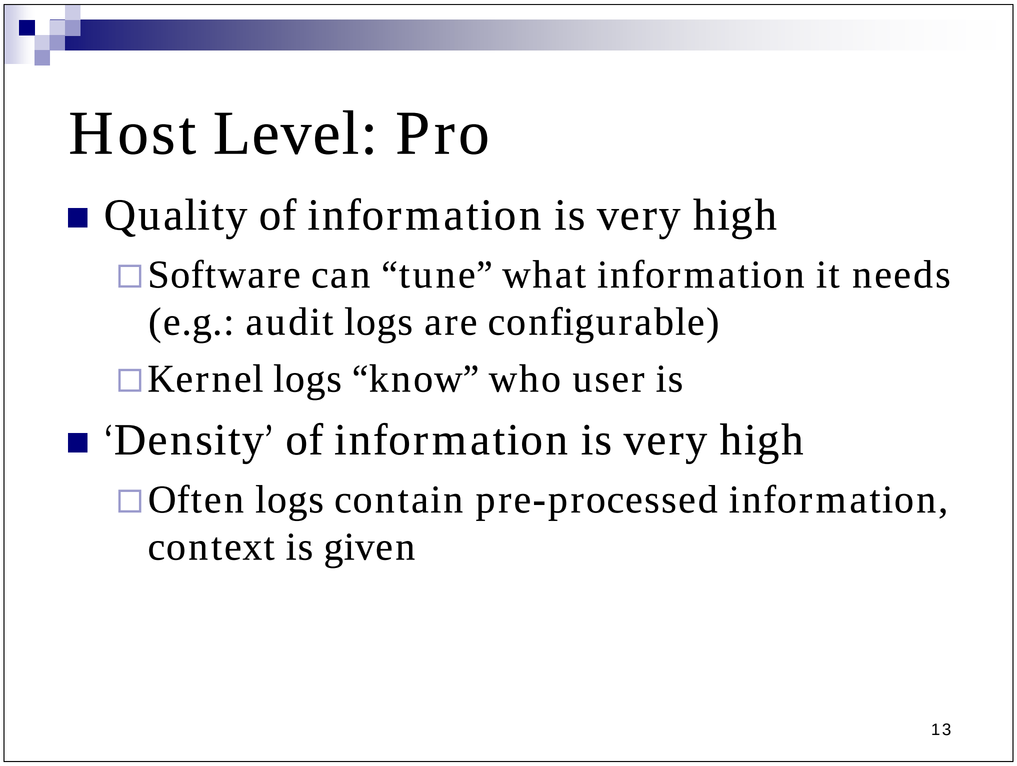
1. Sensors

- Network computer systems generate data during the course of normal and abnormal usage
- Two data source paradigms
  1. Network Level Sources
     - Keep an eye on the wire (or air waves)
  2. Host Level Sources
     - Keep an eye on the computer's files



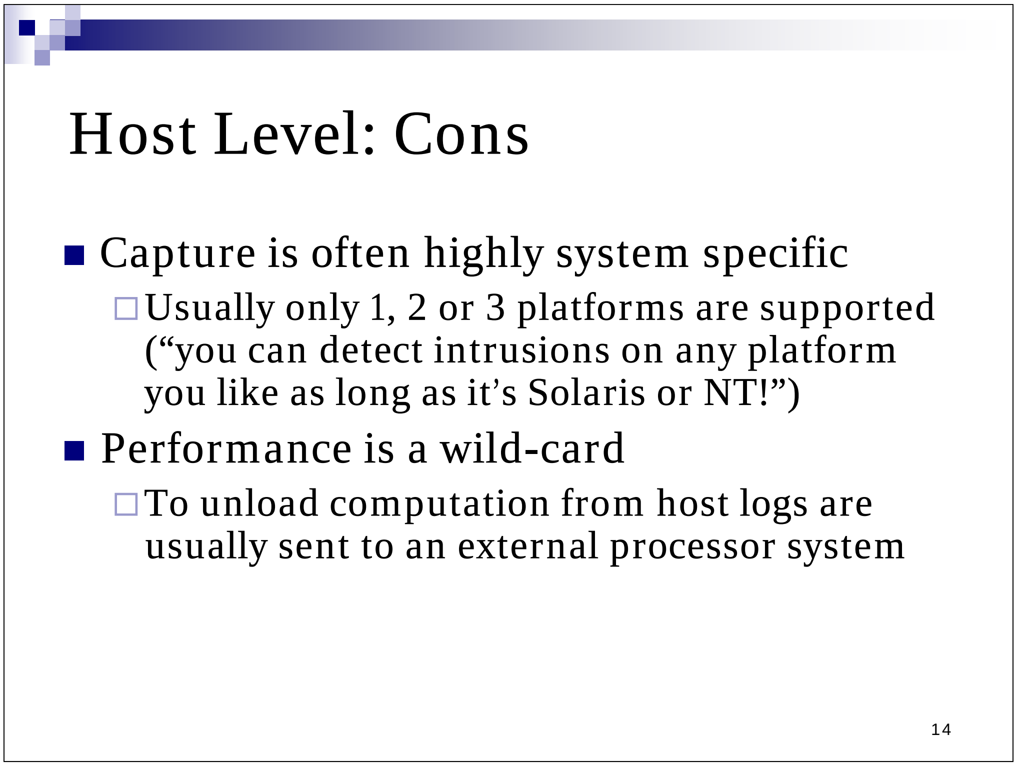
Host Level sources

- Collect data usually from within the operating system
  - □ Audit logs (system calls, kernel messages)
  - □ System logs (syslog, event viewer)
  - □ Application logs (HTTP, SMTP, DNS server logs)
- Have a look at /var/log



Host Level: Pro

- Quality of information is very high
  - □ Software can "tune" what information it needs (e.g.: audit logs are configurable)
  - □ Kernel logs "know" who user is
- 'Density' of information is very high
  - □ Often logs contain pre-processed information, context is given

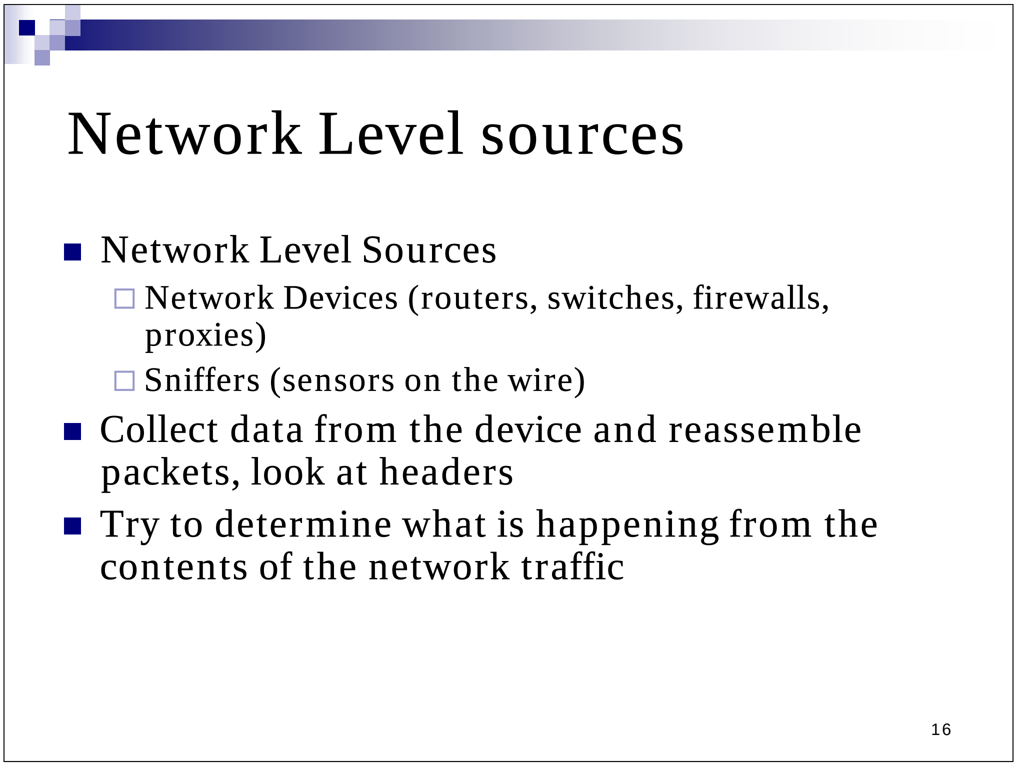


Host Level: Cons

- Capture is often highly system specific
  - □ Usually only 1, 2 or 3 platforms are supported (“you can detect intrusions on any platform you like as long as it’s Solaris or NT!”)
- Performance is a wild-card
  - □ To unload computation from host logs are usually sent to an external processor system

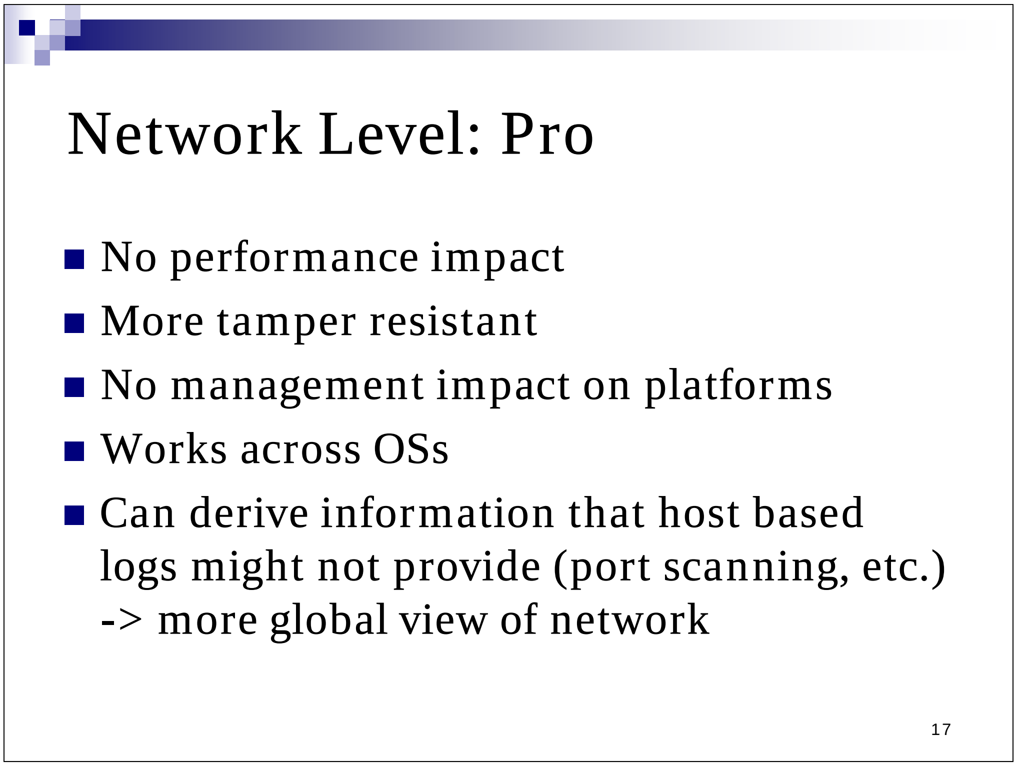
Host Level: Cons *(cont)*

- Hosts are often the target of attack
  - □ If they are compromised their logs may be subverted
  - □ Data sent to the IDS may be corrupted
  - □ If the IDS runs on the host itself it may be subverted
- Only **local** view of the attack



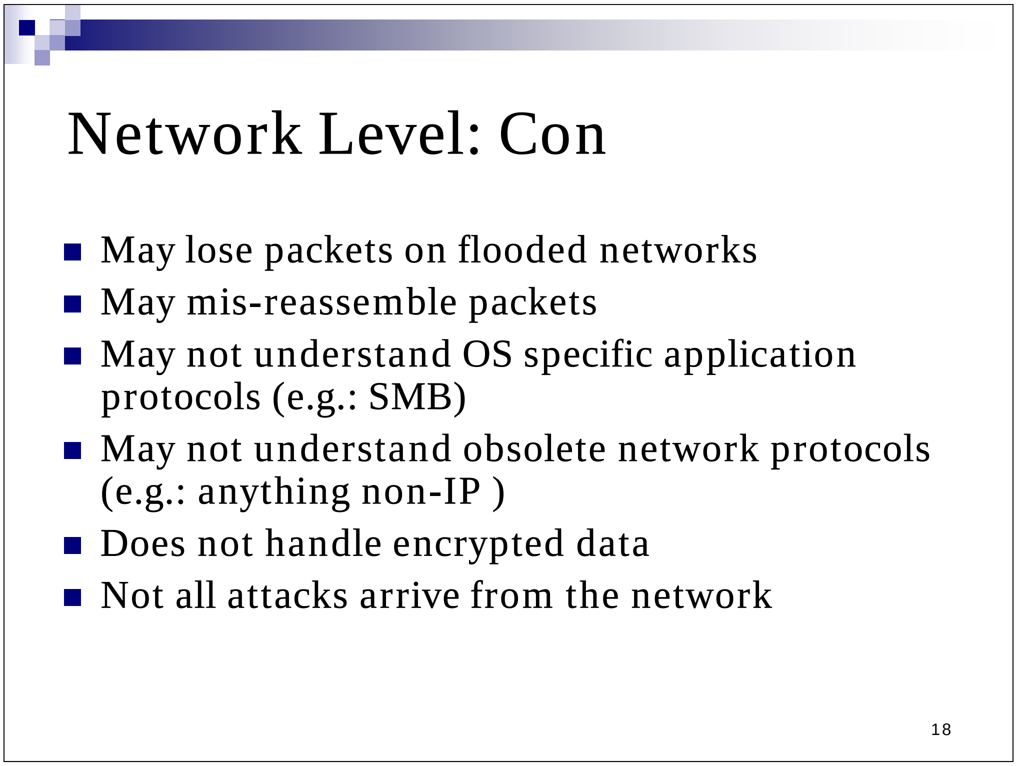
Network Level sources

- Network Level Sources
  - Network Devices (routers, switches, firewalls, proxies)
  - Sniffers (sensors on the wire)
- Collect data from the device and reassemble packets, look at headers
- Try to determine what is happening from the contents of the network traffic



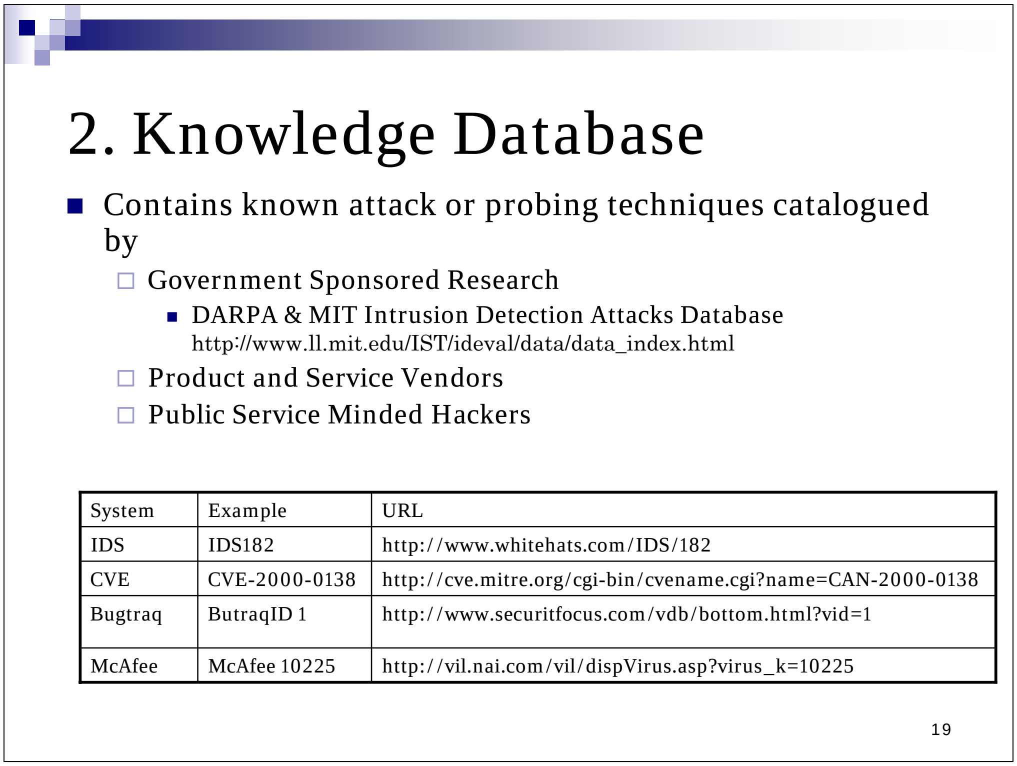
Network Level: Pro

- No performance impact
- More tamper resistant
- No management impact on platforms
- Works across OSs
- Can derive information that host based logs might not provide (port scanning, etc.) -> more global view of network



Network Level: Con

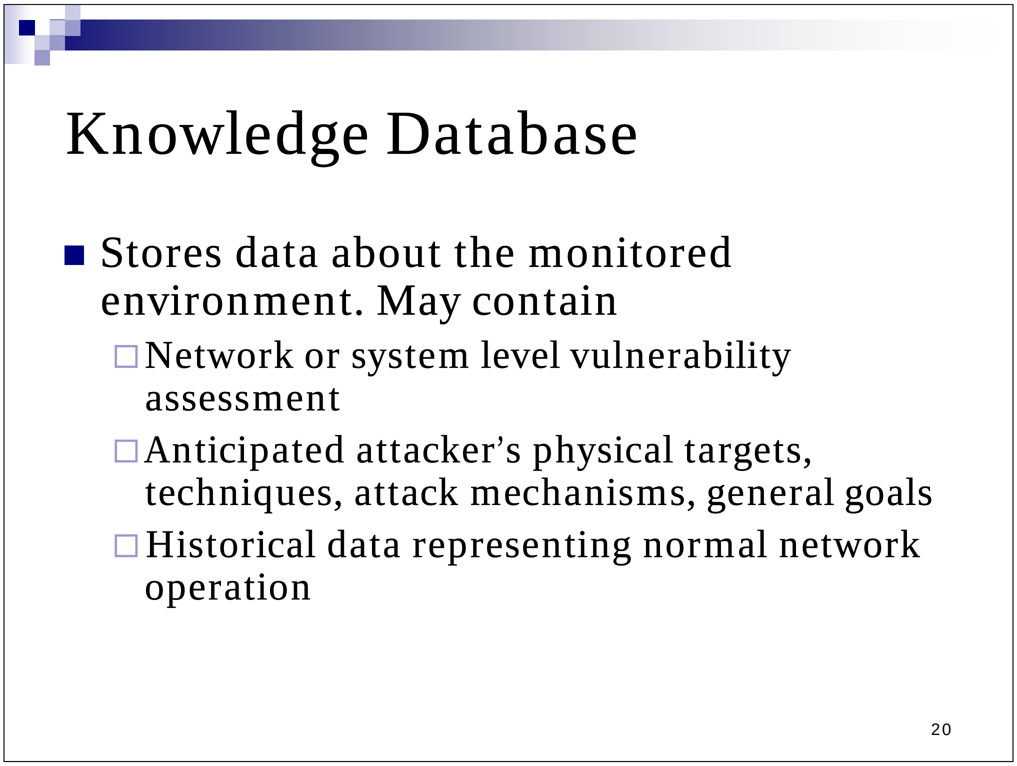
- May lose packets on flooded networks
- May mis-reassemble packets
- May not understand OS specific application protocols (e.g.: SMB)
- May not understand obsolete network protocols (e.g.: anything non-IP)
- Does not handle encrypted data
- Not all attacks arrive from the network



2. Knowledge Database

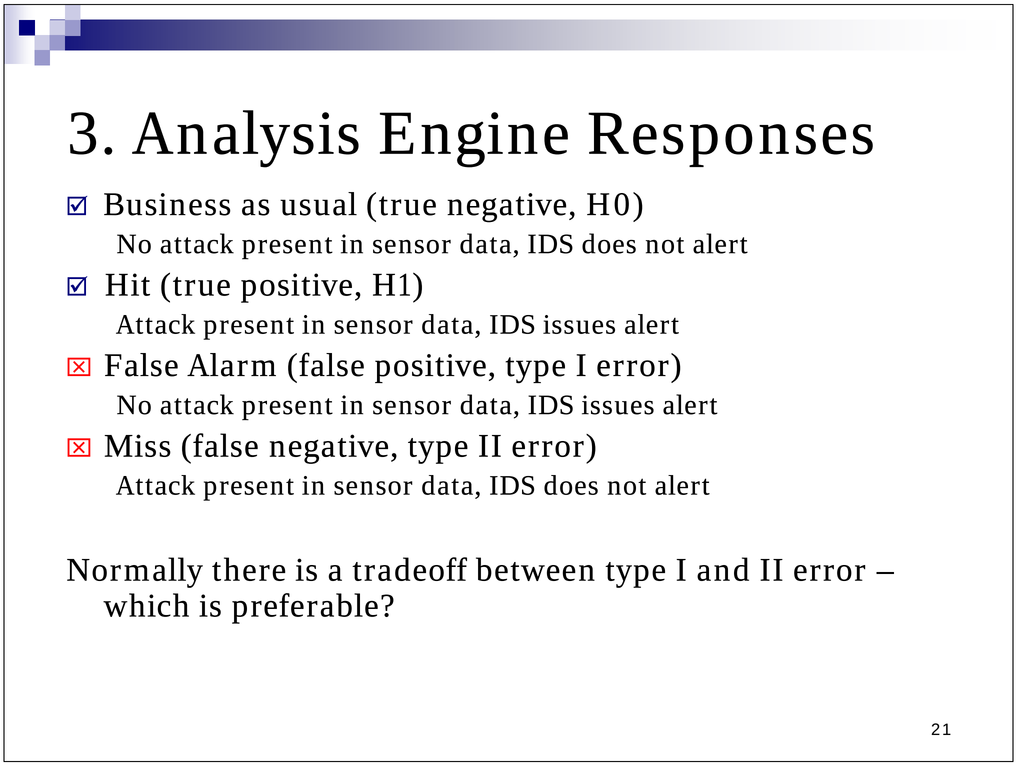
- Contains known attack or probing techniques catalogued by
  - □ Government Sponsored Research
    - DARPA & MIT Intrusion Detection Attacks Database http://www.ll.mit.edu/IST/ideval/data/data_index.html
  - □ Product and Service Vendors
  - □ Public Service Minded Hackers

System	Example	URL
IDS	IDS182	http://www.whitehats.com/IDS/182
CVE	CVE-2000-0138	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138
Bugtraq	ButraqID 1	http://www.securitfocus.com/vdb/bottom.html?vid=1
McAfee	McAfee 10225	http://vil.nai.com/vil/dispVirus.asp?virus_k=10225



Knowledge Database

- Stores data about the monitored environment. May contain
  - □ Network or system level vulnerability assessment
  - □ Anticipated attacker’s physical targets, techniques, attack mechanisms, general goals
  - □ Historical data representing normal network operation



3. Analysis Engine Responses

☑ Business as usual (true negative, H0)

  No attack present in sensor data, IDS does not alert

☑ Hit (true positive, H1)

  Attack present in sensor data, IDS issues alert

☒ False Alarm (false positive, type I error)

  No attack present in sensor data, IDS issues alert

☒ Miss (false negative, type II error)

  Attack present in sensor data, IDS does not alert

Normally there is a tradeoff between type I and II error – which is preferable?

Analysis Engine Approaches

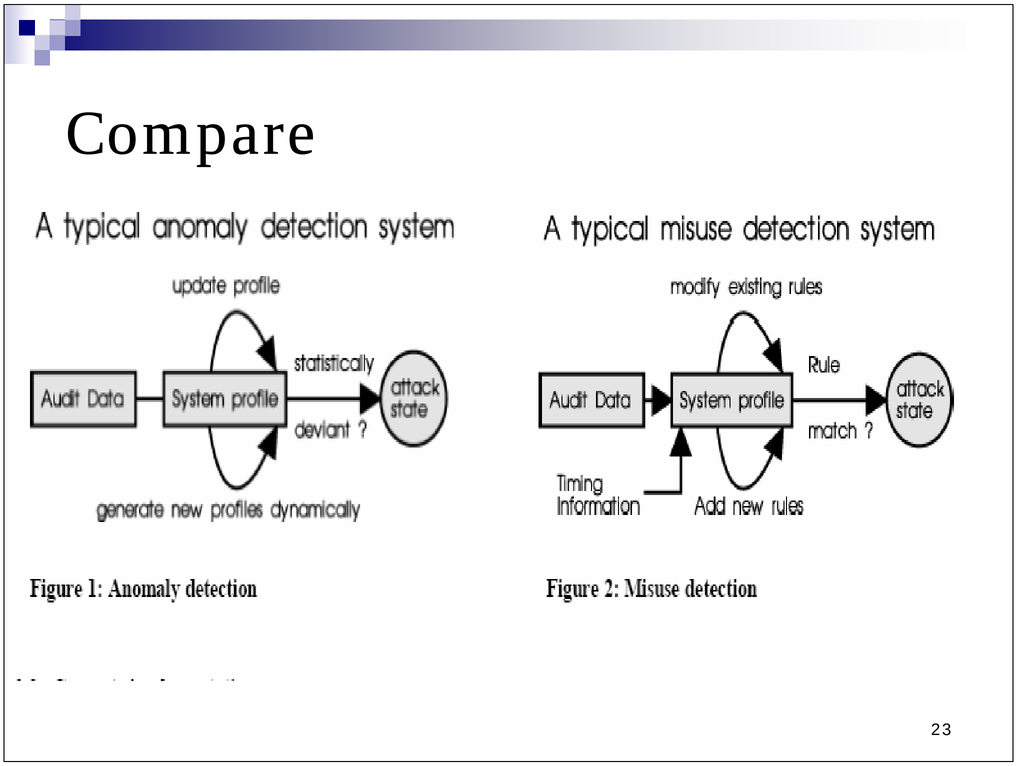
1. Rule based (misuse detection)
   - □ Accumulate knowledge about specific attacks and system vulnerabilities and use that knowledge to analyze events (e.g. SNORT’s rule set)

   Any event that is not explicitly recognized as an attack is considered acceptable

2. Behavior based (anomaly detection)
   - □ Assume that an intrusion can be detected by observing a deviation from normal or expected behavior of the system or the users

   Any event that does not correspond to a previously learned behavior is considered intrusive

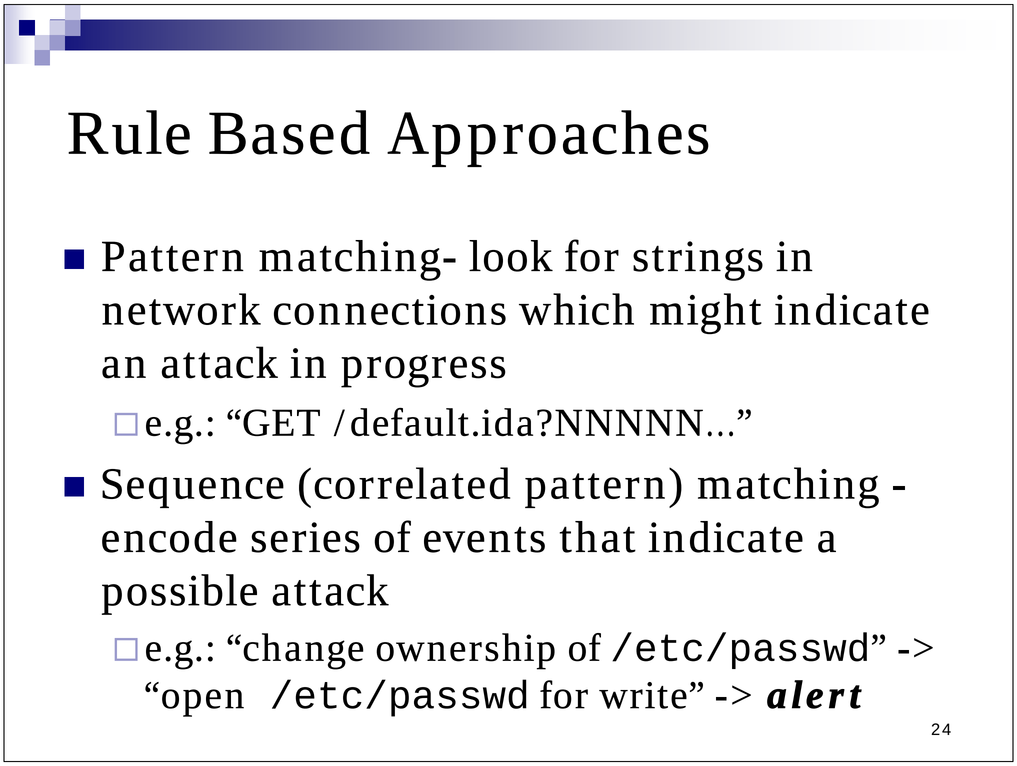
- Problems here?



Compare

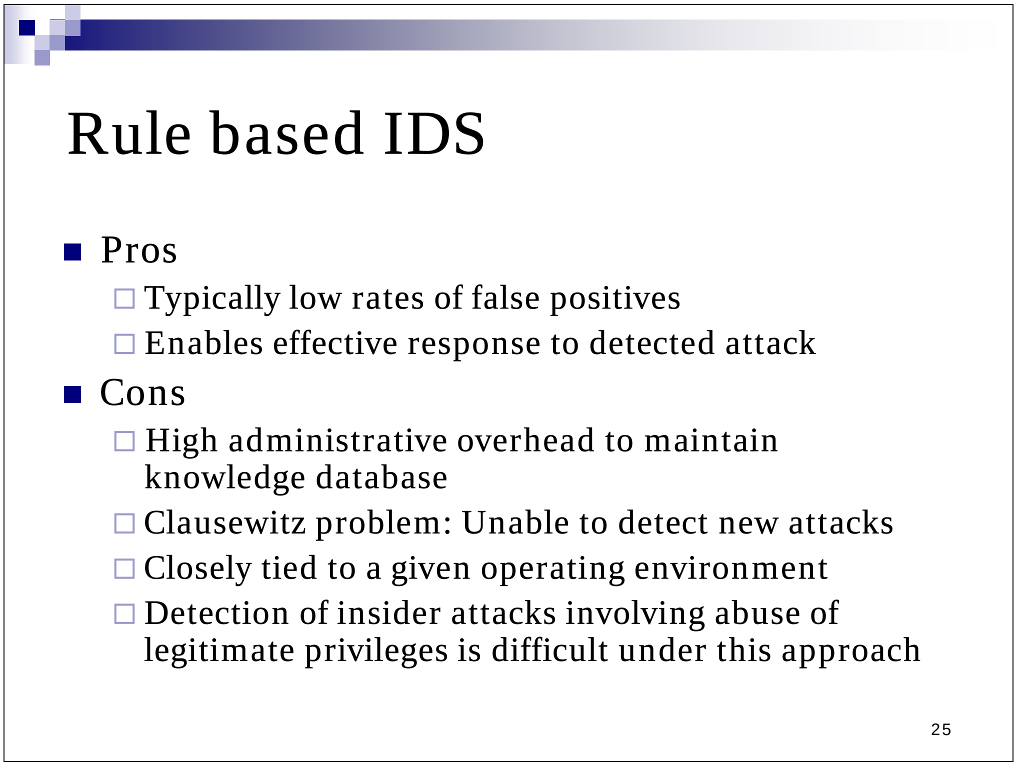
Figure 1: Anomaly detection

Figure 2: Misuse detection



Rule Based Approaches

- Pattern matching- look for strings in network connections which might indicate an attack in progress
  - □ e.g.: “GET /default.ida?NNNNN...”
- Sequence (correlated pattern) matching - encode series of events that indicate a possible attack
  - □ e.g.: “change ownership of `/etc/passwd`” -> “open `/etc/passwd` for write” -> ***alert***



Rule based IDS

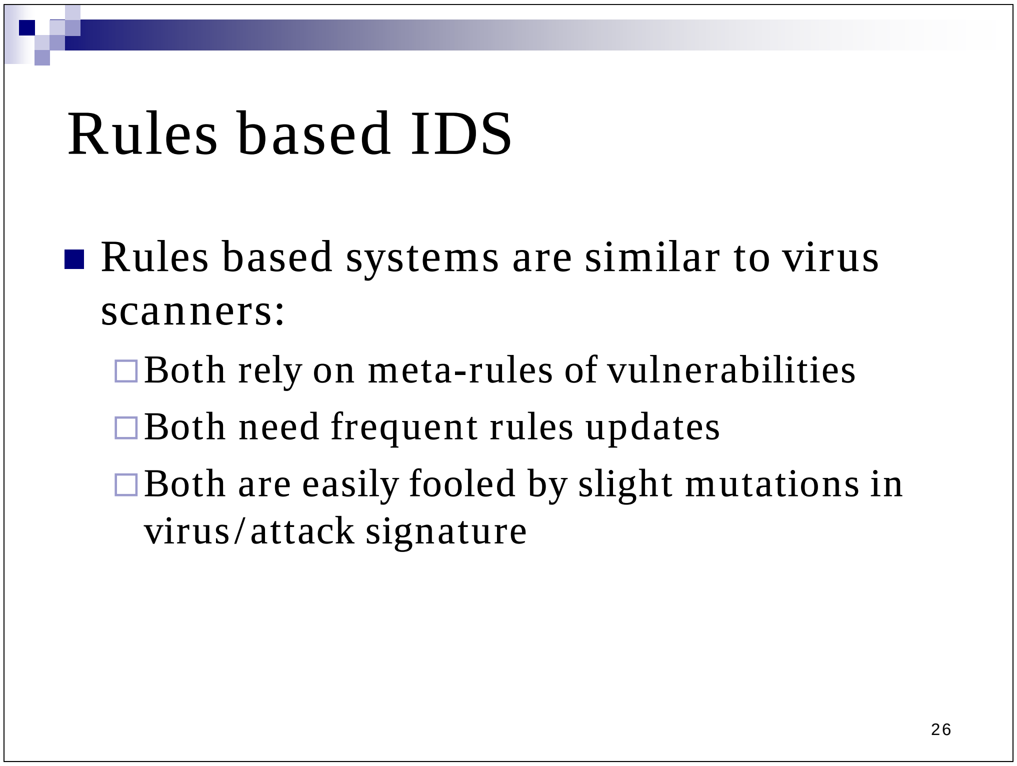
- Pros
  - Typically low rates of false positives
  - Enables effective response to detected attack
- Cons
  - High administrative overhead to maintain knowledge database
  - Clausewitz problem: Unable to detect new attacks
  - Closely tied to a given operating environment
  - Detection of insider attacks involving abuse of legitimate privileges is difficult under this approach

Carl Philipp Gottfried von Clausewitz (June 1, 1780 – November 16, 1831) was a Prussian soldier, military historian and influential military theorist. He is most famous for his military treatise Vom Kriege (translated into English as On War) On War is a long and intricate investigation based on his own experience in the Wars of the French Revolution.

Text from http://www.sans.org/resources/idfaq/knowledge_based.php

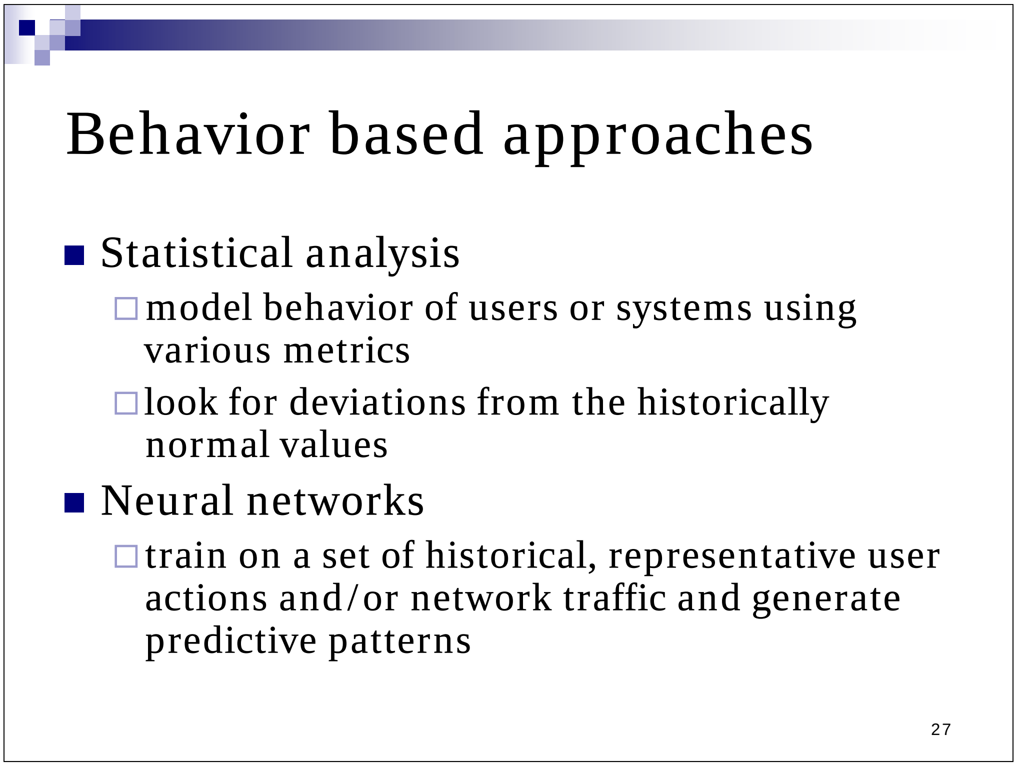
Advantages of the rule-based approaches are that they have the potential for very low false alarm rates, and the contextual analysis proposed by the intrusion detection system is detailed, making it easier for the security officer using this intrusion detection system to take preventive or corrective action.

Drawbacks include the difficulty of gathering the required information on the known attacks and keeping it up to date with new vulnerabilities and environments. Maintenance of the knowledge base of the intrusion detection system requires careful analysis of each vulnerability and is therefore a time-consuming task. rule-based approaches also have to face the generalization issue. Knowledge about attacks is very focused, dependent on the operating system, version, platform, and application. The resulting intrusion detection tool is therefore closely tied to a given environment. Also, detection of insider attacks involving an abuse of privileges is deemed more difficult because no vulnerability is actually exploited by the attacker.



Rules based IDS

- Rules based systems are similar to virus scanners:
  - □ Both rely on meta-rules of vulnerabilities
  - □ Both need frequent rules updates
  - □ Both are easily fooled by slight mutations in virus/attack signature

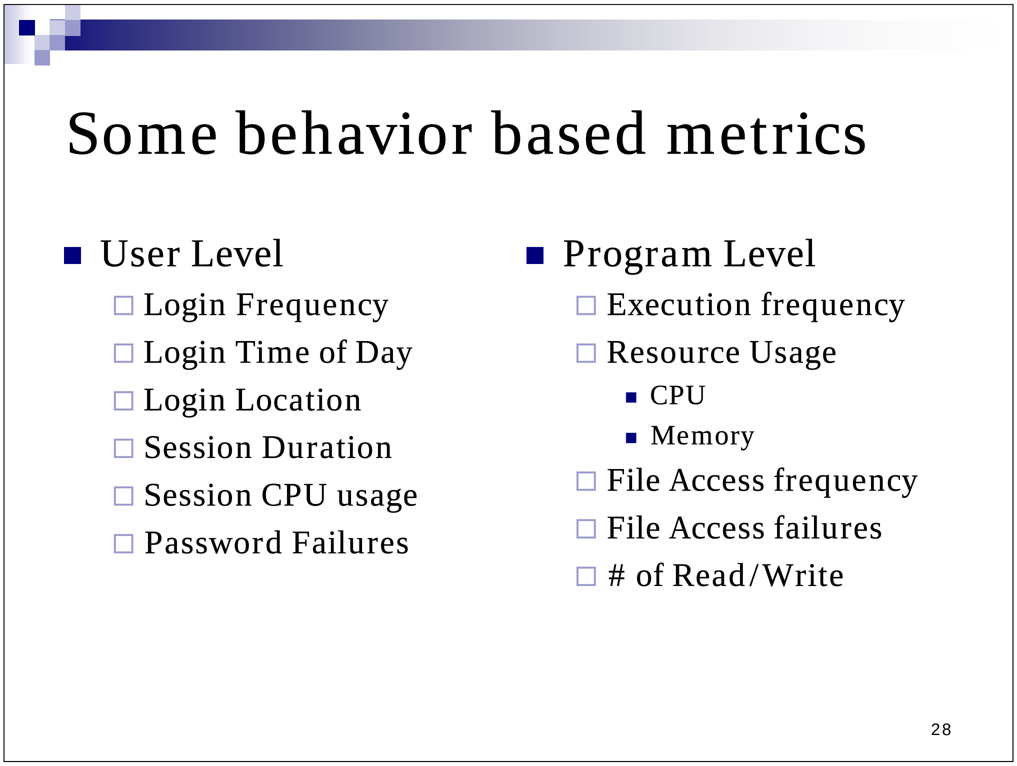


Behavior based approaches

- Statistical analysis
  - □ model behavior of users or systems using various metrics
  - □ look for deviations from the historically normal values
- Neural networks
  - □ train on a set of historical, representative user actions and/or network traffic and generate predictive patterns

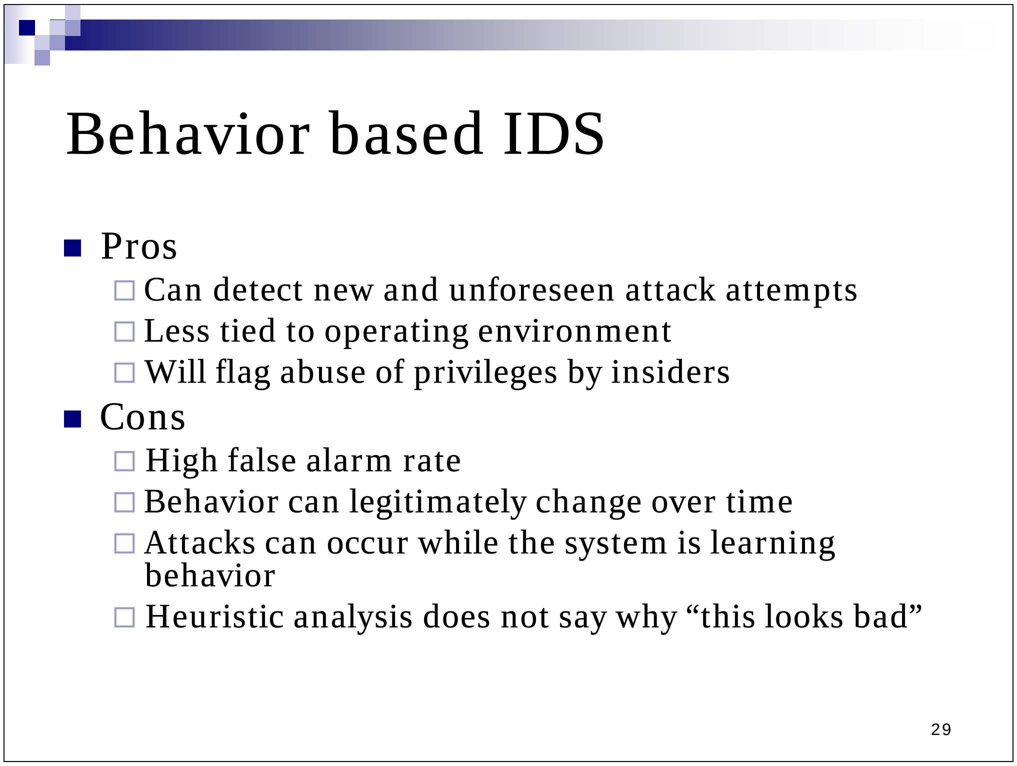
The common approach for anomaly detection concerns the statistical analysis, where the user or the system behavior is measured by a number of variables over the time. These variables may be the login and the logout time of each session, the amount of resources consumed during the session, and the resource duration. The major limitation of this approach is to find a correct threshold without frequent false-alarm detection.

Neural networks are algorithms that learn about the relationship between input-output vectors and "generalize" them to obtain new input-output vectors in a reasonable way. The main use of neural networks for intrusion detection is to learn the behavior of actors in the system (e.g., users, daemons). The advantage of using neural networks over statistics resides in having a simple way to express nonlinear relationships between variables, and in learning/retraining the neural network automatically. Neural networks are still a computationally intensive technique, and are not widely used in the intrusion detection community



Some behavior based metrics

- User Level
  - □ Login Frequency
  - □ Login Time of Day
  - □ Login Location
  - □ Session Duration
  - □ Session CPU usage
  - □ Password Failures
- Program Level
  - □ Execution frequency
  - □ Resource Usage
    - CPU
    - Memory
  - □ File Access frequency
  - □ File Access failures
  - □ # of Read/Write



Behavior based IDS

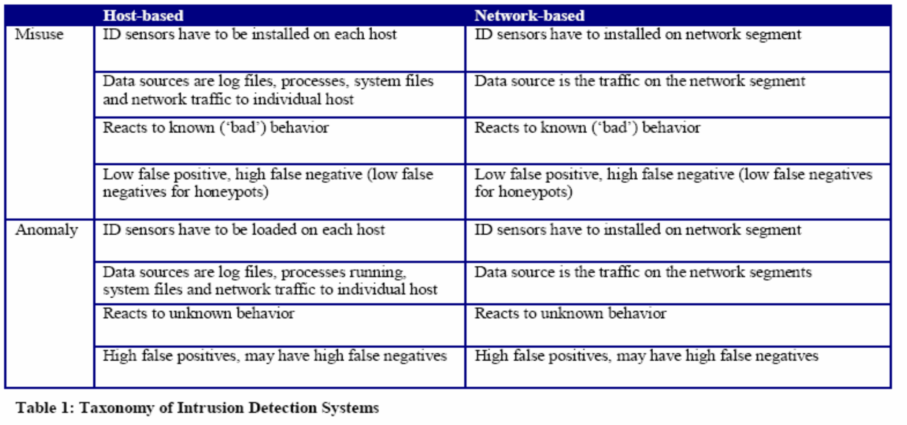
- Pros
  - Can detect new and unforeseen attack attempts
  - Less tied to operating environment
  - Will flag abuse of privileges by insiders
- Cons
  - High false alarm rate
  - Behavior can legitimately change over time
  - Attacks can occur while the system is learning behavior
  - Heuristic analysis does not say why “this looks bad”

Text from http://www.sans.org/resources/idfaq/behavior_based.php

Advantages of behavior-based approaches are that they can detect attempts to exploit new and unforeseen vulnerabilities. They can even contribute to the (partially) automatic discovery of these new attacks. They are less dependent on operating system-specific mechanisms. They also help detect 'abuse of privileges' types of attacks that do not actually involve exploiting any security vulnerability. In short, this is the paranoid approach: Everything which has not been seen previously is dangerous.

The high false alarm rate is generally cited as the main drawback of behavior-based techniques because the entire scope of the behavior of an information system may not be covered during the learning phase. Also, behavior can change over time, introducing the need for periodic online retraining of the behavior profile, resulting either in unavailability of the intrusion detection system or in additional false alarms. The information system can undergo attacks at the same time the intrusion detection system is learning the behavior. As a result, the behavior profile contains intrusive behavior, which is not detected as anomalous.

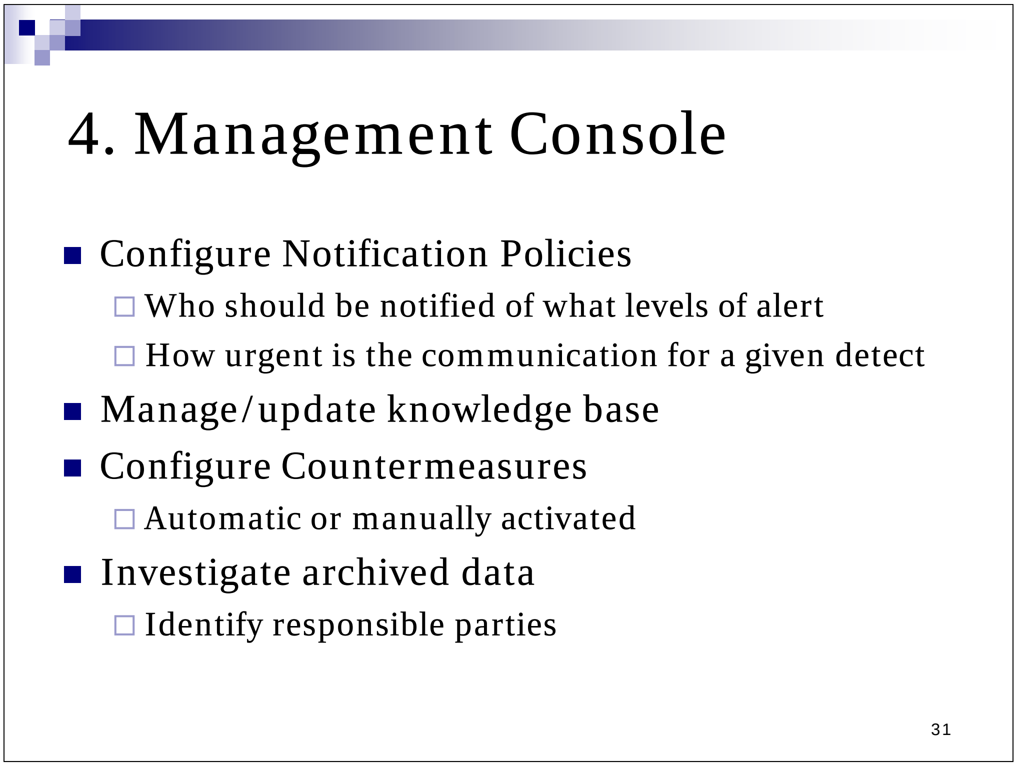
Recapitulation IDS taxonomy



	Host-based	Network-based
Misuse	ID sensors have to be installed on each host	ID sensors have to installed on network segment
	Data sources are log files, processes, system files and network traffic to individual host	Data source is the traffic on the network segment
	Reacts to known ('bad') behavior	Reacts to known ('bad') behavior
	Low false positive, high false negative (low false negatives for honeypots)	Low false positive, high false negative (low false negatives for honeypots)
Anomaly	ID sensors have to be loaded on each host	ID sensors have to installed on network segment
	Data sources are log files, processes running, system files and network traffic to individual host	Data source is the traffic on the network segments
	Reacts to unknown behavior	Reacts to unknown behavior
	High false positives, may have high false negatives	High false positives, may have high false negatives

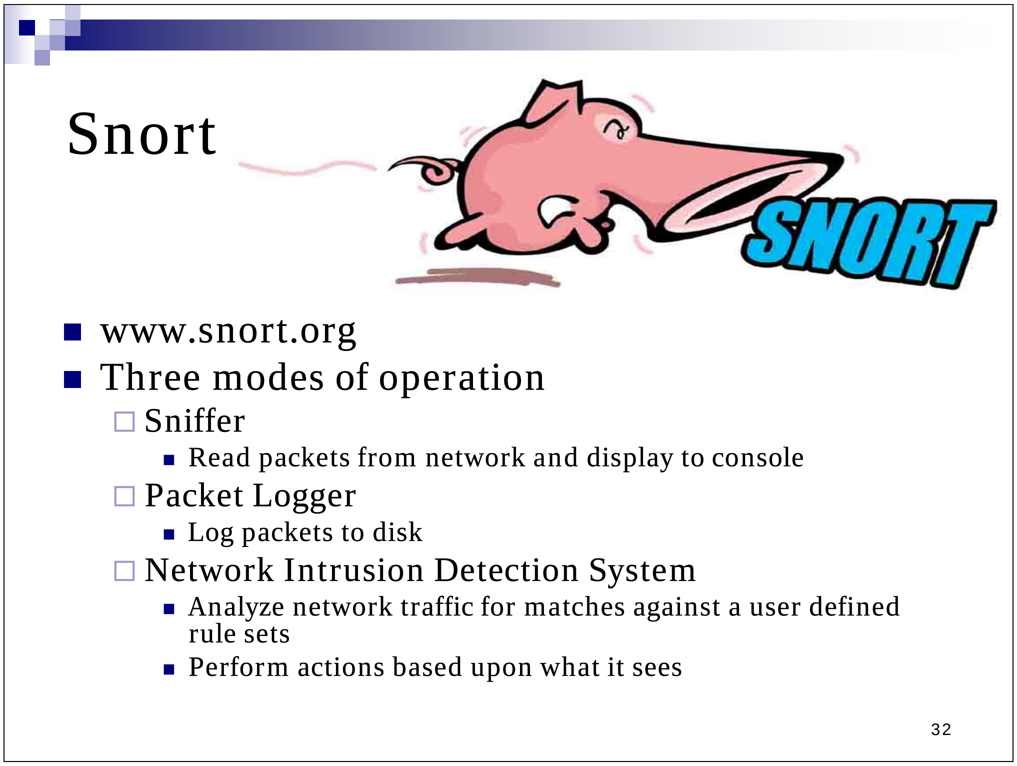
Table 1: Taxonomy of Intrusion Detection Systems

Read more at http://cs.wellesley.edu/~cs342/SPIE.pdf



4. Management Console

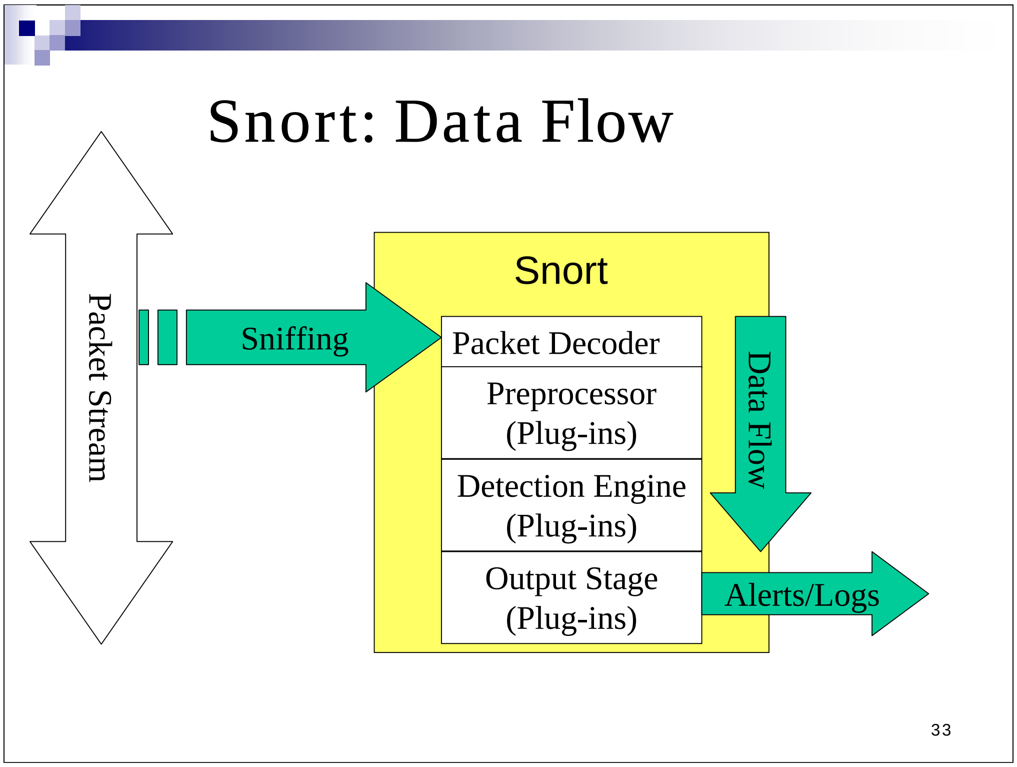
- Configure Notification Policies
  - □ Who should be notified of what levels of alert
  - □ How urgent is the communication for a given detect
- Manage/update knowledge base
- Configure Countermeasures
  - □ Automatic or manually activated
- Investigate archived data
  - □ Identify responsible parties



Snort

SNORT

- www.snort.org
- Three modes of operation
  - □ Sniffer
    - Read packets from network and display to console
  - □ Packet Logger
    - Log packets to disk
  - □ Network Intrusion Detection System
    - Analyze network traffic for matches against a user defined rule sets
    - Perform actions based upon what it sees



Snort: Data Flow

Packet Stream

Sniffing

Snort

Packet Decoder

Preprocessor
(Plug-ins)

Detection Engine
(Plug-ins)

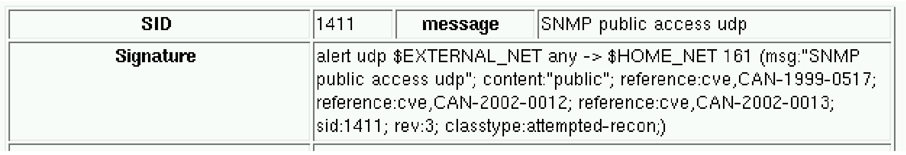
Output Stage
(Plug-ins)

Data Flow

Alerts/Logs

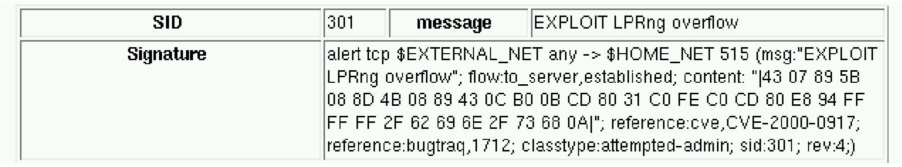
Snort Rules

- More than 2000 come with distribution



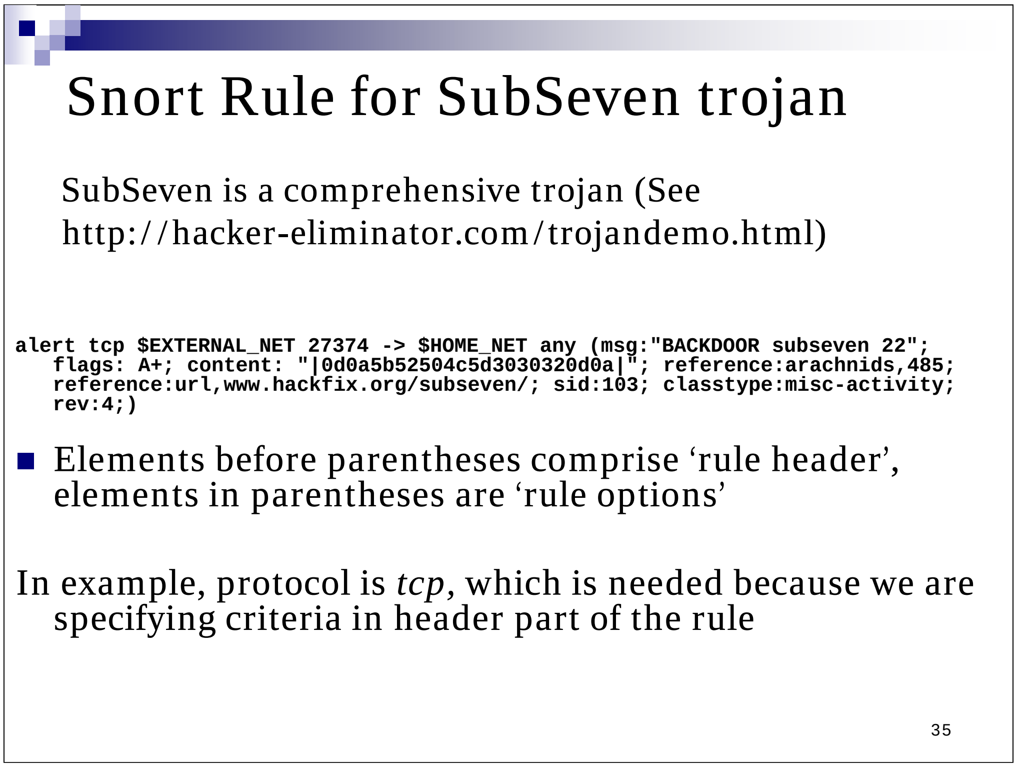
SID	1411	message	SNMP public access udp
Signature	alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP public access udp"; content:"public"; reference:cve,CAN-1999-0517; reference:cve,CAN-2002-0012; reference:cve,CAN-2002-0013; sid:1411; rev:3; classtype:attempted-recon;)		

  - □ Stateful Reassembly and Session Tracking



SID	301	message	EXPLOIT LPRng overflow
Signature	alert tcp $EXTERNAL_NET any -> $HOME_NET 515 (msg:"EXPLOIT LPRng overflow"; flow:to_server,established; content: "|43 07 89 5B 08 8D 4B 08 89 43 0C B0 0B CD 80 31 C0 FE C0 CD 80 E8 94 FF FF FF 2F 62 69 6E 2F 73 68 0A|"; reference:cve,CVE-2000-0917; reference:bugtraq,1712; classtype:attempted-admin; sid:301; rev:4;)		

- Can create and include custom rules, too



Snort Rule for SubSeven trojan

SubSeven is a comprehensive trojan (See http://hacker-eliminator.com/trojandemo.html)

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR subseven 22";
   flags: A+; content: "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
   reference:url,www.hackfix.org/subseven/; sid:103; classtype:misc-activity;
   rev:4;)
```

- Elements before parentheses comprise ‘rule header’, elements in parentheses are ‘rule options’

In example, protocol is *tcp*, which is needed because we are specifying criteria in header part of the rule

If the protocol is *IP*, Snort checks the link layer header to determine the packet type. If any other type of protocol is used, Snort uses the *IP* header to determine the protocol type.

The protocols only play a role in specifying criteria in the header part of the rule. The options part of the rule can have additional criteria unrelated to the specified protocol. For example, consider the following rule where the protocol is ICMP

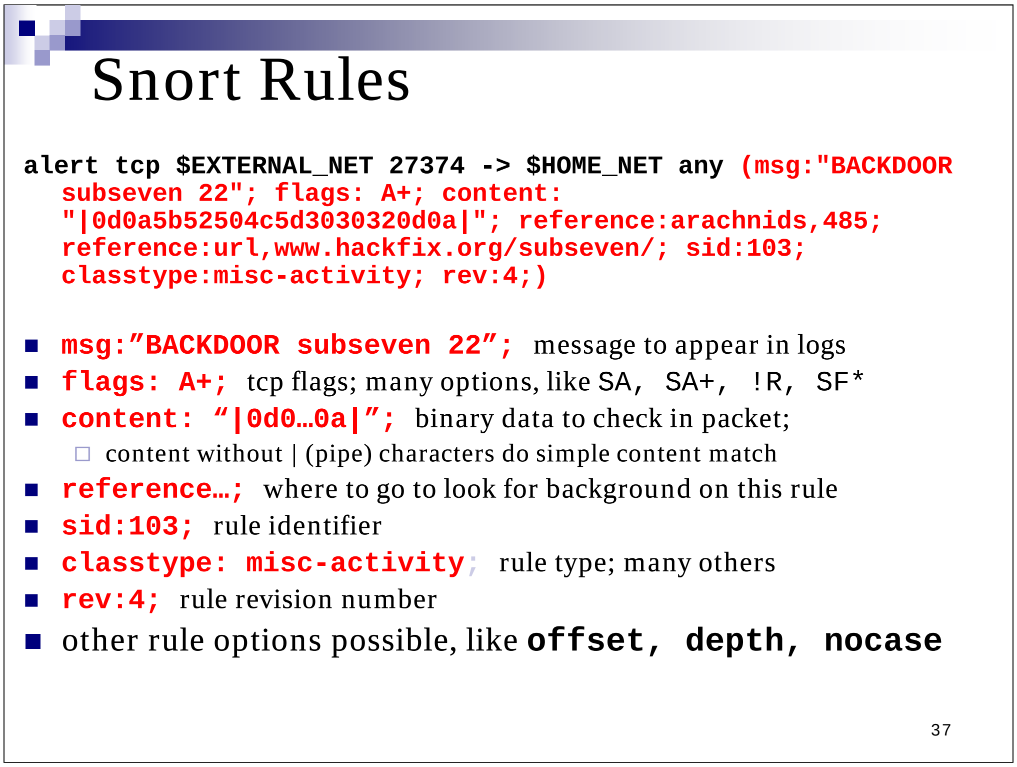
alert icmp any any -> any any (msg: "Ping with TTL=100"; ttl: 100;)

The options part checks the *TTL* (Time To Live) value, which is not part of the ICMP header. *TTL* is part of *IP* header instead. This means that the options part can check parameters in other protocol fields as well.

Snort Rules

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR
   subseven 22"; flags: A+; content:
   "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
   reference:url,www.hackfix.org/subseven/; sid:103;
   classtype:misc-activity; rev:4;)
```

- `alert` action to take; also **`log, pass, activate, dynamic`**
- `tcp` protocol; also **`udp, icmp, ip`**
- `$EXTERNAL_NET` source address; this is a variable – specific IP is ok
- `27374` source port; also **`any`**, negation (**`!21`**), range (**`1:1024`**)
- `->` direction; best not to change this, although **`<>`** is allowed
- `$HOME_NET` destination address; this is also a variable here
- `any` destination port



Snort Rules

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR
   subseven 22"; flags: A+; content:
   "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
   reference:url,www.hackfix.org/subseven/; sid:103;
   classtype:misc-activity; rev:4;)
```

- `msg:”BACKDOOR subseven 22”;` message to appear in logs
- `flags: A+;` tcp flags; many options, like `SA, SA+, !R, SF*`
- `content: “|0d0…0a|”;` binary data to check in packet;
  - content without | (pipe) characters do simple content match
- `reference…;` where to go to look for background on this rule
- `sid:103;` rule identifier
- `classtype: misc-activity;` rule type; many others
- `rev:4;` rule revision number
- other rule options possible, like **`offset, depth, nocase`**

Snort Rules

- Rules which actually caught intrusions

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433
  (msg:"MS-SQL xp_cmdshell - program execution";
  content:
  "x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|0
  0|l|00|"; nocase; flags:A+; classtype:attempted-
  user; sid:687; rev:3;)
```

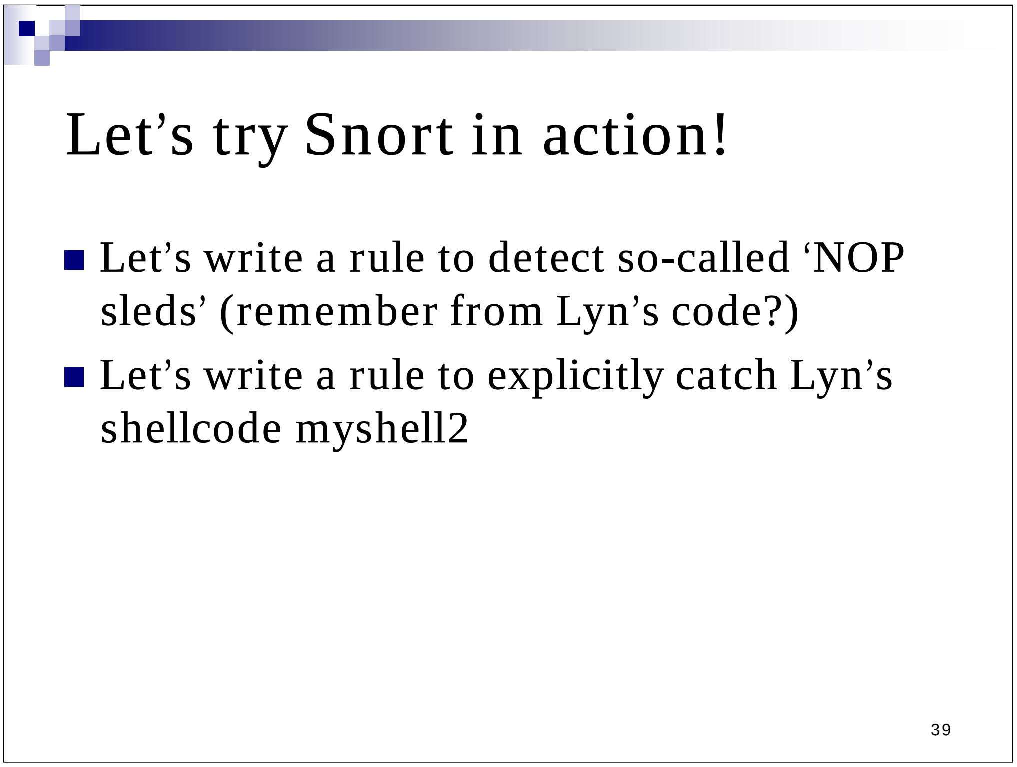
caught compromise of Microsoft SQL Server

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80
  (msg:"WEB-IIS cmd.exe access"; flags: A+;
  content:"cmd.exe"; nocase; classtype:web-application-
  attack; sid:1002; rev:2;)
```

caught Code Red infection

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"INFO
  FTP \"MKD / \" possible warez site"; flags: A+;
  content:"MKD / "; nocase; depth: 6; classtype:misc-
  activity; sid:554; rev:3;)
```

caught anonymous ftp server



Let’s try Snort in action!

- Let’s write a rule to detect so-called ‘NOP sleds’ (remember from Lyn’s code?)
- Let’s write a rule to explicitly catch Lyn’s shellcode myshell2

Network IDS Challenges

- Fundamental technological problem: Passive network monitor can not accurately predict whether a given system
  - .. will receive a given packet
  - .. will accept a given packet

- Because .. **IDS point of view is different than target system's**
  - □ Operating system (i.e., TCP/IP stack implementation)
  - □ Physical location on the network (time lag) affects data
  - □ Network saturation
- And of course ... IDS are inherently vulnerable to Denial of Service attacks
- Read about evasion techniques here : http://www.securityfocus.com/printable/infocus/1577

Insertion Attack

- Network sniffer accepts packet that target system discards (or never sees)

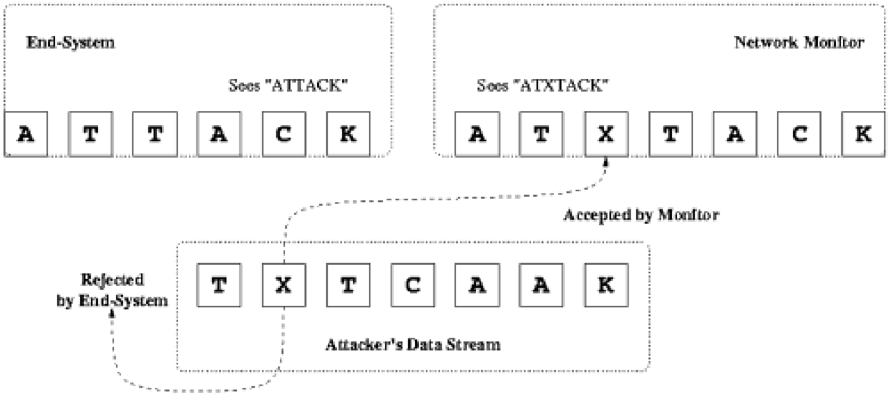


Image from http://www.insecure.org/stf/secnet_ids/secnet_ids.html

Evasion Attack

- Network sniffer misses packet that target system accepts (packet flies ‘under the radar’)

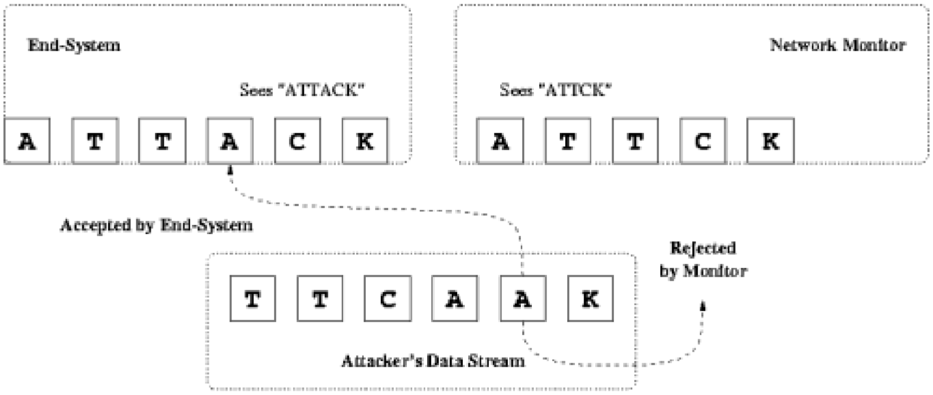


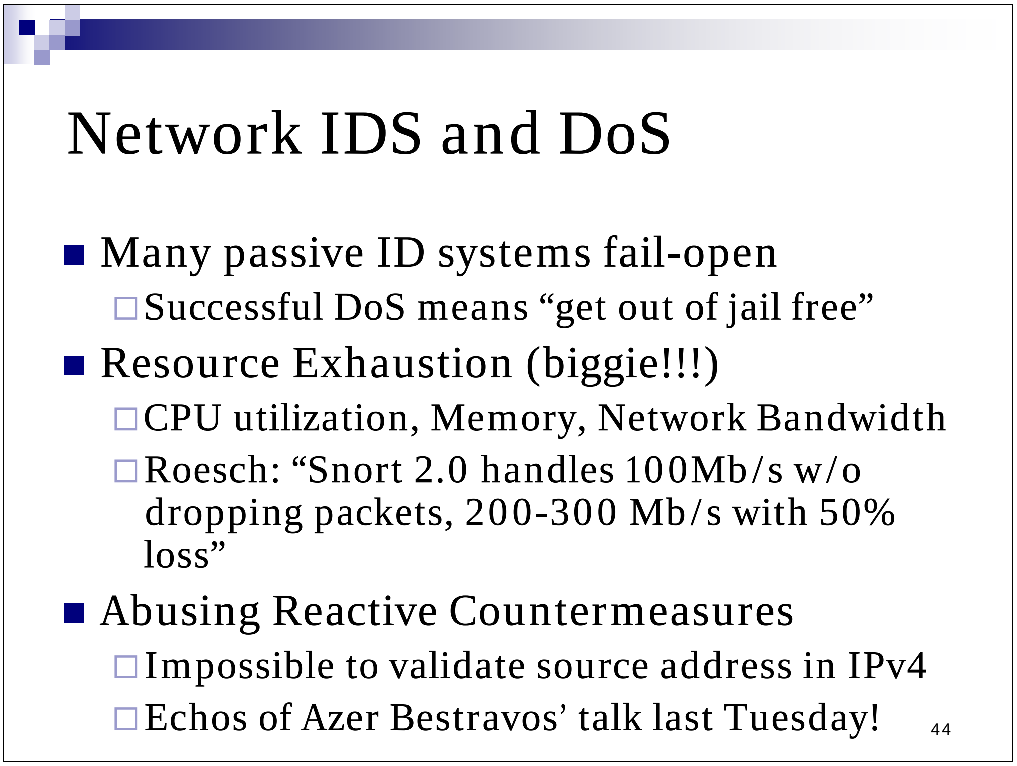
Image from http://www.insecure.org/stf/secnet_ids/secnet_ids.html

Why do these attacks work?

- ➔ Insufficient information on the wire to reconstruct what will happen later on
- ➔ The IDS does not necessarily have the same ‘view’ of the data stream as the end systems!

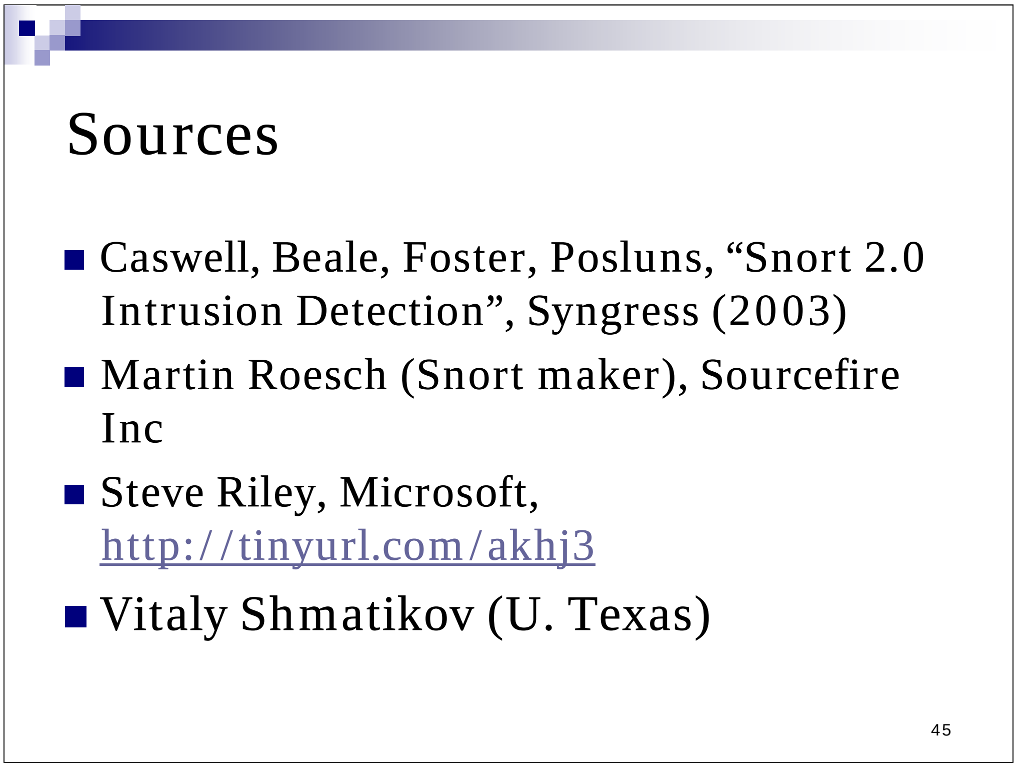
How can views differ?

- ■ Packet TTL not large enough to reach target but will reach IDS
- ■ Target (and not IDS) drops packets with certain TCP options set (e.g., source routed packets)
- ■ IDS may not verify packet checksums
- ■ Reassembly of overlapping or conflicting fragments handled differently by target and IDS
- ■ Out of sequence RST packets may be mishandled (RFC says ignore)
- ■ End systems TCP/IP stack different from IDS (this is how **nmap –O** works)



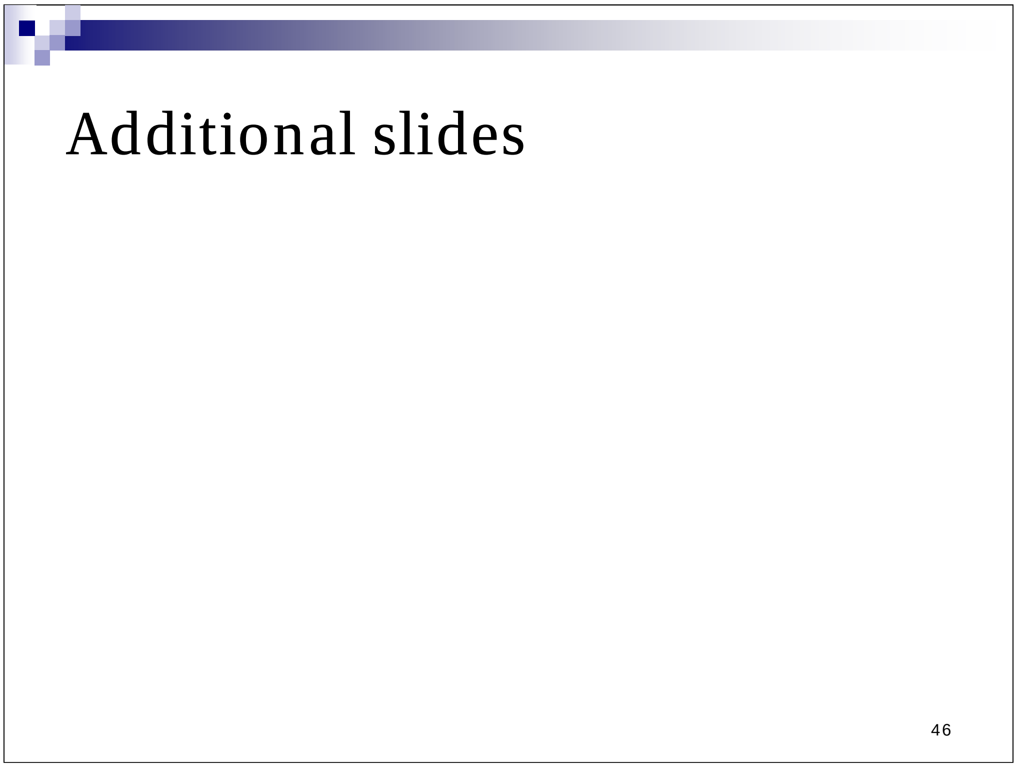
Network IDS and DoS

- Many passive ID systems fail-open
  - □ Successful DoS means “get out of jail free”
- Resource Exhaustion (biggie!!!)
  - □ CPU utilization, Memory, Network Bandwidth
  - □ Roesch: “Snort 2.0 handles 100Mb/s w/o dropping packets, 200-300 Mb/s with 50% loss”
- Abusing Reactive Countermeasures
  - □ Impossible to validate source address in IPv4
  - □ Echos of Azer Bestravos’ talk last Tuesday!



Sources

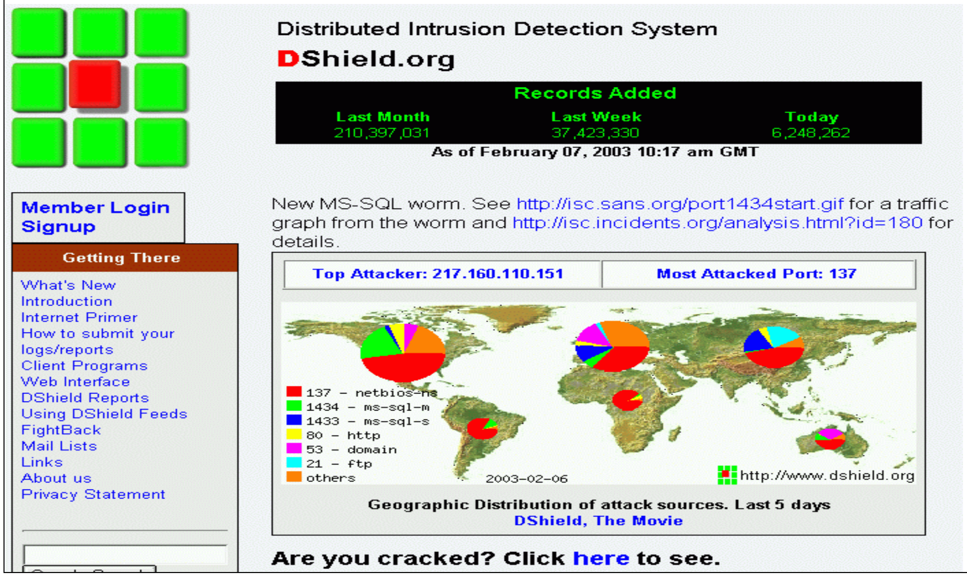
- Caswell, Beale, Foster, Posluns, “Snort 2.0 Intrusion Detection”, Syngress (2003)
- Martin Roesch (Snort maker), Sourcefire Inc
- Steve Riley, Microsoft, http://tinyurl.com/akhj3
- Vitaly Shmatikov (U. Texas)



Additional slides

DShield

(http://www.dshield.org)



Distributed Intrusion Detection System

DShield.org

Records Added

Last Month	Last Week	Today
210,397,031	37,423,330	6,248,262

As of February 07, 2003 10:17 am GMT

Member Login
Signup

Getting There

- What's New
- Introduction
- Internet Primer
- How to submit your logs/reports
- Client Programs
- Web Interface
- DShield Reports
- Using DShield Feeds
- FightBack
- Mail Lists
- Links
- About us
- Privacy Statement

New MS-SQL worm. See http://isc.sans.org/port1434start.gif for a traffic graph from the worm and http://isc.incidents.org/analysis.html?id=180 for details.

Top Attacker: 217.160.110.151	Most Attacked Port: 137

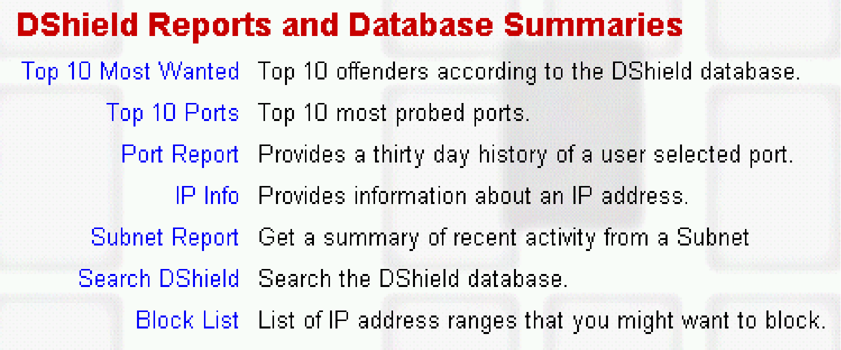
🟥 137 – netbios-ns
🟩 1434 – ms-sql-m
🟦 1433 – ms-sql-s
🟨 80 – http
🟪 53 – domain
🟦 21 – ftp
🟧 others

Geographic Distribution of attack sources. Last 5 days
DShield, The Movie

Are you cracked? Click here to see.

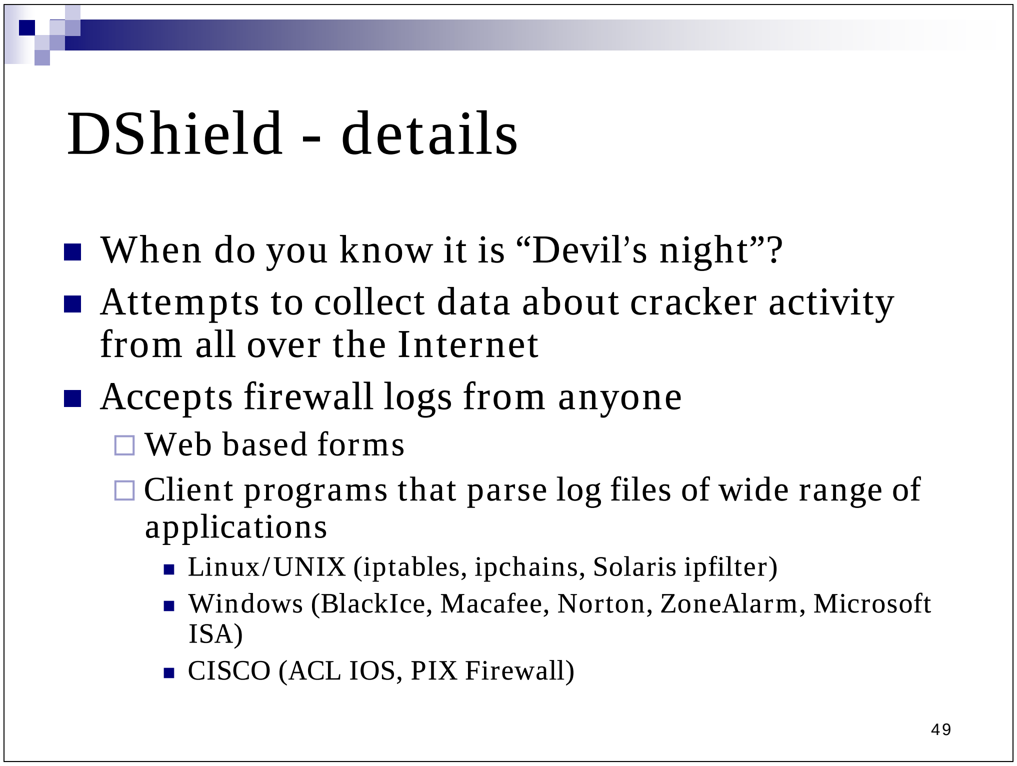
DShield – Details (cont)

- Archives and Correlates data
- Generates Reports



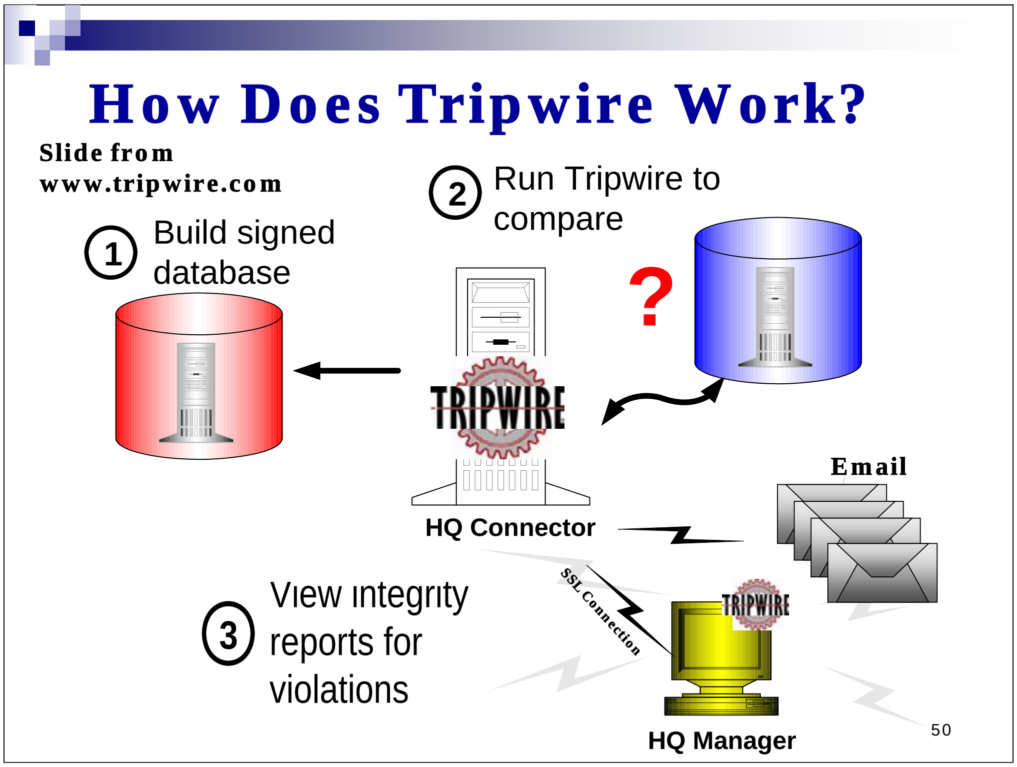
DShield Reports and Database Summaries

Top 10 Most Wanted	Top 10 offenders according to the DShield database.
Top 10 Ports	Top 10 most probed ports.
Port Report	Provides a thirty day history of a user selected port.
IP Info	Provides information about an IP address.
Subnet Report	Get a summary of recent activity from a Subnet
Search DShield	Search the DShield database.
Block List	List of IP address ranges that you might want to block.



DShield - details

- When do you know it is “Devil’s night”?
- Attempts to collect data about cracker activity from all over the Internet
- Accepts firewall logs from anyone
  - □ Web based forms
  - □ Client programs that parse log files of wide range of applications
    - Linux/UNIX (iptables, ipchains, Solaris ipfilter)
    - Windows (BlackIce, Macafee, Norton, ZoneAlarm, Microsoft ISA)
    - CISCO (ACL IOS, PIX Firewall)



How Does Tripwire Work?

Slide from www.tripwire.com

1 Build signed database

2 Run Tripwire to compare

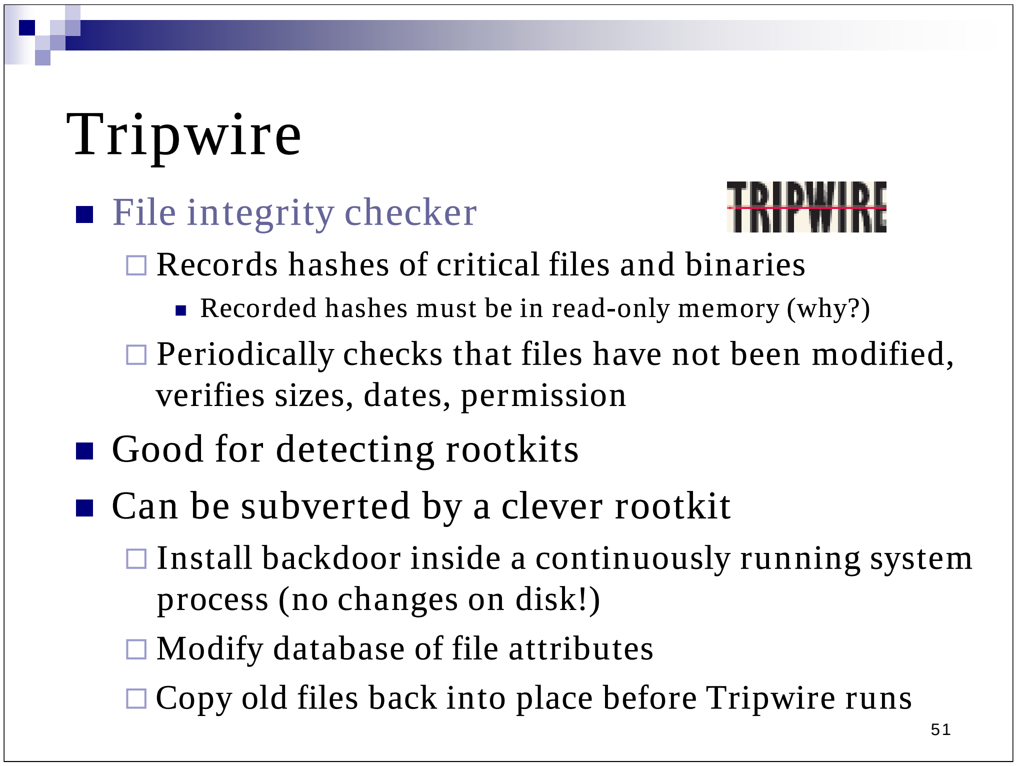
3 View integrity reports for violations

How does Tripwire work? It answers that fundamental question, “Is my system the same since I last checked it?”

Tripwire works by first creating a snapshot or database of a system existing file system in a known good state. The this snapshot is digitally signed using the El Gamal, cryptographic signature, so that no changes can be made without you knowing it.

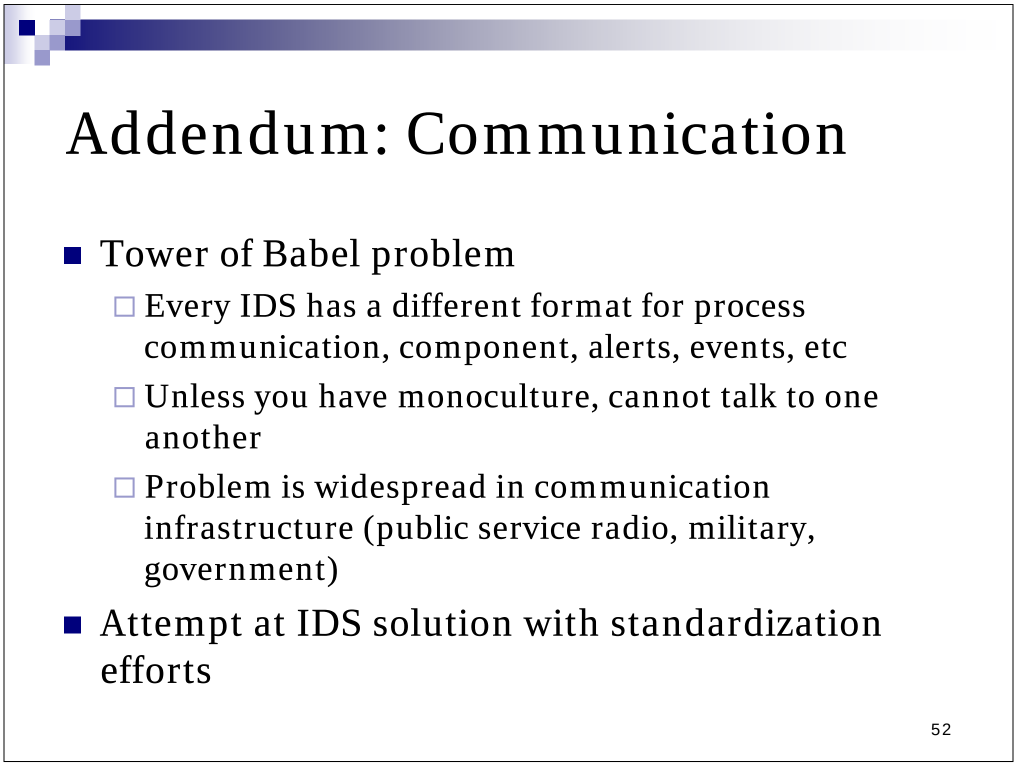
Then at a later time another snapshot is created and compared to the baseline to see if there are differences. If difference exist between the two snapshots then a report is generated and can be emailed, sent to syslog or sent to the HQ Manager for viewing.

Again Tripwire is basic in its operation, but fundamental to a core security strategy.



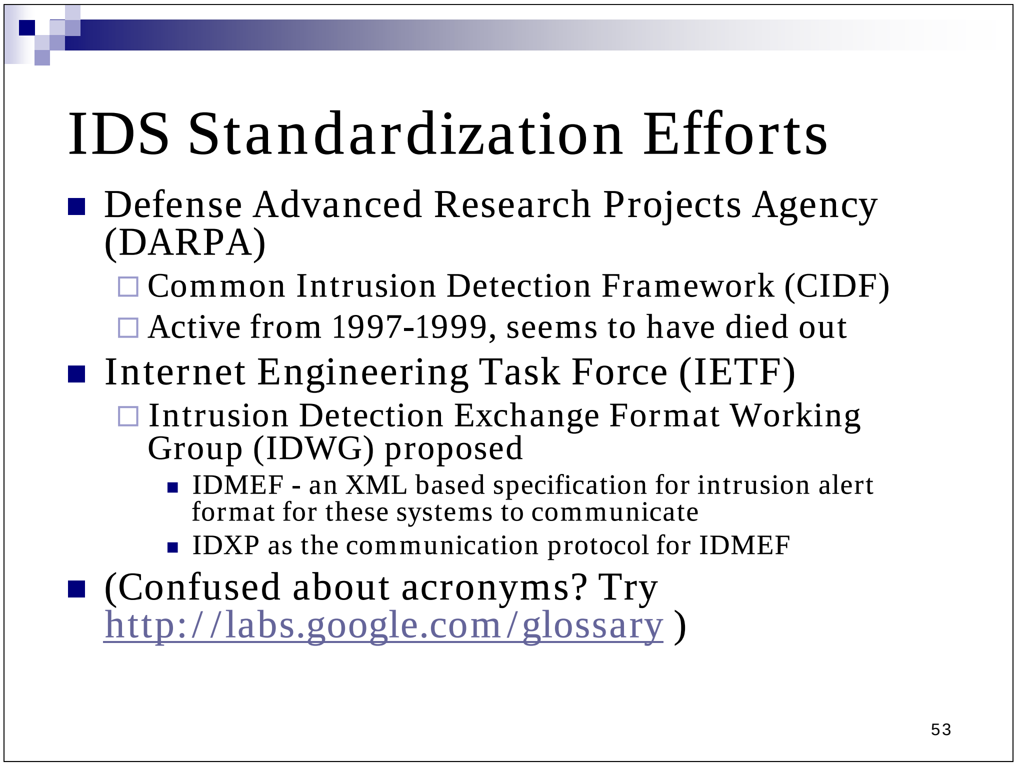
Tripwire

- File integrity checker
  - Records hashes of critical files and binaries
    - Recorded hashes must be in read-only memory (why?)
  - Periodically checks that files have not been modified, verifies sizes, dates, permission
- Good for detecting rootkits
- Can be subverted by a clever rootkit
  - Install backdoor inside a continuously running system process (no changes on disk!)
  - Modify database of file attributes
  - Copy old files back into place before Tripwire runs



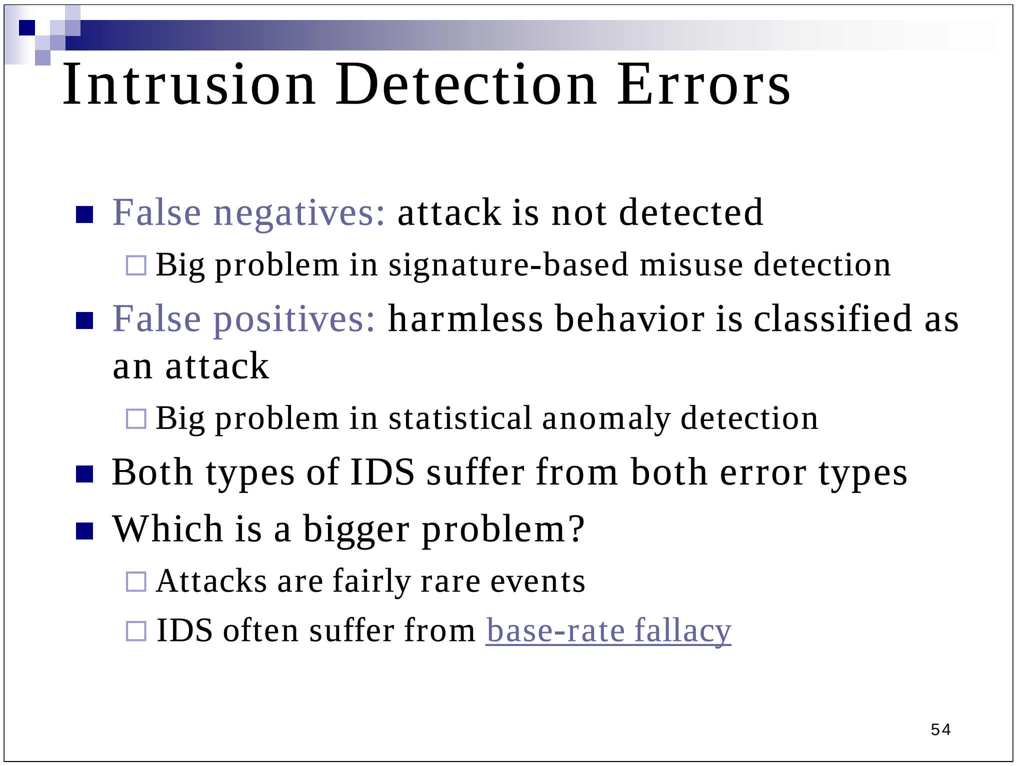
Addendum: Communication

- Tower of Babel problem
  - □ Every IDS has a different format for process communication, component, alerts, events, etc
  - □ Unless you have monoculture, cannot talk to one another
  - □ Problem is widespread in communication infrastructure (public service radio, military, government)
- Attempt at IDS solution with standardization efforts



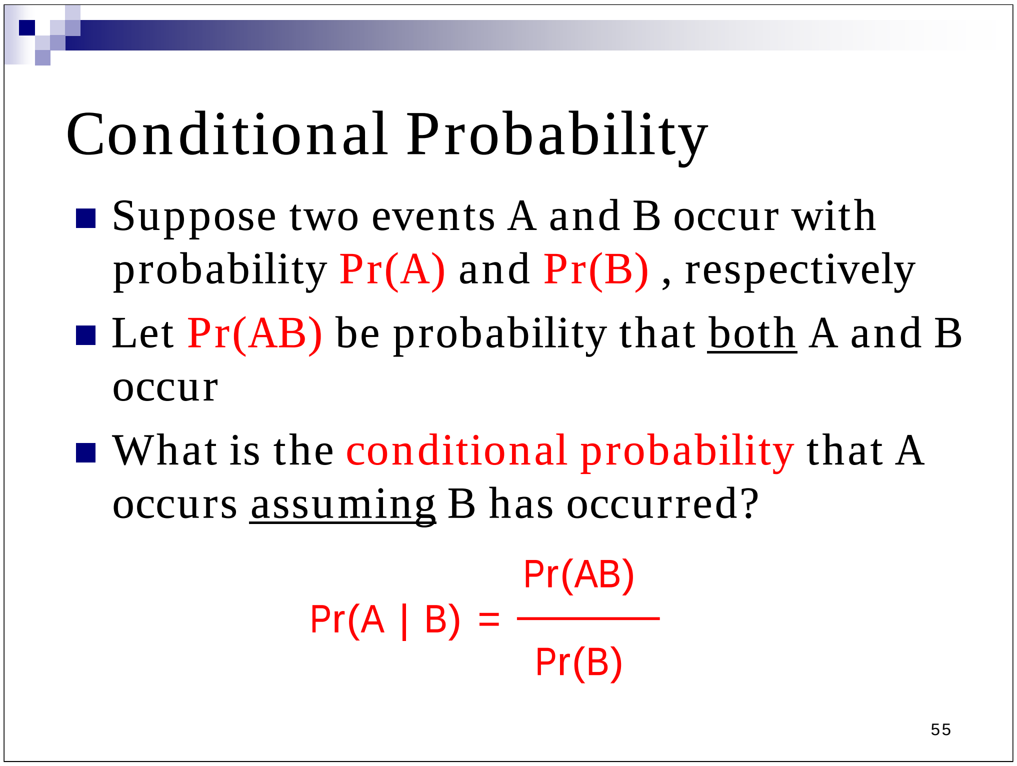
IDS Standardization Efforts

- Defense Advanced Research Projects Agency (DARPA)
  - □ Common Intrusion Detection Framework (CIDF)
  - □ Active from 1997-1999, seems to have died out
- Internet Engineering Task Force (IETF)
  - □ Intrusion Detection Exchange Format Working Group (IDWG) proposed
    - IDMEF - an XML based specification for intrusion alert format for these systems to communicate
    - IDXP as the communication protocol for IDMEF
- (Confused about acronyms? Try http://labs.google.com/glossary)



Intrusion Detection Errors

- False negatives: attack is not detected
  - Big problem in signature-based misuse detection
- False positives: harmless behavior is classified as an attack
  - Big problem in statistical anomaly detection
- Both types of IDS suffer from both error types
- Which is a bigger problem?
  - Attacks are fairly rare events
  - IDS often suffer from base-rate fallacy



Conditional Probability

- Suppose two events A and B occur with probability Pr(A) and Pr(B) , respectively
- Let Pr(AB) be probability that both A and B occur
- What is the conditional probability that A occurs assuming B has occurred?

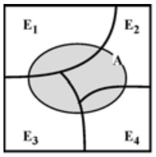
$$\Pr(A \mid B) = \frac{\Pr(AB)}{\Pr(B)}$$

Bayes' Theorem

- Suppose mutually exclusive events $E_1, \ldots, E_n$ together cover the entire set of possibilities
- Then probability of <u>any</u> event A occurring is

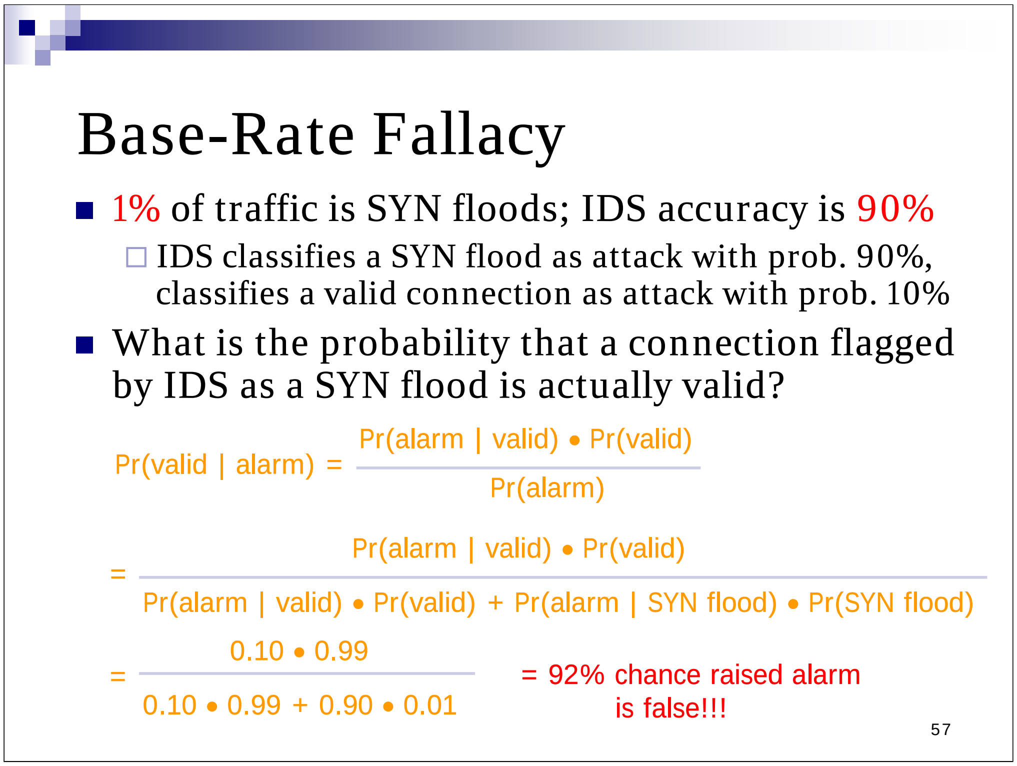
  $$\Pr(A) = \Sigma_{1 \le i \le n} \Pr(A \mid E_i) \bullet \Pr(E_i)$$

  - Intuition: since $E_1, \ldots, E_n$ cover entire probability space, whenever A occurs, some event E_i must have occurred



- Can rewrite this formula as

$$\Pr(E_i \mid A) = \frac{\Pr(A \mid E_i) \bullet \Pr(E_i)}{\Pr(A)}$$



Base-Rate Fallacy

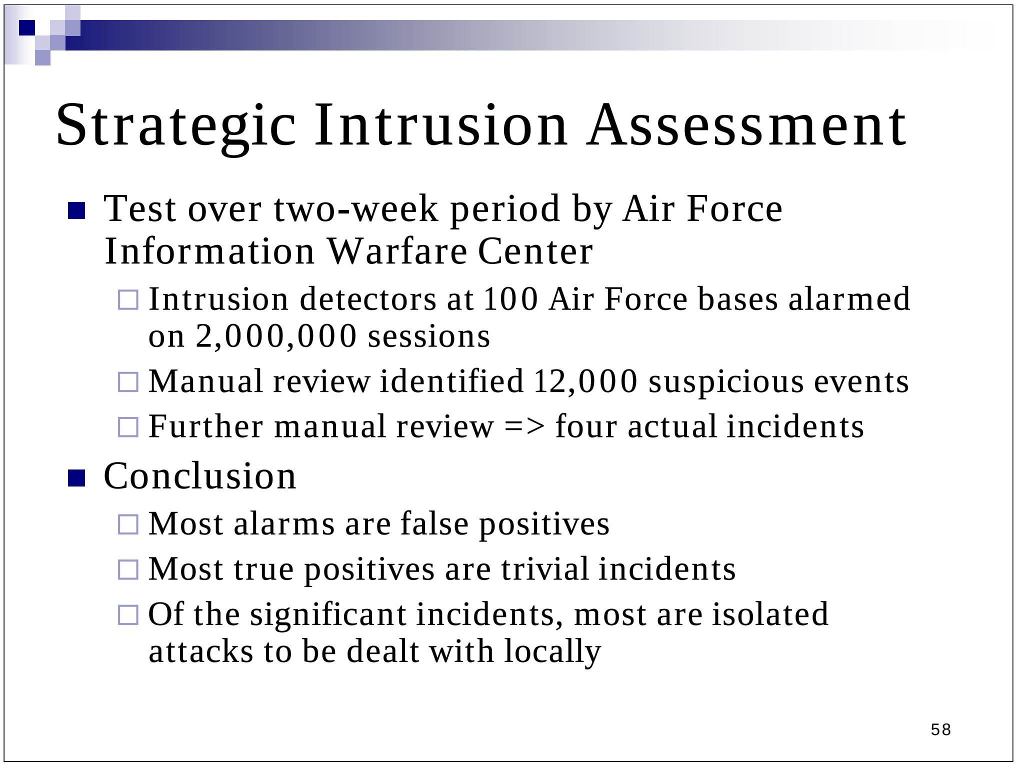
- 1% of traffic is SYN floods; IDS accuracy is 90%
  - IDS classifies a SYN flood as attack with prob. 90%, classifies a valid connection as attack with prob. 10%
- What is the probability that a connection flagged by IDS as a SYN flood is actually valid?

$$\Pr(\text{valid} \mid \text{alarm}) = \frac{\Pr(\text{alarm} \mid \text{valid}) \bullet \Pr(\text{valid})}{\Pr(\text{alarm})}$$

$$= \frac{\Pr(\text{alarm} \mid \text{valid}) \bullet \Pr(\text{valid})}{\Pr(\text{alarm} \mid \text{valid}) \bullet \Pr(\text{valid}) + \Pr(\text{alarm} \mid \text{SYN flood}) \bullet \Pr(\text{SYN flood})}$$

$$= \frac{0.10 \bullet 0.99}{0.10 \bullet 0.99 + 0.90 \bullet 0.01}$$

= 92% chance raised alarm is false!!!



Strategic Intrusion Assessment

- Test over two-week period by Air Force Information Warfare Center
  - □ Intrusion detectors at 100 Air Force bases alarmed on 2,000,000 sessions
  - □ Manual review identified 12,000 suspicious events
  - □ Further manual review => four actual incidents
- Conclusion
  - □ Most alarms are false positives
  - □ Most true positives are trivial incidents
  - □ Of the significant incidents, most are isolated attacks to be dealt with locally