

This module will discuss what Intrusion Detection systems are, how they work, and what they can do. It will also cover some methods of evading detection by an IDS, and what the network administrator can do to counter these methods. Finally there will be a brief mention of various currently available IDS products as well as a discussion of how IDS can be deployed.



## Instructional Objectives

- Describe the core functions of Intrusion Detection Systems (IDS)
- Describe a host-based IDS
- Describe a network-based IDS
- Compare and contrast host-based and network-based IDS; including advantages and disadvantages of each
- Describe a signature-based IDS
- Describe an anomaly-based IDS
- Compare and contrast signature-based and anomaly-based IDS; including advantages and disadvantages of each
- Describe IDS evasion techniques and countermeasures
- Describe Practical Problems associated with administering IDS
- Describe IDS implementation strategies

The students will be able to do the above after this module.



## Overview

- Evolution of IDS
- Host based IDS
- Network IDS
- Signature-based IDS
- Anomaly-based IDS
- Evasion and countermeasures
- Deploying IDS



These are the topics that will be covered.



## What is an Intrusion Detection System?

Device on a network that monitors traffic and/or host activity looking for the following:

- Malicious traffic, such as attempts to circumvent identification & authorization or other access controls
- Reconnaissance traffic, such as port scans
- Unusual traffic: type, level, source, etc.
- Activity on host systems that is outside of known patterns

Device then logs and reports activity in prescribed manner

What is an IDS?

An IDS is essentially a network burglar alarm, similar to the alarms placed on doors and windows of a building. If a potential intruder is testing doors and windows, or trying to break in some other way, the burglar alarm will generate an alert. The alert is generally user-configurable: it may simply set off a local siren; it may turn on floodlights; it may call the police; or it may do some combination of the above. The IDS can be configured to log traffic, to generate an alert or console message, or to page the system administrator. The action that the IDS takes must be configured according to site policy and regulations.



## Evolution of the IDS -1

Computers can generate lots of log/audit data

System administrators began to write tools to automate the process of logging/analysis

These tools evolved and were modified to focus on security-related issues in the late 1980's

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 5

As computers were used for more and more tasks, the amount of log and audit data grew at a rapid pace. Soon the amount of log data was too great for humans to read and evaluate effectively. It was only natural that the early system administrators would automate the process of acquiring and parsing log and audit data. Eventually, as security became a concern, the automation of security-related tasks became a separate audit entity. Now, instead of monitoring who was using how much disk space and CPU time, personnel could see who was logging in and out and when, and who might be trying to get in without authorization. This type of auditing on hosts became more and more common and eventually led to auditing of network data packets as well.



## Evolution of the IDS -2

Network Security Monitor was developed at the University of California Davis

This was an early IDS-like tool to analyze actual traffic vice log entries

Essentially a packet sniffer feeding data to an analysis engine

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 6

In the late 1980s, when the Personal Computer boom was well underway, a tool called *Network Security Monitor* was created at the Davis campus of the University of California. Until this time, security-related tools were gathering and parsing log data; NSM was the first tool to gather packets right off the wire and examine them directly vice going through the logging process. At the time, networks were broadcast/shared media connections: all packets addressed to one machine on a network went to **all** machines on the network, but only the machine for which the packets were addressed would respond. Placing a network interface into *promiscuous mode* allowed the interface to collect all packets, even those that were addressed to other machines. This is principle behind packet sniffers and is still in use today in tools like *tcpdump*.



## IDS Terminology

- Sensor
- Analyzer
- Alert mechanism
- Logging mechanism
- Use interface
- False negative
- False positive
- Honeygot (as an optional, supporting component)



*Demo - honeyd*

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 7

### *Sensor*

Sensors are responsible for collecting data. The input for a sensor may be any part of a system that could contain evidence of an intrusion. Example types of input to a sensor are network packets, log files, and system call traces. Sensors collect and forward this information to the analyzer.

### *Analyzer*

Analyzers receive input from one or more sensors or from other analyzers. The analyzer is responsible for determining if an intrusion has occurred. The output of this component is an indication that an intrusion has occurred. The output may include evidence supporting the conclusion that an intrusion occurred. The analyzer may provide guidance about what actions to take as a result of the intrusion.

### *Alert mechanism*

This is the component that generates the console message, the email, the numeric page, etc. As mentioned previously, this is typically configured by administrators based on the nature of the intrusion and its severity.

### *Logging mechanism*

This is the component that stores information in non-volatile storage for further analysis.

### *User interface*

The user interface to an IDS enables a user to view output from the system or control the behavior of the system. In some systems, the user interface may equate to a “manager,” “director,” or “console” component.

*False negative* – fails to identify an intrusion when one has in fact occurred

*False positive* – incorrectly identified as being an intrusion when none has occurred

### *Honeygot*

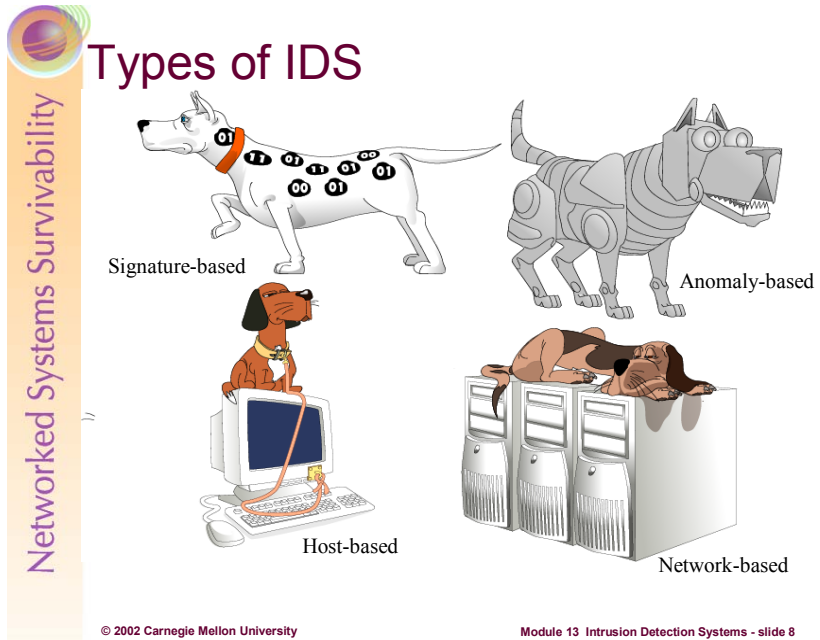
In addition to the essential components, an IDS may be supported by a “honeygot,” i.e., a system designed and configured to be visible to an intruder and to appear to have known vulnerabilities. A honeygot provides an environment and additional information that can be used to support intrusion analysis. The honeygot serves as a sensor for an IDS by waiting for intruders to attack the apparently vulnerable system. Having a honeygot serve as a sensor provides indications and warnings of an attack.

## Student Workbook – Module 13: Intrusion Detection Systems

Honeypots have the ability to detect intrusions in a controlled environment and preserve a known state [Allen 2000]. For more information on honeypots see:

<http://www.sans.org/newlook/resources/IDFAQ/honeypot3.htm>





There are four basic categories of Intrusion Detection Systems: host-based and network-based; and signature-based and anomaly based. The concept of host-based versus network-based speaks only to the type of machine or machines on which the IDS can be deployed; it does not address the method of analysis. Similarly, a signature-based IDS can be host-based or network-based, because the traffic analysis technique is not tied into the type of platform on which the analysis occurs. We will discuss host-based IDS first.



## IDS Topics

- Characteristics
- Advantages
- Disadvantages
- Typical deployment environment



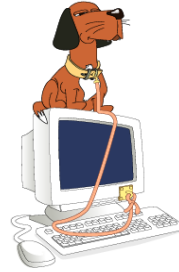
We will cover four topics relating to IDS:

- Characteristics: a description of the design and core functionality, and what is unique about this approach
- Advantages: strong points of this approach; how and why it may be better than other designs
- Disadvantages: weak points of this approach; how and why the design may fall short in a given set of circumstances
- Typical deployment environment: where and how this approach is most often used, described in terms of the aforementioned pros and cons

## Host-based IDS -1

### Characteristics

- Runs on single host
- Can analyze network packets, audit-trails, logs, integrity of files and directories, etc.
  - BlackIce Defender
  - Tripwire, Fcheck
  - Windows NT/2000 Event Viewer



© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 10

### Characteristics:

This is relatively straightforward: a host-based intrusion detection system (HIDS) runs on a single host. This one machine captures traffic off the wire, examines it against the preconfigured ruleset, and generates alerts as appropriate. No other network entities are required, and the host-based IDS typically collects data destined for the machine it is running on. This is the approach used by personal or individual IDS deployed on machines connected to the Internet by DSL or cable. BlackIce Defender is an inexpensive personal firewall and HIDS<sup>1</sup>. It is solely concerned with network data packets and does not concern itself with property changes on the systems itself (like changes to critical files, etc. This type of IDS can also be used on a single mission-critical server<sup>2</sup> on a corporate network: when resources are limited, such a machine will often be protected with an HIDS while the average user's desktop is not.

Other HIDS tools are designed to monitor the state of the operating system of a host. Fcheck<sup>3</sup> and Tripwire<sup>4</sup> are concerned with validating the integrity of files and folders. They can alert administrators if key files have changed—they're especially good for identifying rogue Trojan horse programs.

Although not normally described as a HIDS, Windows NT/2000 system monitor and event viewer can be used to audit for many security related events. Alerts can be configured to prompt administrators for these events—and the event viewer is a great tool for categorizing and organizing various log files. Because of these capabilities, these Windows tools function as HIDS in some respects.

<sup>1</sup> [http://www.iss.net/products\\_services/hsoffice\\_protection/blkice\\_protect\\_pc.php](http://www.iss.net/products_services/hsoffice_protection/blkice_protect_pc.php)

<sup>2</sup> [http://www.iss.net/products\\_services/hsoffice\\_protection/blkice\\_protect\\_server.php](http://www.iss.net/products_services/hsoffice_protection/blkice_protect_server.php)

<sup>3</sup> <http://www.geocities.com/fcheck2000/fcheck.html>

<sup>4</sup> <http://www.tripwire.org>



## Host-based IDS -2

### Advantages

- Relatively easy to deploy and to manage
  - Only one machine is involved
  - May require only one administrator
  - Creates single source of log and alert information
- Generally not resource intensive - in most cases
  - Often will not require CPU, memory, etc. beyond what is needed for OS and applications

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 11

### Advantages:

The host-based IDS generally requires only one machine, so deployment is relatively straightforward.

If the IDS is running on a particular machine, the system administrator will likely be tasked with monitoring the HIDS; in this case no additional personnel would be required. Some host based systems like Tripwire can send individual system data to a central console. This feature is not necessarily a function of HIDS, as using central console architecture can also be deployed for network-based IDS. Regardless, this is one method of centrally monitoring multiple HIDS from a single location—this can become quite a burden as is addressed later.

Log entries and alerts are typically available in a single location, which simplifies the monitoring process.

Since the HIDS sensor is monitoring only itself, vice multiple machines on the network, the CPU and memory overhead is generally somewhat lower than that of a network-based IDS. This does depend on the product and the way it is implemented however.

Since most functions take place on a single machine, the host-based approach adds only minimal traffic – if any –to the network.



## Host-based IDS -3

### Disadvantages

- Works well for single machine; extremely labor-intensive to monitor multiple machines each running a host-based IDS
- If the host is compromised, the IDS may cease to function and thus no more alerts will be generated

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 12

### Disadvantages:

A host-based IDS can work well if an organization wishes to protect one or two machines. However, in order to monitor several machines using several HIDS, the organization will have to address the following issues:

- Keeping all versions, patches, and configurations consistent across all platforms
- Collating and analyzing information and alerts from multiple sources multiple geographic locations, even if the systems are only in different buildings

This can be quite an administrative challenge and potentially a burden. Obviously, if the host system is compromised than the intrusion detection capability can be turned off.



## Host-based IDS -4

### Typical deployment environments

- Single mission-critical machine that can be monitored from the console
- User's desktop machine, e.g. a DSL or cable user's system

### Typical Deployment Environments:

If an organization has one mission-critical server to monitor, this can often be done via physical access, i.e. sitting down at the keyboard in the NOC or server room. The concept of the console also includes the use of a terminal server on systems that support it (many Unix, Windows platforms; others with third-party software).

One of the most common HIDS deployments is in the small office/home office (SOHO) environment that utilizes DSL or cable modems for broadband connectivity. BlackIce Defender is commonly used here; allowing users to monitor their network connection for intrusions.

## Network-based IDS -1

### Characteristics

- Network monitor
  - Passively captures traffic and inspects it
- Can also function in a client-server model
  - Sensors are located on multiple machines across the network
  - All sensors feed data to console
  - Console machine handles logging and alerting



© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 14

### Characteristics:

A network-based intrusion detection system (NIDS) monitors the traffic on its network segment as a data source. This is generally accomplished by placing the network interface card in promiscuous mode to capture all network traffic that crosses its network segment. Network traffic on other segments, and traffic on other means of communication (like phone lines) can't be monitored.

Network-based ID involves looking at the packets on the network as they pass by some sensor. The sensor can only see the packets that happen to be carried on the network segment it's attached to. Packets are considered to be of interest if they match a signature or a normal behavior. Since most NIDS are signature-based, we'll focus on these. Three primary types of signatures are string signatures, port signatures, and header condition signatures.

String signatures look for a text string that indicates a possible attack. An example string signature for UNIX might be "cat "+ +" > /.rhosts" , which if successful, might cause a UNIX system to become extremely vulnerable to various network attacks. To refine the string signature to reduce the number of false positives, it may be necessary to use a compound string signature. A compound string signature for a common Web server attack might be "cgi-bin" AND "aglimpse" AND "IFS".

Port signatures simply watch for connection attempts to well-known, frequently attacked ports. Examples of these ports include telnet (TCP port 23), FTP (TCP port 21/20), SUNRPC (TCP/UDP port 111), and IMAP (TCP port 143). If any of these ports aren't used by the site, then incoming packets to these ports are suspicious.

Header signatures watch for dangerous or illogical combinations in packet headers. The most famous example is WinNuke, where a packet is destined for a NetBIOS port and the Urgent pointer, or Out Of Band pointer set. This results in the "blue screen of death" for some Windows systems. Another well-known header signature is a TCP packet with both the SYN and FIN flags set, signifying that the requestor wishes to start and stop a connection at the same time.<sup>5</sup>

<sup>5</sup> [http://www.sans.org/newlook/resources/IDFAQ/network\\_based.htm](http://www.sans.org/newlook/resources/IDFAQ/network_based.htm)

## Student Workbook – Module 13: Intrusion Detection Systems

In a network-based client-server IDS implementation:

Each host that is being monitored will have its sensor. These sensors are not exactly packet sniffers since they capture only the traffic addressed to the machine on which they run. The sensors capture the traffic and feed data to the central management console. How much analysis is performed on the individual hosts can vary depending on the product in use and on the local site policy. The management console system then examines the packet data against the ruleset and issues alerts as necessary. All of the data from the individual machines is then stored on the central console machine.





## Network-based IDS -2

### Advantages

- Positioned properly, can test effectiveness of firewalls, router access lists, etc.
- Can monitor multiple machines from one physical and logical location
- Console can generate an alert if a monitored machine/network has ceased to send information
- Operator can see patterns in traffic
  - Amount
  - Type

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 15

### Advantages:

A small network with limited personnel resources can benefit greatly from a network-based IDS. One person can monitor a number of machines effectively, reducing personnel requirements. The management console often can be configured to send an alert if communication is lost with a monitored machine. Having all the data from multiple machines in one location can allow the operator to spot trends and patterns – and anomalies – in traffic.

If a network IDS is placed strategically (see deployment scenarios later in module) then critical network traffic can be inspected and centrally analyzed. Additionally, trend information can be obtained as to what kinds of data traffic is traversing the IDS. This can lend itself to optimization and filtering implementations. It can also be used to evaluate and fine-tune access control rules on firewalls and routers.



## Network-based IDS -3

### Disadvantages

- Since it is capturing all network packets, can produce large log/alert files
  - Can be difficult to cull through vast amount of information
- Console machine generally must be quite powerful, similar to a workgroup server
- If console machine goes down then multiple machines may be left unmonitored
- Communication from sensors to console may increase overall network traffic levels

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 16

### Disadvantages:

This biggest disadvantage of NIDS is the problem of sifting through the (potentially) vast amount of log/alert files. It is important to select only those signatures that apply to your network traffic, thereby minimizing the number, size of the alerts and logs.

Since the central management console is doing a lot of work – collating, analyzing, and storing data – it must be a fairly powerful machine. Quite often the console must be designed as a small workgroup server vice a user workstation. This can add significantly to the expense of deploying a network-based IDS.

If no monitoring or analysis is taking place on the individual machines, these machines may go unmonitored if the central console should go down.

In addition to the normal level of network traffic on the segment, the monitored workstations are sending information back to the central management system; this may raise the level of network traffic to an unacceptable level on a marginal network. This may be a concern for some HIDS implementations as well.

Traditional network IDS packet capturing has its limitations as well. As mentioned previously, the IDS can capture packets only on the local network—therefore, multiple packet capturing sensors may be required depending on the deployment scheme.



## Network-based IDS -4

### Typical Deployment Environments

- Generally seen in a corporate or organizational environment
  - Multiple assets to protect
  - Maintain personnel levels
- Inside/outside protected network to monitor firewall effectiveness, public hosts (DMZ)

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 17

### Typical Deployment Environment:

Network-based IDS are most often deployed in organizational environments where there are multiple distributed assets and not enough personnel to monitor them individually. Larger organizations can more easily change to meet personnel requirements, but larger organizations also typically have more assets to protect.

A number of available products have sensors that will run on many different operating systems. In a case such as this, even though the central console may only run on one or two platforms, the sensors can be deployed across the site to monitor a mixed network.

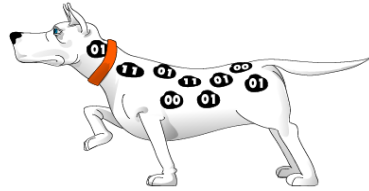
Positioning NIDS just inside the firewall can be beneficial, because it's alerts can be used to optimize the rule set on the firewalls. Placing a NIDS in the DMZ is also warranted, as the DMZ likely contains services (like the organization's public web server) that may be popular targets.

We will discuss specific deployment scenarios later in the module.

## Signature-based IDS -1

### Characteristics

- Uses known pattern matching to signify attack
- Can identify intrusions from packet header/data
- May use Boolean operators in rule set
  - 'this\_string'
  - 'this\_variable' AND 'that\_number'
  - 'this\_string' AND 'that\_variable' NOT 'that\_tcp\_flag'



© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 18

### Characteristics:

Attack signature detection (sometimes called “misuse detection”) identifies patterns corresponding to known attacks. This includes passive protocol analysis through the use of network packet sniffers. It also includes signature analysis through the interpretation of a series of packets (or a piece of data contained in those packets) that are determined, in advance, to represent a known pattern of attack. The attack signature may also be manifest in audit records, logs, or in changes in the compromised system [Allen 2000].

Data packets contain many types of information; some of this information is in the form of ASCII strings. A particular attack will often have certain strings that occur in every implementation of the attack. By searching for a given string – usually text – the IDS will be able to detect and to flag suspicious or known malicious packets.

Boolean Operators are the terms AND, OR, and NOT. They are used to describe conditions on which a later action might be dependent. For example, *if* a car’s engine has (air) AND (fuel) AND (spark) NOT (water in the gas tank) *then* the car will start. However, if any of these conditions marked with AND is not met, the car will not start. Similarly, if the condition marked by NOT is true, the car will not start.



## Signature-based IDS -2

### Advantages

- Widely available
- Fairly fast
- Not overly complex
- Easy to implement
- Easy to update



© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 19

### Advantages:

Signature-based IDS are fast and generally reliable: computers can compare a number of items – such as this AND that NOT the other - and reach a result very quickly. Such comparisons are easy to implement as well: either there is a match or there is not a match, so a binary comparison is all that is needed. Typically this type of IDS uses a signature database, similar to what anti-virus vendors use, to determine what packets may be malicious.

Signature-based IDS have the potential for lower false alarm rates, and the contextual analysis proposed by the intrusion detection system is detailed, potentially making it easier for the security officer using this intrusion detection system to take preventive or corrective action.<sup>6</sup>

---

<sup>6</sup> [http://www.sans.org/newlook/resources/IDFAQ/knowledge\\_based.htm](http://www.sans.org/newlook/resources/IDFAQ/knowledge_based.htm)



## Signature-based IDS -3

### Disadvantages

- Simple
  - No artificial intelligence
  - Cannot detect attacks for which it has no signature
- Must be updated for each new attack and attack variant
  - Lag time from new exploit to update can be dangerous
  - ‘New’ attack variant can be created by changing a single string
- No protection against abuse of privileges by insiders

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 20

### Disadvantages:

While signature-based IDS are generally easy to update, the fact that they **must** be updated can be a drawback. Without any kind of artificial intelligence, the IDS can rely only on its signature database: if an attack is not listed, it will not be flagged and no alert will be generated. A signature-based IDS must be updated in a timely fashion as new attacks are developed. To complicate matters further, if an IDS has a one-string signature of ATTACK, and an attacker modifies this string to KITTEN, the IDS will not pick up the attack if KITTEN is not in the signature database.

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. Signature-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.<sup>7</sup>

Another problem is that users who abuse their privileges inside a protected network likely will not set off the alarms of an IDS. For example, if users with elevated privileges (like IT staff, some managers, etc.) are using these to access accounting and payroll information (which is supposed to be confidential) then any IDS will be unlikely to help here.

---

<sup>7</sup> [http://www.sans.org/newlook/resources/IDFAQ/knowledge\\_based.htm](http://www.sans.org/newlook/resources/IDFAQ/knowledge_based.htm)



## Signature-based IDS -4

### Typical deployment environments

- Can be host-based or network-based
- Nearly everywhere
- Currently there is a lack of alternatives

### Typical Deployment Environment:

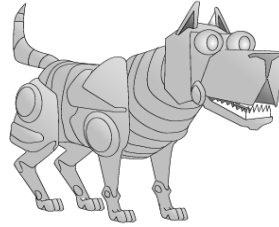
Nearly every IDS available today has a signature-based component; many are completely signature-based. Currently this is the most reliable approach to intrusion detection. Anomaly-based IDS (which we will discuss shortly), the only real alternative to signature-based, is currently unreliable due to some inherent limitations.

There are products that incorporate both signature and anomaly based approaches, however the vast majority rely solely on signatures.

## Anomaly-based IDS -1

### Characteristics

- Uses statistical variance or (sometimes) artificial intelligence (AI) engine to evaluate traffic, normal usage behaviors
- Does not use signatures; must be 'trained', i.e. given a baseline of 'normal' traffic



© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 22

### Characteristics:

Anomaly-based intrusion detection techniques assume that an intrusion can be detected by observing a deviation from normal or expected behavior of the system or the users. The model of normal or valid behavior is extracted from reference information collected by various means. The intrusion detection system later compares this model with the current activity. When a deviation is observed, an alarm is generated. In other words, anything that does not correspond to a previously learned behavior is considered intrusive.<sup>8</sup>

Expected behavior is defined, in advance, by a manually developed profile or by an automatically developed profile. An automatically developed profile is created by software that collects and processes characteristics of system behavior over time and forms a statistically valid sample of such behavior. Note that unexpected behavior is not necessarily an attack; it may represent new, legitimate behavior that needs to be added to the category of expected behavior.

An anomaly-based IDS starts with a clean slate. Typically the IDS will be installed as a passive node on a network; from this position it will monitor traffic for a suitable length of time, such as 30 days. The time period must be long enough to encompass events such as increased traffic levels when reconciling accounts at the end of the month. Once the operators have decided the baseline is sufficient, the IDS will be deployed. For example, if Joe the Accountant logs in every Monday-Friday at 0700 for a month, and he then starts logging in on Saturdays, this activity may be flagged. Similarly, if Joe logs in from his desktop machine on the corporate LAN every day, then one Saturday he logs in from guest.bigcompetitor.com, this activity will very likely be flagged.

<sup>8</sup> [http://www.sans.org/newlook/resources/IDFAQ/behavior\\_based.htm](http://www.sans.org/newlook/resources/IDFAQ/behavior_based.htm)





## Anomaly-based IDS -2

### Advantages

- Can detect attempts to exploit new and unforeseen vulnerabilities
- Can recognize unusual traffic based on a number of characteristics:
  - Payload
  - Source address
  - Time
- Can recognize authorized usage that falls outside the normal pattern

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 23

### Advantages:

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 potentially 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 that has not been seen previously is dangerous.<sup>9</sup>

Anomaly-based IDS can also pick up unusual types and amounts of traffic. If Jane the HR Person typically looks at a few web-based forms on a certain server, but one day she downloads the entire HR database, the IDS should send an alert because this is an unusual amount and type of traffic.

---

<sup>9</sup> [http://www.sans.org/newlook/resources/IDFAQ/behavior\\_based.htm](http://www.sans.org/newlook/resources/IDFAQ/behavior_based.htm)



## Anomaly-based IDS -3

### Disadvantages

- Generally slower, more resource intensive compared to signature-based IDS
- Greater complexity, difficult to configure
- Higher percentages of false alerts

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 24

### Disadvantages:

Signature-based IDS can simply compare two items and decide if they match. An anomaly-based IDS must compare more complex objects – e.g. a complete ftp session – to the network baseline to determine if the traffic is ‘normal’. This process is much more complex than a yes/no answer, and as a result the anomaly-based IDS will generate more erroneous alerts.

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.<sup>10</sup>

Anomaly-based systems in general are more difficult to configure because a comprehensive definition of known and expected behavior for a system is required. This demands that the user discover, understand, represent, and maintain the expected behavior of their system. In many cases, automated support is provided but this takes time to develop and the data that is used must be unambiguous [Allen 2000].

---

<sup>10</sup> [http://www.sans.org/newlook/resources/IDFAQ/behavior\\_based.htm](http://www.sans.org/newlook/resources/IDFAQ/behavior_based.htm)



## Anomaly-based IDS -4

### Typical deployment environments

- Currently few typical deployments:
  - Anomaly-based IDS considered immature, too error-prone for widespread use
- As Artificial Intelligence and other approaches advance, may become more feasible and widely used

### Typical Deployment Environment:

Currently few typical deployment environments exist: Completely anomaly-based IDS are considered too error-prone for use in a production environment. There certainly are products available that combine some anomaly and signature approaches. As Artificial Intelligence technology advances, anomaly-based IDS will become more feasible. However, the greater reliability of the signature-based IDS indicates that signature- and anomaly-based IDS probably will exist side by side well into the future.



## How Can Intruders Detect IDS?

DNS entries reveals NID

Active identifying is possible based on

- Open ports
- Network interface status

NID components are inserted in network topology maps etc. or mentioned on public content (“We use product ABC”)

NID communication (e. g. between sensors and analyzer, alert messages) can be observed

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 26

Many administrators actually put the IDS domain name into the network’s DNS servers. Bad idea! This can help identify your IDS--which should remain as obscure if possible. If unauthenticated DNS zone transfers are permitted in the network, it is possible that intruders can learn information about the IDS system and thereby set it up for an attack.

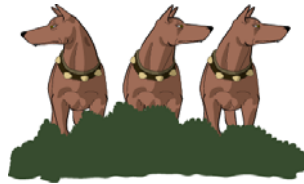
Almost all administrators rely on printed network maps to graphically illustrate the network. These should be kept secure and out of public areas. It may be wise to not include the IDS on this—thereby adding to the stealthy nature of your IDS.

Because traffic between consoles and sensors can be intercepted via packet sniffing, it may be useful to keep these systems (if possible) within an isolated admin network and VLAN.

Some IDS products use standard ports for transmitting traffic—this is well documented in the product’s literature and therefore available to intruders. If possible, change these to non-standard ports.



## IDS Evasion Techniques



Flooding/resource exhaustion

Crashing or compromising the underlying platform

Packet fragmentation

Staying below IDS thresholds



© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 27

We will discuss some common methods of evading IDS detection and alerting:

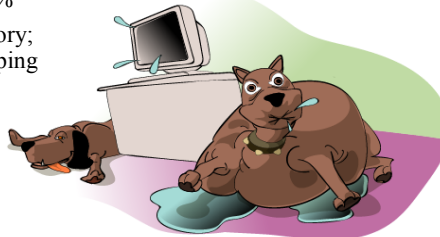
- Flooding and resource exhaustion
- Crashing or compromising the underlying platform
- Fragmentation of packets
- Utilizing attacks that are crafted purposely to not set off IDS alerts—that is, Staying below IDS thresholds

These methods can be quite simple or they can be rather elegant. Some can be automated and some must be completed manually; some can be a combination of both [manual and automated] functions. We will discuss how these methods can be deployed against your IDS, and then we will show you what to do about them.

## IDS Evasion Techniques -1

### Flooding/resource exhaustion

- Send high levels of traffic to IDS network segment
- Deplete resources until IDS drops packets or crashes
  - CPU utilization at 100%
  - Exhaust physical memory; induce excessive swapping
  - Fill local storage with messages and alerts: IDS will likely crash when file system is full



© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 28

### Flooding/Resource Exhaustion:

Each IDS platform has only a finite amount of resources. These limitations can generally be expressed numerically by referring to the hardware: the network is 100Mb, switched; the CPU is a P-III 500; it has 512MB RAM; it has 40GB of hard drive space, etc. Each combination of the various resources allows each IDS platform to handle a certain number of events during a given period of time; we will call this N. All the attacker has to do is provide or generate (N+1) events in the same given time period, and the last event will not be logged and will not generate an alert. If you have 5 targets of opportunity in front of you and you have only 4 rounds left, you will be in a similar situation. The key here is to optimize the resources that are currently available (which we will discuss later in Countermeasures).

This evasion method differs slightly from the next one, which is Crashing/Compromising the Underlying Platform.

## IDS Evasion Techniques -2

### Crashing or compromising the underlying platform

- Denial of Service (DoS) attack
  - Distributed or one-to-one
  - Flooding
  - Other, such as winnuke
- Remote exploit against Operating System or application
  - Buffer overflow
  - Misconfiguration
  - Gain privileges; disable IDS and purge logs

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 29

### Crashing/Compromising the underlying platform:

Where the previous slide covered resource exhaustion directed against the IDS application itself, here we will discuss the ideas of crashing or compromising the underlying platform.

Every IDS must have an underlying platform. Whether you're using ISS RealSecure on Windows or Snort on Linux, there is an operating system under the IDS that can be attacked. These attacks fall into two categories: crashing (Denial of Service) or compromising. We will talk about crashing first.

Several years ago there was a group of Denial of Service attacks that would crash a Windows NT 4.0 system with a single packet. Resource exhaustion can require careful planning and be rather time-consuming; sending a single crafted packet requires very little time on the part of the attacker. These vulnerabilities have long been patched, of course, but the concept is important. Certain web-based applications are vulnerable to buffer overflows: sending an intentionally malformed URL, for example, may cause the system to crash immediately or to become unstable, possibly crashing later for no apparent reason. This has tended to be a somewhat common vulnerability for some versions of Windows operating system. Unix operating systems generally tend to handle rogue applications better due to their memory management and security architecture design. In any case, crashing the operating system on which the IDS is running will by extension bring down the IDS as well; the IDS cannot function without the OS.

The other way to attack the operating system is to attempt a compromise. We are not going to cover all the ways in which this can occur, but we do want to mention one very important point: if the underlying operating system is compromised, and the intruder gains privileged access, i.e. root or Administrator, then he can (a) disable or reconfigure the IDS, and (b) purge the IDS logs. In short, if the intruder gains root, the organization can no longer rely on **any** information provided by the IDS.

## IDS Evasion Techniques -3

### Fragmentation of packets

- Construct intentionally fragmented packets
- Fragments are examined by IDS
- Each packet is too small to contain entire attack signature
- Packets are passed to destination
- Fragments are reassembled at destination
- Packet payload reaches destination intact



### Staying below IDS thresholds

- Low number of connections over long period of time
- Random addresses and ports wherever possible

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 30

### Fragmentation of Packets:

A signature-based IDS matches the content or other characteristics of a packet against a known set of indicators that will flag suspicious or known malicious traffic. For example, if a recent exploit contains the word ATTACK, the rule might be written to look for just that: the word ATTACK. However, if the sender of the malicious traffic breaks down the payload and creates three packet fragments – AT, TA, and CK, for example – the IDS will not generate an alert because the fragments contain strings that are too small to match anything in the signature database. At the same time, the IDS cannot flag every packet that contains two letters together like CK, because there are many valid reasons this letter combination would appear in traffic. The end result: none of the fragments are blocked, and they are assembled when they reach their destination. Bottom line: the IDS needs to do packet reassembly exactly as the destination host.

### Staying Below IDS Thresholds:

Some intruders can be very patient when leveraging an attack. Many scanning tools (such as Nmap) have stealth modes that are designed to operate such that an IDS may never notice it. Many attacks utilize random address and port generators to further confuse an IDS.





## IDS Evasion Countermeasures -1

Extensive resources and use of quotas

Building a robust IDS platform

Reassemble fragmented packets before examination



© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 31

The evasion techniques in the previous section can potentially be mitigated with a series of countermeasures:

- Extensive resources and the use of quotas
- Building a robust IDS platform
- Reassembling fragmented packets before examining them

We will look at these on the following pages.



## IDS Evasion Countermeasures -2

Extensive resources and use of quotas

- Fast CPU is desirable
- Multiple CPUs if IDS supports SMP
- Maximize amount of physical RAM
- Utilize fast storage, such as UW-SCSI hard drives in RAID 5
- Use large amount of storage
- Create quota so IDS does not completely fill file system
- Rule of thumb: multiply vendor's minimum recommendations by 5

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 32

Extensive resources and use of quotas

- Fast CPU is desirable
- Multiple CPUs if IDS supports SMP; however one very fast CPU is better than two slow ones
- Maximize amount of physical RAM – this allows the IDS to examine packets in RAM without swapping to a page file on the hard disk
- Utilize fast storage, such as UW-SCSI hard drives with optimized, hardware-based RAID configurations – using fast and redundant storage will allow the IDS to keep up with extensive logging
- Use large amount of storage—it's cheap, so over-allocate here
- Create quota so IDS does not completely fill filesystem—also should have archiving procedures in place so that only fresh logs remain on the active system
- Rule of thumb: multiply vendor's minimum recommendations by 5 – if the documentation says to allocate 2GB for logs, allocate 10GB instead.

Also consider utilizing multiple network cards and load-balancing the traffic so packets are not dropped.



## IDS Evasion Countermeasures -3

### Build a robust IDS platform

- Choose an operating system with a good security record
  - Then harden it!
- Do not run any services (web, email, etc.) on this machine
  - Unless it is a host-based IDS for that critical system
- Do not use this machine as a user's workstation
- Do not allow users to have accounts on this machine
- Control physical access to the machine

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 33

### Build a robust IDS platform:

- Choose an operating system with a good security record – for instance, OpenBSD generally is considered to have a better security record than Windows NT 4.0. The next step is to harden that operating system—see Host System Hardening module earlier in course.
- Do not run **any** services (web, email, etc.) on this machine – any listening service is a potential avenue for compromise
- Do not use this machine as a user's workstation, and do not allow users to have accounts on this machine – user accounts are a potential avenue for compromise
- Control physical access to the machine – a system that does not have vulnerabilities that can be exploited remotely may have some that can be exploited locally, i.e. with keyboard access. In the worst case scenario the intruder could simply unplug and/or smash the machine: no more IDS.



## IDS Evasion Countermeasures -4

### Reassemble fragmented packets before examination

- Most IDS now do this by default
- Be sure that your IDS is implemented correctly – do not assume that it is
- If fragments are designed to overlap or to leave a gap
  - The fragments should be logged and dropped
  - The reassembly of the bogus packet should not affect the underlying platform

### Fragment Reassembly:

In some cases, fragmentation and reassembly is handled at the Firewall, however most IDS products allow for this. Snort has preprocessors that handle this, prior to any actual packet inspection.

It should be noted that there is a processing and memory burden that comes with this capability. If an intruder sends a tremendous amount of fragmented packets (especially ones that are crafted so that it is impossible to reassemble them) at your IDS, it may be unable to inspect legitimate ones.



## Encryption and IDS

### Virtual Private Network (VPN)

- Traffic is generally decrypted as it passes the local perimeter, i.e. at the router or at the firewall
- Traffic traversing the local network is passed in the clear
- IDS on internal network can function normally
- External IDS can be defeated

### End to End Encryption

- Some IPSec/SSH implementations, SSL, PGP, etc.
- Key escrowing on IDS possible but complicated
- Generally defeats IDS capability

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 35

VPNs typically do not perform end-to-end encryption. Usually the encryption ends at the VPN server which is usually inside the protected network or in a DMZ. Therefore, depending on the placement of your VPN Server and your IDS, it is possible to allow for inspection of all plaintext packets.


End to End Encryption presents a problem for IDS systems. Because the data is encrypted throughout its transport, the IDS is basically defeated. Secure Sockets Layer (SSL) is the most widely used method of encryption on the Internet. Because it is end-to-end, IDS will not be able to analyze all SSL traffic. Programs like PGP and SSH and many other application layer encryptions implementations use public key cryptography. By escrowing appropriate keys on the IDS, it may be possible to receive the traffic, decrypt it, and then inspect it. This can be a very daunting task and is almost always unmanageable.

Message encryption is a problem, especially for network-based intrusion systems. Encryption makes the practice of looking for particular patterns in packet bodies futile. Useful analysis can be performed only after the message has been decrypted on the target host, and this often occurs within a specific application. Driven by commercial and defense needs, message encryption is likely to substantially increase in the near future. For this reason alone, greater emphasis needs to be placed on host-based or application-based ID systems that have the ability to view message content even if the message is encrypted in transit. While encryption may make intrusion detection more challenging, this is likely to be offset by its positive benefits. With effective encryption, theft of information becomes much harder, and the motivation to penetrate computer networks is significantly diminished. IPSEC provides a mechanism (encapsulating security payload) that can be used to hide both the contents and addresses of network packets between cooperating agents such as firewalls. However, this renders the actual source, destination, and contents of the packet opaque while they are in transit between agents. If an IDS is positioned along the agent-to-agent path, it will be unable to determine the real origin or destination of the traffic [Allen 2000].

Networked Systems Survivability

## Currently Available IDS Products -1

Closed-source  
Open-source



© 2002 Carnegie Mellon University      Module 13 Intrusion Detection Systems - slide 36

### Currently Available IDS Products

Closed-Source – these are generally produced and marketed by companies, usually with a fairly restrictive product licensing scheme; also called “shrink wrapped” software, after the boxes in which the products are shipped. The source code is unable to be inspected by anyone outside of the development organization—hence the term closed source. Microsoft products are examples of closed-source products.

Open-source – programs that are generally free (as in no monetary cost); the program and its source code are freely available. Linux and many Linux applications fall into this category.



## Currently Available IDS Products -2

### Closed-source examples

- Cisco secure IDS –  
[www.cisco.com/warp/public/cc/pd/sqsw/sqidsz](http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz)
  - Signature-based; formerly NetRanger IDS
  - Appliance based, can be integrated with Cisco switches
- ISS – [www.iss.net](http://www.iss.net)
  - RealSecure product line – signature-based; host and network-based

### Some Currently Available IDS Products

Cisco's Secure IDS product was formerly known as NetRanger and has been one of the more popular commercial IDS solutions. It is signature-based and has automated features that make updating the signature database transparent to the administrator. It is an appliance-type system, in that it does not run on top of generic operating systems like Unix or Windows. It has its own Cisco proprietary operating system that is potentially more secure out of the box than non-appliance based IDS products. It can also be integrating with Cisco switches, potentially relieving the burden that switched environments can have on IDS.

Internet Security Systems, now called just ISS, offers a product called RealSecure. This is a signature-based IDS that can be either host- or network-based depending on the needs of the organization, i.e. the sensor and server can exist on the same machine, but this is not required.

RealSecure uses a three level architecture consisting of a network-based recognition engine, a host-based recognition engine, and an administrator's module. The network recognition engine runs on dedicated workstations to provide network intrusion detection and response. Each network recognition engine watches the packet traffic traveling over a specific network segment for attack signatures — telltale evidence that an attempted intrusion is taking place. When a network recognition engine detects unauthorized activity, it can respond by terminating the connection, sending email or pager alerts, recording the session, reconfiguring selected firewalls, or taking other user-definable actions. In addition, a network recognition engine passes an alarm to the administrator's module or a third-party management console for administrative follow-up and review. The host-based recognition engine is a host-resident complement to the network recognition engine. It analyzes host logs to recognize attacks, determines whether the attack was successful or not, and provides other forensic information not available in a real-time environment. Each host engine is installed on a workstation or server, and thoroughly examines that system's logs for tell-tale patterns of network misuse and breaches of security. The host engine reacts to prevent further incursions by terminating user processes and suspending user accounts. It can send alarms, log events, send traps, send e-mails, and execute user-defined actions [Allen 2000].

## Currently Available IDS Products -3

### Open-source examples

- Snort – [www.snort.org](http://www.snort.org)
  - Signature-based
  - Extensive rule set
  - Facility to create custom rules
- Demarc PureSecure – [www.demarc.com](http://www.demarc.com)
  - Signature-based network monitoring tool
  - Uses snort signatures/engine
  - Client/server—centralized monitoring

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 38

### Snort:

Snort is a packet sniffer and logger that can be used as a lightweight network intrusion detection system. It features rules based logging to perform content pattern matching and detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, and much more. Snort has real-time alerting capability, with alerts being sent to syslog, Server Message Block (SMB) "WinPopup" messages, or a separate "alert" file. Snort is configured using command line switches and optional Berkeley Packet Filter commands. The detection engine is programmed using a simple language that describes per packet tests and actions. Ease of use simplifies and expedites the development of new exploit detection rules. For example, when the IIS Showcode web exploits were revealed on the Bugtraq mailing list, Snort rules to detect the probes were available within a few hours.<sup>11</sup>

One of the complaints surrounding Snort is that it requires a good deal of post-alert analysis. One open-source tool under heavy development that can help with this is called Analysis Console for Intrusion Databases (ACID).<sup>12</sup>

### Demarc PureSecure:

Demarc PureSecure is a free (as of the time of this writing) IDS console that provides additional capabilities to SNORT as well as a nice graphical user interface. It is categorized as open-source because most of its pieces are just that—it uses Snort as its IDS engine, MySQL as its database, and [can use] Apache as its web server platform. Much of the program is PERL and CGI based and this code can be viewed in the Linux/Unix distribution.

Its features have been extracted from [www.demarc.com](http://www.demarc.com) below:

- Network Intrusion Detection System (NIDS) management console, integrating the raw power of the Open Source "Snort" IDS engine with the convenience and power of a centralized interface for all network sensors.
- Monitor all servers / hosts to make sure network services such as a mail or web servers remain accessible at all times.

<sup>11</sup> <http://www.snort.org/docs/lisapaper.txt>

<sup>12</sup> <http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html>



## Student Workbook – Module 13: Intrusion Detection Systems

- Monitor local processes on a host and optionally restart them if they terminate unexpectedly.
- Monitor system logs looking for anomalous log entries that may indicate intruders or system malfunctions.
- Distributed File Integrity Checking that allows not only for immediate discovery of files that have been tampered with but also an additional level of security over standard file integrity checkers because the "known good" data is stored in the central database away from the potential intruder of a specific host.
- Multilevel authentication to allow different users access to specific configuration options, or "monitoring only" accounts, which only allow viewing of the aggregate data.
- Advanced search and graphing capabilities that allow operators to easily extract useful information from the NIDS data, facilitating an efficient and effective investigation into possible intrusion attempts and trends.
- Complex alerting capabilities that allow for highly specific alert settings for all areas of the software, so that you only get paged for the events you chose.
- Integrates all necessary security software into an all-inclusive centralized management console.
- Client and Server are fully compatible with Windows NT/200/XP, Linux and virtually every version of Unix.



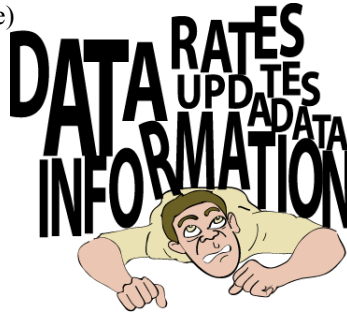
## Practical Problems -1

### Information about attacks

- New attacks will be developed, constant updates necessary
- Number and update period becomes logistical problem
- Quality of signatures provided (false positive / false negative rate)

### Ability to collect data at high speeds

- Bandwidth is increasing--hard for most network IDS to keep up with data rate



© 2002 Carnegie Mellon University

The rate at which new attacks are being developed is growing. As a result, rules must be developed to update IDS tools. This is the same scenario that anti-virus vendors and network administrators have been dealing with for years. It is a challenge to keep the systems up to date, and the administrators' knowledgeable enough to understand the alerts themselves.

Because of the multiple variables involved, i.e., different operating systems and applications etc, it is difficult to ensure that only applicable signatures are implemented and that those are effective.

Another problem is how IDS deal with the increasing bandwidth of modern networks. It is commonplace for 1 Gbps pipes to be used both for backbones and for links between infrastructure (and even down to the desktop). IDS must capture all packets and analyze them in order to be effective. Therefore, hardware and software must be architected to handle the increased load.<sup>13</sup>

<sup>13</sup> <http://rr.sans.org/intrusion/volume.php>



## Practical Problems -2

Number of protocols to support

- Observed protocols need to be fully understood

Only local traffic can be collected

- Larger number of IDS might be necessary
- Switched networks make more complicated
- Aggregation and analysis of combined traffic necessary

Because IDS must inspect all packets, it must understand numerous protocols. The signatures must be written for these as well. It is important in most implementations to block protocols and traffic that have no business on the protected network. By doing this, the IDS is freed up to a degree from having to deal with the numerous protocol formats and signatures.

Additionally, if there are numerous networks deemed to be critical enough to warrant IDS protection than there will be the added administrative, training, and resource burdens as well. The overall complexity of the IDS implementation may be difficult to “keep on top of” and may be less effective than a simpler implementation.

Collision domains have become more and more segmented through the wide-spread use of switching technology (see Network Infrastructure Module). This makes the deployment of NIDS more challenging because in default configurations, the NIDS will not be able to capture all of the traffic on the network. This may cause administrators to deploy multiple NIDS throughout their topology to provide the appropriate IDS coverage. To gain the big-picture perspective, all of the alerts from these NIDS should (as a best practice) be compiled onto a single system. This level of analysis can be a challenge for administrators, but will better enable them to holistically evaluate the security state of their networks



## Administrative Responsibilities

### Act on alerts

- Decide whether it is a real attack
- Document known false positives and inform other personnel

### Keep signature database up-to-date

### Make sure IDS components are not compromised

### Make sure critical assets are covered by IDS

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 41

Network administrators must be vigilant when managing IDS. They must have the training and experience to be able to make good decisions as a result of IDS alerts. When alerted to a possible attack, the administrator must determine whether the attack is real or not. If the IDS alert is determined to be a False Positive, the administrator should document this for future reference and inform others.

Much like anti-virus updates, signature updates must be performed constantly. The updates should not be applied blindly. For example, signatures for Novell IPX/SPX should not be applied to homogeneous TCP/IP networks.

Monitoring of IDS hosts should be conducted to determine if they are still operating normally. These systems are subject to compromise, and should be treated as any other critical network asset.

Lastly, only use IDS where needed—based on a comprehensive risk analysis of your network environment.

## Deploying IDS



- Step 1: Identify what needs IDS protection
- Step 2: Determine type(s) needed
- Step 3: Harden host system
- Step 4: Keep updated (like anti-virus updates)
- Step 5: Deploy IDS
- Step 6: Management and configuration

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 42

### Step 1 - Identify what needs to be protected

To maximize the utilization of IDS, the organization must first determine in order of priority what needs to be protected. For many organizations, the various servers, i.e., application, database, file and domain controllers, contain mission critical resources. Furthermore, depending on the organization, some departments may be more critical than others or must enforce different trust relationships. All of this must be defined in a priority list prior to deploying any IDS.

### Step 2 - Determine what types of sensors are required

The types of sensors that are required are dependant on the priority list defined in Step 1. A host sensor would be used to monitor a critical server, whereas a network sensor would be used to monitor network entry points and critical network segments. Another important issue to consider is how many sensors the organization can afford to buy. This number will influence how the sensors are deployed throughout the network, as the number of critical resources must be balanced against how many sensors can be acquired and maintained.

### Step 3 - Configure host system securely

Prior to loading any IDS, the host that the IDS will reside on must be configured securely. Often, the vendor of the IDS will supply its own host to run the IDS sensor, in which case, the vendor should supply guidelines on how to secure that host. Otherwise, the IDS typically reside on Unix and Microsoft Windows NT/2000 hosts. The guidelines for securing Unix and Microsoft Windows NT/2000 systems are documented in the Host System Hardening module.

### Step 4 - Keep signature database current

The majority of IDS that are currently available for use are signature based. Because new vulnerabilities and attacks are being discovered daily, the signature database must be kept current. The respective vendors should supply the latest signatures for their IDS.

### Step 5 - Deploy IDS sensors

The final phase is to actually deploy the IDS. The following scenarios (next 3 slides) are based on how many sensors are available for deployment versus what is deemed critical.

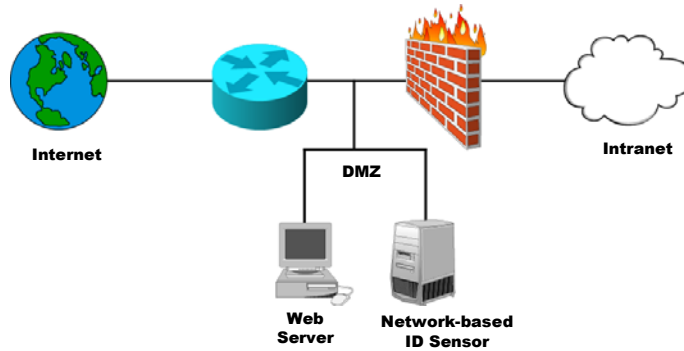
## Student Workbook – Module 13: Intrusion Detection Systems

### Step 6 - Management and Configuration

The other component of IDS, the management console, should be centrally located where dedicated security staff can monitor the health of the systems and network. Many organizations have a Network Operations Centers (NOC) that fulfills the role of a central location to place the manager. IDS sensors could then report all alerts to the NOC, thereby allowing the security staff to respond quickly to attacks and to notify the appropriate authorities, such as CERT technicians. The other issue to consider is how to configure the sensors. Careful configuration of the sensors can increase the effectiveness of IDS and all unnecessary signatures should be disabled. For example, if the network is entirely composed of Microsoft Windows NT systems, then the sensors can be configured to ignore any attacks that are directed against Unix systems. Therefore, if the organization has a priority list as defined in Step 1, as well as knowing the network intimately, it can benefit greatly from having a properly configured IDS [NSA 2001].

## IDS Deployment Scenario -1

### External network IDS

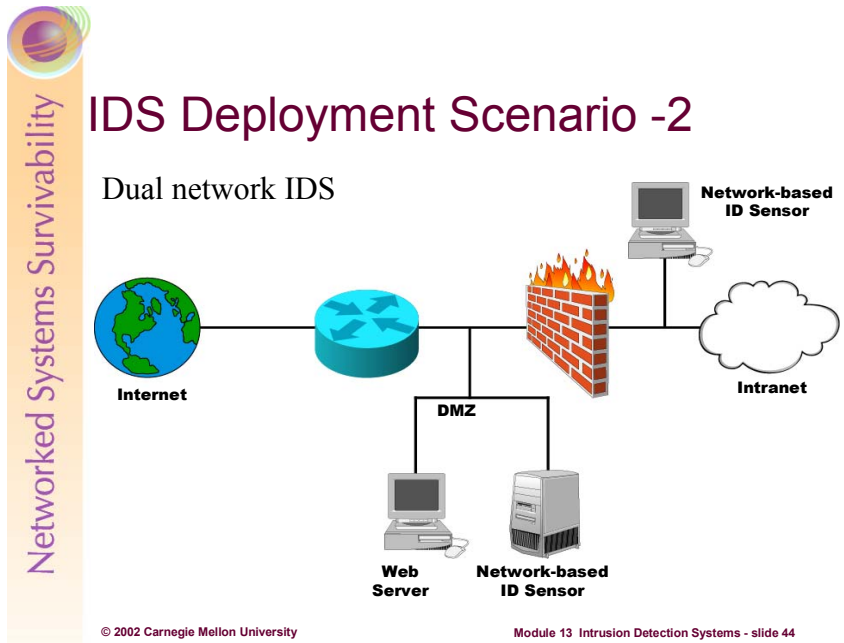


© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 43

### Scenario 1 External Network IDS only:

If the organization can only afford to purchase and monitor one sensor of any type, then it should be a network sensor. As described earlier, a network sensor is much better suited to monitoring large segments of a network, whereas a host sensor is limited to monitoring the system that it resides on. In this scenario, the ideal location to place the sole network sensor is in the DMZ, between the external router and the firewall, as shown above. In spite of having only one sensor, this design allows the IDS to be used for maximum effectiveness. By placing the IDS sensor between the external router and the firewall, the sensor can monitor all network traffic going to and coming from the Internet. Furthermore, because the router can filter all incoming traffic from the Internet, the IDS sensor can be tuned to ignore certain types of attacks, thereby allowing the sensor to operate with maximum efficiency [NSA 2001].



### Scenario 2, Dual Network IDS

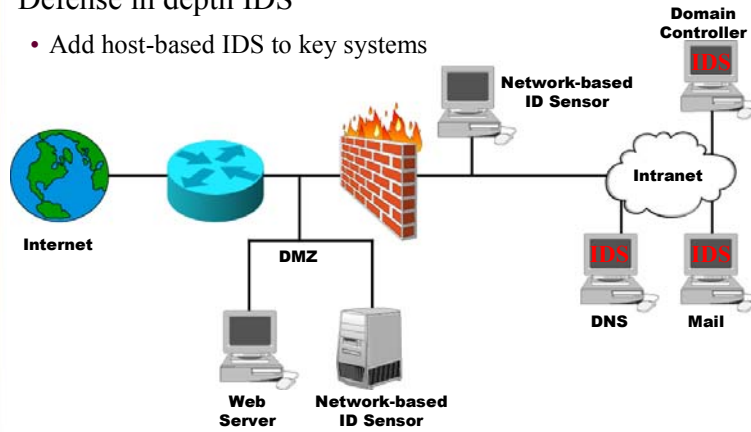
In the case where only two sensors of any type can be acquired and maintained, then they should be network sensors. Like the previous scenario, one of the sensors should be placed in the DMZ, between the external router and the firewall. The second sensor should then be placed between firewall and the intranet, as shown above. The second sensor can indicate what attack breached the firewall. By strategic placement of these two sensors, all access points from the Internet will be monitored [NSA 2001].



## IDS Deployment Scenario -3

### Defense in depth IDS

- Add host-based IDS to key systems



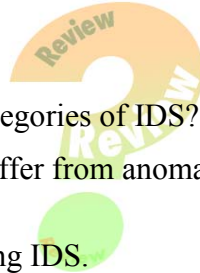
© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 45

### Scenario 3, Defense in Depth IDS

If more than two sensors of any type can be acquired and maintained, then at least two should be network sensors. Those sensors should be deployed as described in Scenario 2. If a critical LAN within the intranet needs to be protected, then a network sensor should be placed at the entry point to that LAN. The remaining sensors should be host sensors that are loaded onto critical servers, such as domain controllers, file servers, web servers, and mail servers. The order of what is deemed critical is determined by the organization, as directed in Step 1 [NSA 2001].

## Review Questions



1. What are the four general categories of IDS?
2. How does signature-based differ from anomaly-based intrusion detection?
3. List two techniques of evading IDS.
4. Describe one countermeasure for IDS evasion.
5. Identify two responsibilities administrators have in regards to IDS.
6. If an organization can afford only one IDS sensor, where might it be placed in the topology?

© 2002 Carnegie Mellon University

Module 13 Intrusion Detection Systems - slide 46

1. What are the four general categories of IDS?

Answer: Host-Based IDS, Network-Based IDS, Signature Based-IDS, and Anomaly-Based IDS

2. How does signature-based differ from anomaly-based intrusion detection?

Answer: Signature-based IDS can only provide alerts for intrusions that are deemed to matches for exact signatures already known to the IDS. Anything outside of that knowledge base will not be identified. Anomaly-based IDS looks for behaviors that are outside of a predefined baseline for normal operations.

3. List two techniques of evading IDS.

Answer: Fragmentation, Flooding, Staying below IDS thresholds, etc.

4. Describe one countermeasure for IDS evasion.

Answer: Maximize resources for IDS platforms, i.e., over-allocate memory, processing power, storage, etc.

5. Identify two responsibilities administrators have in regards to IDS.

Answer: Updating IDS Signatures, Documenting False Positives, make sure critical assets are IDS protected, etc.

6. If an organization can afford only one IDS sensor, where might it be placed in the topology?

Answer: Between the border router and the perimeter firewall—in the DMZ.



## Summary

- Evolution of IDS
- Host based IDS
- Network IDS
- Signature-based IDS
- Anomaly-based IDS
- Evasion and countermeasures
- Implementing IDS

**References:**

[Allen 2000] Allen, Julia et al., Networked Systems Survivability Program, Software Engineering Institute, Carnegie Mellon University. *State of the Practice of Intrusion Detection Technologies Technical Report*, January 2000. Available at: <http://www.cert.org/archive/pdf/99tr028.pdf>

[NSA 2001] National Security Agency; *60 Minute Network Security Guide*, Version: 1.0 October 16, 2001. Available at: <http://nsa1.www.conxion.com/support/guides/sd-7.pdf>

**Recommended further reading:**

Anderson, James P., *Computer Security Threat Monitoring and Surveillance*. James P. Anderson Co., Fort Washington, Pa., 1980.

Denning, Dorothy E., An Intrusion Detection Model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, February 1987.

Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., and Stoner, E., *State of the Practice of Intrusion Detection Technologies*. CMU/SEI-99-TR-028, Carnegie Mellon University, Software Engineering Institute, January 2000. Available at [www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html](http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html)

Egan, James P., *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

Larson, Amy K., Global Security Survey: Virus Attack. *Information Week*, July 12, 1999.

Briney, Andy, Got Security? *Information Security Magazine*, July 1999.

Northcutt, Steven, *Network Intrusion Detection*. New Riders, Indianapolis, Ind., 1999.

Amoroso, Edward and Kwapniewski, Richard, A Selection Criteria for Intrusion Detection Systems. *Proceedings of the 14th Annual Computer Security Applications Conference*. IEEE Computer Society Press, December 1998.

Ptacek, Thomas H. and Newsham, Timothy N., *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*, 1998. Available from [www.snort.org/IDSpaper.pdf](http://www.snort.org/IDSpaper.pdf)