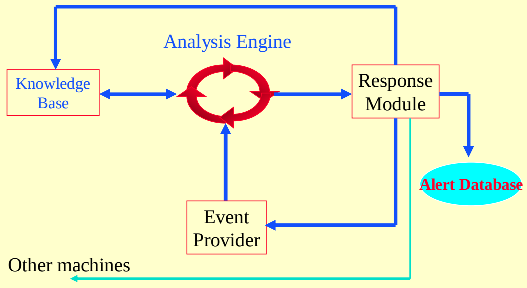
CPE5021
Advanced Network Security

--- IDS: Theory and Practice---

Lecture 6

Intrusion Detection System (IDS)



Intrusion Detection System (IDS)

- Intrusion detection is the process of identifying and responding to malicious activity targeted at resources
- IDS is a system designed to test/analyse network system traffic/events against a given set of parameters and alert/capture data when these thresholds are met.
- IDS uses collected information and predefined knowledge-based system to reason about the possibility of an intrusion.
- IDS also provides services to cop with intrusion such as giving alarms, activating programs to try to deal with intrusion, etc.

Intrusion Detection Theory

- In designing an IDS, we need to know what we want to detect. There are two man types of intrusion detection:
  - Anomaly detection: Detect anomalously intrusive activities.
    - Detect intrusive activities what vary from established profile by statistically significant amount.
    - Create a model constituting normal activities for the observed network, and then decide on what percentage an activity must be flagged as abnormal.
  - Misuse model: Detect misuses.
    - Using known attack patterns or signatures to detect similar attacks.

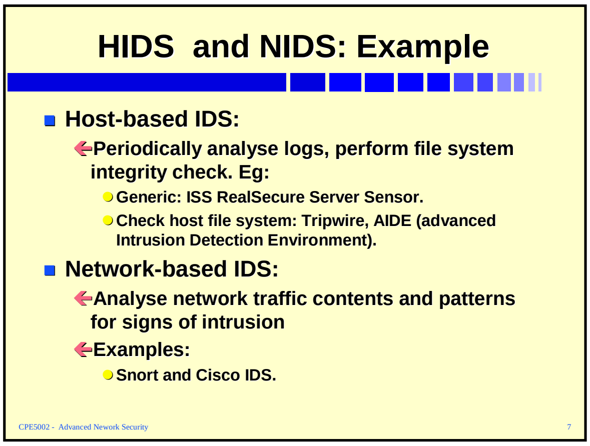
Example of IDS with attack signatures

- Most IDSs include a set of signatures and match attacks against these signatures.
- For each attack a set of signatures must be defined.
  - The IDS system will capture data and use this data to match against the set of know signatures.
  - A positive match results in an alert being generated.
- When creating attack signatures we need to minimise false positives, or false negatives.
  - A false positive is when something looks like an attack but it is not.
  - A false negative is when something does not look like an attack but it is.

IDS in Practice

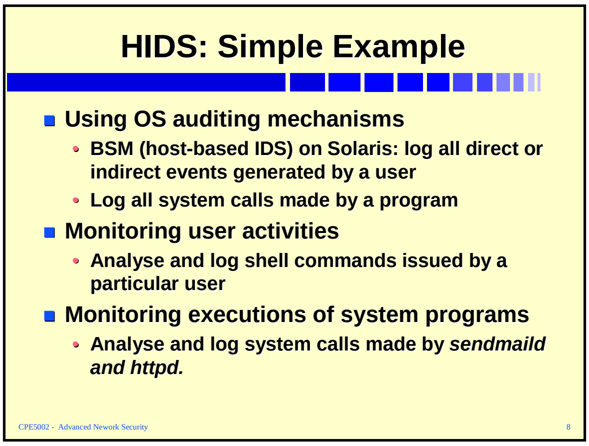
There are two main IDS types:

- A Network-based IDS system **examines the individual packets flowing through a network and should be able to understand all the different flags and options that can exist within a network packet.**
  - **It can then detect malicious packets (that may be overlooked by firewalls' rules).**
  - **It can also look at packet payload, (try to understand what program is being accessed and with what options).**
- A Host based IDS system – **examines activity on individual computers (hosts). It can detect repeatedly failed access attempts or changes to the local's critical system files.**



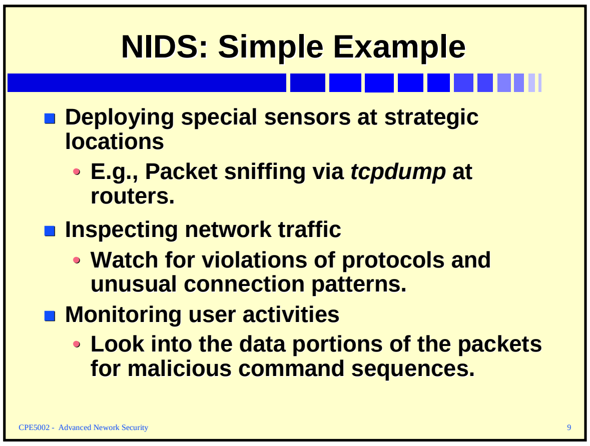
HIDS and NIDS: Example

- Host-based IDS:
  - Periodically analyse logs, perform file system integrity check. Eg:
    - Generic: ISS RealSecure Server Sensor.
    - Check host file system: Tripwire, AIDE (advanced Intrusion Detection Environment).
- Network-based IDS:
  - Analyse network traffic contents and patterns for signs of intrusion
  - Examples:
    - Snort and Cisco IDS.



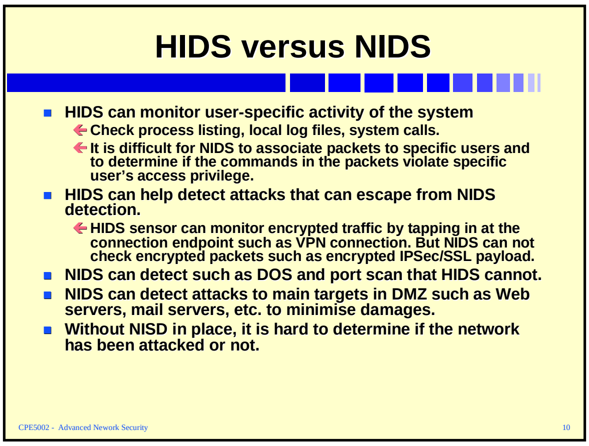
HIDS: Simple Example

- Using OS auditing mechanisms
  - BSM (host-based IDS) on Solaris: log all direct or indirect events generated by a user
  - Log all system calls made by a program
- Monitoring user activities
  - Analyse and log shell commands issued by a particular user
- Monitoring executions of system programs
  - Analyse and log system calls made by *sendmaild and httpd.*



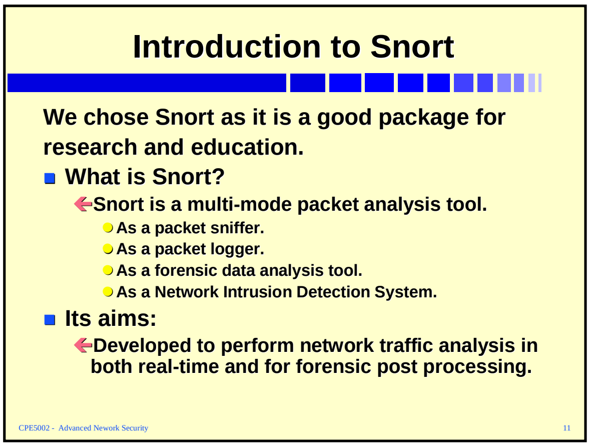
NIDS: Simple Example

- Deploying special sensors at strategic locations
  - E.g., Packet sniffing via *tcpdump* at routers.
- Inspecting network traffic
  - Watch for violations of protocols and unusual connection patterns.
- Monitoring user activities
  - Look into the data portions of the packets for malicious command sequences.



HIDS versus NIDS

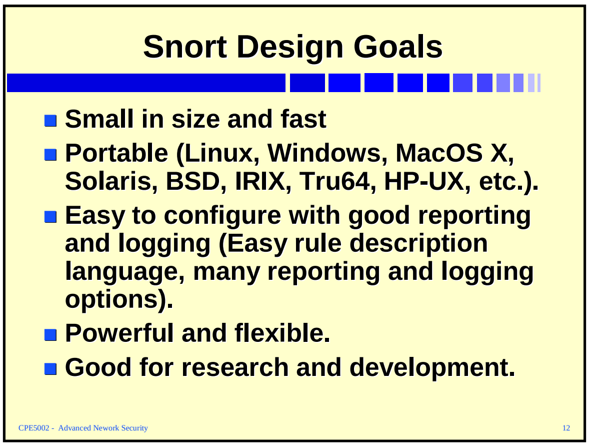
- HIDS can monitor user-specific activity of the system
  - Check process listing, local log files, system calls.
  - It is difficult for NIDS to associate packets to specific users and to determine if the commands in the packets violate specific user's access privilege.
- HIDS can help detect attacks that can escape from NIDS detection.
  - HIDS sensor can monitor encrypted traffic by tapping in at the connection endpoint such as VPN connection. But NIDS can not check encrypted packets such as encrypted IPSec/SSL payload.
- NIDS can detect such as DOS and port scan that HIDS cannot.
- NIDS can detect attacks to main targets in DMZ such as Web servers, mail servers, etc. to minimise damages.
- Without NISD in place, it is hard to determine if the network has been attacked or not.



Introduction to Snort

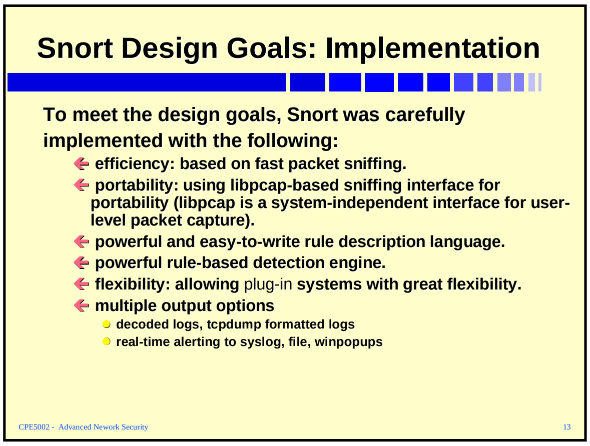
We chose Snort as it is a good package for research and education.

- What is Snort?
  - Snort is a multi-mode packet analysis tool.
    - As a packet sniffer.
    - As a packet logger.
    - As a forensic data analysis tool.
    - As a Network Intrusion Detection System.
- Its aims:
  - Developed to perform network traffic analysis in both real-time and for forensic post processing.



Snort Design Goals

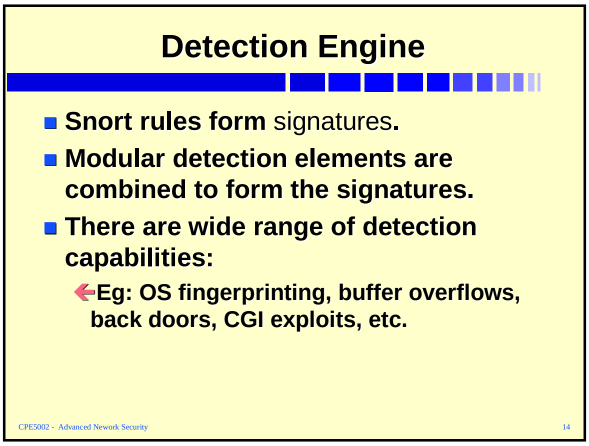
- Small in size and fast
- Portable (Linux, Windows, MacOS X, Solaris, BSD, IRIX, Tru64, HP-UX, etc.).
- Easy to configure with good reporting and logging (Easy rule description language, many reporting and logging options).
- Powerful and flexible.
- Good for research and development.



Snort Design Goals: Implementation

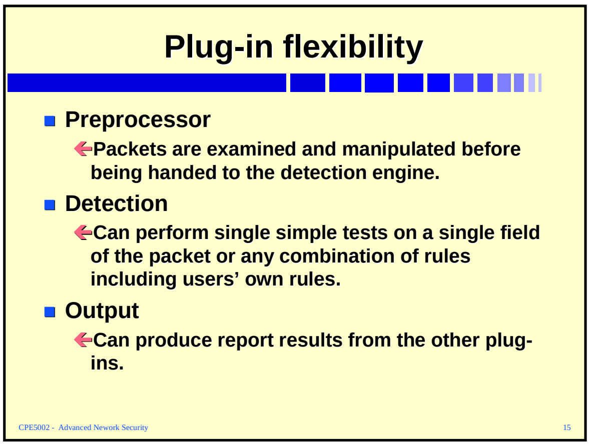
To meet the design goals, Snort was carefully implemented with the following:

- efficiency: based on fast packet sniffing.
- portability: using libpcap-based sniffing interface for portability (libpcap is a system-independent interface for user-level packet capture).
- powerful and easy-to-write rule description language.
- powerful rule-based detection engine.
- flexibility: allowing plug-in systems with great flexibility.
- multiple output options
  - decoded logs, tcpdump formatted logs
  - real-time alerting to syslog, file, winpopups



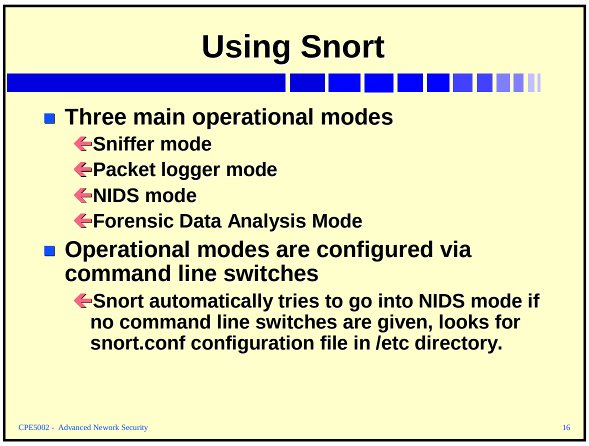
Detection Engine

- Snort rules form signatures.
- Modular detection elements are combined to form the signatures.
- There are wide range of detection capabilities:
  - Eg: OS fingerprinting, buffer overflows, back doors, CGI exploits, etc.



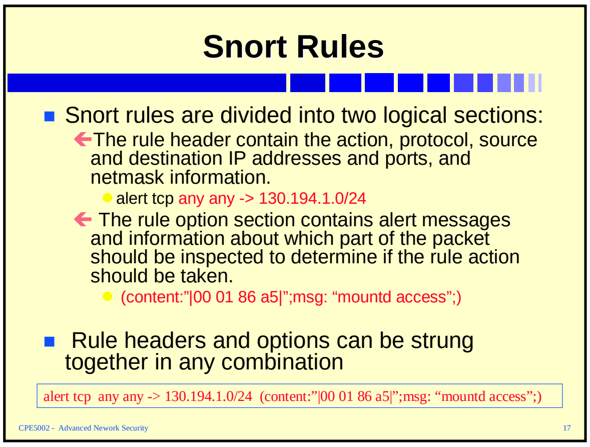
Plug-in flexibility

- Preprocessor
  - Packets are examined and manipulated before being handed to the detection engine.
- Detection
  - Can perform single simple tests on a single field of the packet or any combination of rules including users' own rules.
- Output
  - Can produce report results from the other plug-ins.



Using Snort

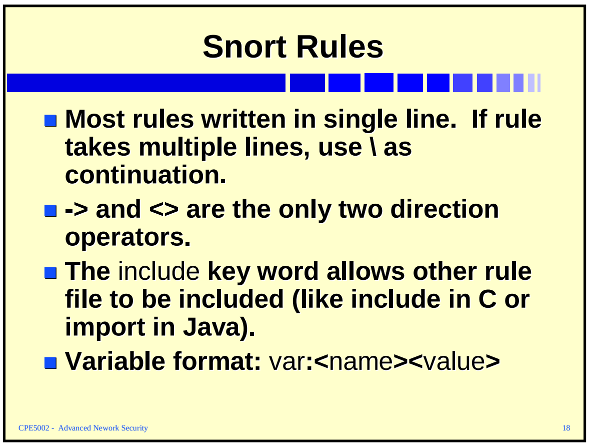
- Three main operational modes
  - Sniffer mode
  - Packet logger mode
  - NIDS mode
  - Forensic Data Analysis Mode
- Operational modes are configured via command line switches
  - Snort automatically tries to go into NIDS mode if no command line switches are given, looks for snort.conf configuration file in /etc directory.



Snort Rules

- Snort rules are divided into two logical sections:
  - The rule header contain the action, protocol, source and destination IP addresses and ports, and netmask information.
    - alert tcp any any -> 130.194.1.0/24
  - The rule option section contains alert messages and information about which part of the packet should be inspected to determine if the rule action should be taken.
    - (content:"|00 01 86 a5|";msg: "mountd access";)
- Rule headers and options can be strung together in any combination

```
alert tcp  any any -> 130.194.1.0/24  (content:"|00 01 86 a5|";msg: "mountd access";)
```



Snort Rules

- Most rules written in single line. If rule takes multiple lines, use \ as continuation.
- -> and <> are the only two direction operators.
- The include key word allows other rule file to be included (like include in C or import in Java).
- Variable format: var:<name><value>

Snort Rule Syntax

- Rule Actions:
  - Alert : generate an alert using the selectd alert method and log the packet.
  - Log : log the packet.
  - Pass : ignore the packet.
  - Activate : alert then turn on another dynamic rule.
  - Dynamic: remain idle until activated by a rule, then act as a log rule.

Define Your Own Rules

- **You can also define your own rule type. Then use it as rule action.**
- ruletype redalert **{**

```
 type alert output
alert_syslog LOG_AUTH  LOG_ALERT
output database: log, mysql, user=snort dbname=snort host=localhost
}
```

- **This example will create a rule type (**redalert**) that will log to** syslog **file and** MySQL **database**
-

Rule Header Features

- **IP addresses**
  - **negation, CIDR blocks (A CIDR block is simply another term for a subnet. CIDR: Classless Internet Domain Routing - Example: "130.194.226.13/24" means "that subnet which contains address 130.194.226.13 and whose subnet mask begins with 24 1s (the rest 0s).").**
- **TCP/UDP ports**
  - **negation, ranges, and greater than/less than**
- **Uni and bi-directional port and address consideration**

Some Rule Option Features

- **IP TTL**
- **Fragment size**
- **TCP Flags**
- **TCP Ack number**
- **TCP Seq number**
- **Payload size**
- **Content**
- **Content offset**
- **Content depth**
- **Session recording**
- **ICMP type**
- **ICMP code**

Other Feature of Snort

- **Snort is a packet sniffer and it can be used to analyse traffic in real-time.**
- **You can write your rules to pick up most sorts of things that are involved with intrusions. For example, you can write rules to catch intrusions by exploiting:**
  - **SQL/ODBC, ActiveX, Java/JavaScript, Macro Viruses**

Basic Snort Usage: examples

- **Snort has three main modes:**
  - **Sniffer mode: read packets and display on console.**
    - **E.g., >Snort -dev**
    - **v: verbose; d: dump application data; e: extensive**
  - **Packet Logger: read packets and log information about them to the disk.**
    - **E.g., > snort –dev –l ./log –h 192.168.1.0/24**
    - **l: log, h: only capture packets relative to the host**
  - **NIDS: analyse packets and match them against user defined rules and perform actions.**
    - **E.g., > snort –dev –l ./log –c snort.conf**
    - **add –D will have snort run as daemon.**
    - **-A [fast | full | unsock | non]**
    - **-b for binary (tcpdump) format; faster.**

/etc/snort/snort.conf

- Snort read the snort.conf file for the default variables, additional pre/post processing plug-in (if any), output specification (to a mysql for example), and a set of rule files. For example

```
include bad-traffic.rules
include exploit.rules
include scan.rules
include finger.rules
include ftp.rules
include telnet.rules
include rpc.rules
include dns.rules

etc.
```

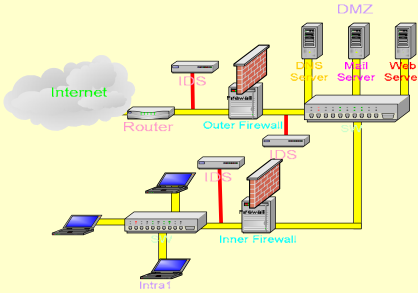
HIDS: Host-based Intrusion Detection

- Detect and examine malicious activity (same as network-based intrusion detection.)
- Optimize for monitoring individual hosts.
- Monitor system network activity, file system, log files, user actions.
- Integrate the finding of several host-based intrusion detection provide unified view of multiple systems in the network.
- Detect escalation of privileges for a user or system account. (from guest user to have admin privilege).
- NIDS can not usually see or interpret such actions which takes place on a host.

A Host-based IDS : Tripwire

- Developed at Purdue Univ. 1992 by Dr. Eugene Spafford and Gene Kim
  - http://www.tripwire.org/
- Commercial evaluation version tripwire3.0 (with manager and server, run on both Linux/windows) available at http://www.tripwire.com/downloads/
  - Tripwire managers provide GUI and unified interface to monitor multiple instances of tripwire program.
  - It can also monitor configuration of routers/switches.
- Also have a look into BSM (host-based IDS) on Solaris.

IDSs and Firewalls



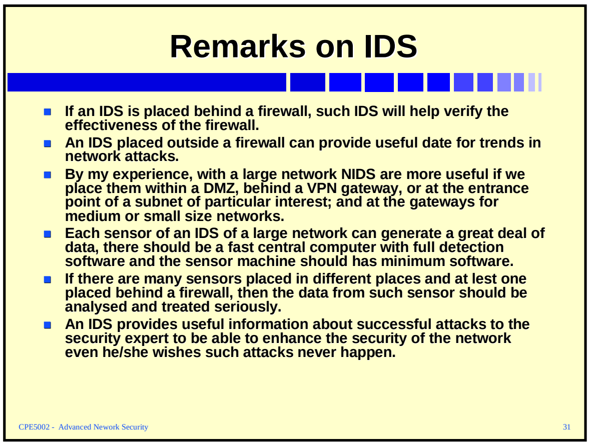
Borrowed from Chow – Another uni

Remarks on IDS

- In a large corporate, it is not easy to decide where to place the detection engine because the corporate may have many different gateways.
  - The DE can be placed either outside or inside the gateway protected by a secure firewall. But this may not be possible due to the corporate security policy or cost considerations.
  - If the DE is placed inside a firewall or behind a firewall, the whole IDS is better protected but we will not fully utilise the IDS.
  - It is difficult to make an optimal decision to place the sensor in a large network where there are many possible places that need an IDS.
  - In a large network, there is likely very high bandwidth usage, a gateway may require many sensors working to balance the network workload.

Remarks on IDS

- IDS together with firewalls increase the defence of a network, however in order to be able to provide a high security environment, you need to be able to combine your security analysis, cryptographic skill, and other network security components.
- IDS is vital in defending a network against intrusion, however it may also generate more network load and security holes.
- Different IDS design approaches allow different intrusions to be detected, however to detect many attacks from wireless mobile attackers and act in time are still beyond the capacity of current IDSs.
- Though an attack can be detected and logged by an IDS quickly but it is difficult to make rapid and proper response. This depends so much on the knowledge and skill of the security expert and the organisation's security policy (the skill of configuration of firewalls and other network security component will not be very useful under this circumstances).
- So how do I deal with an intrusion?
  - From my experience and other people's in the field, we strongly recommend any security professional to prepare a plan of actions in advance to deal with intrusions because it may take you not only hours but days before you can make an appropriate response.



Remarks on IDS

- If an IDS is placed behind a firewall, such IDS will help verify the effectiveness of the firewall.
- An IDS placed outside a firewall can provide useful date for trends in network attacks.
- By my experience, with a large network NIDS are more useful if we place them within a DMZ, behind a VPN gateway, or at the entrance point of a subnet of particular interest; and at the gateways for medium or small size networks.
- Each sensor of an IDS of a large network can generate a great deal of data, there should be a fast central computer with full detection software and the sensor machine should has minimum software.
- If there are many sensors placed in different places and at lest one placed behind a firewall, then the data from such sensor should be analysed and treated seriously.
- An IDS provides useful information about successful attacks to the security expert to be able to enhance the security of the network even he/she wishes such attacks never happen.