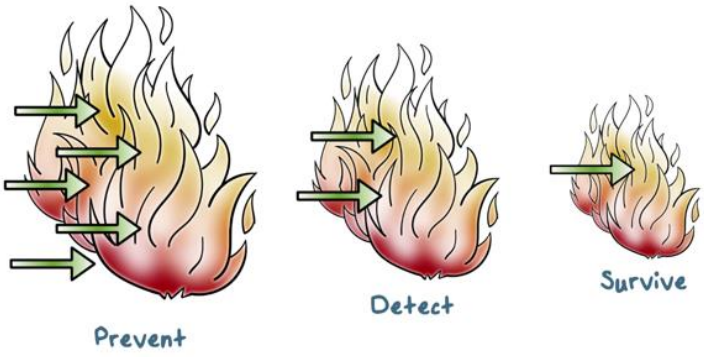
Reference: Computer Security by Stallings and Brown, Chapter 8

Intrusion Detection
Lesson Introduction

- Part of network defense-in-depth
- System architecture, algorithms, and deployment strategies of Intrusion detection
- Performance metrics
- Attacks on intrusion detection systems

Intrusion detection is also a part of the network defense-in-depth mechanism. In this lesson, we're going to discuss the system architecture, algorithms, deployment strategies, as well as performance metrics of intrusion detection systems. We're also going to discuss attacks on intrusion detection systems.

Defense-in-Depth



We call that defense in depth principle. We need multiple layers of defense mechanisms. That is, we need detection mechanisms even after we have deployed prevention mechanisms to detect attacks that can be easily prevented. That is, there are always some attacks that we can not simply keep out of our networks and systems. We have discussed firewalls, which are prevention mechanisms. Today we will discuss detection mechanisms. Typically, these mechanisms are the intrusion detection systems.

Intrusion Examples

- Remote root compromise
- Web server defacement
- Guessing/cracking passwords
- Copying databases containing credit card numbers
- Viewing sensitive data without authorization
- Running a packet sniffer
- Distributing pirated software
- Using an unsecured modem to access internal network
- Impersonating an executive to get information
- Using an unattended workstation

By intrusion, we mean any attack that aims to compromise the security ghost of an organization. For example, performing a remote root compromise, of an email server. Defacing a web server to display inappropriate content. Guessing and cracking stolen passwords. Stealing a database containing credit card numbers. Reading sensitive data without authorization. Running a packet sniffer on a workstation, to capture user names and passwords on a network. Using a permission error on an anonymous FTP server, to distribute pirated software and music files. Dialing into an unsecured modem, and gaining internal network access. Posing as a company executive, calling the help desk and resetting the executives email account. Using a workstation without permission.

Intrusion Detection Quiz

Select the characteristic that best describes each network security system.

Type (F) for Firewalls or (I) for IDS:

- ☐ tries to stop intrusion from happening
- ☐ tries to evaluate an intrusion after it has happened
- ☐ watches for intrusions that start within the system
- ☐ limits access between networks to prevent intrusion

Select whether it is firewalls or intrusion detection system that best describes each of these systems.

Instructor Notes:

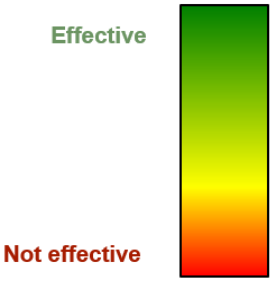
- Network Intrusion Detection Systems

- ☐ F tries to stop intrusion from happening
- ☐ I tries to evaluate an intrusion after it has happened
- ☐ I watches for intrusions that start within the system
- ☐ F limits access between networks to prevent intrusion

The first one tries to stop intrusion from happening. In other words, try to prevent attack from happening. This is obviously a firewall. The second one tries to evaluate an intrusion after it has happened. In other words, try to detect whether there's an intrusion happening. This is obviously an intrusion detection system. The third one, watches for intrusions that start within the system. Again, this means detection, so it is intrusion detection system. The fourth one, limits access between networks to prevent intrusion. Again, this tries to prevent attacks, so this is firewall.

Intrusion Detection Systems (IDS)

- Designed to Counter Threats:



- Known, less sophisticated attacks
- Sophisticated targeted attacks
- New, Zero-day exploits

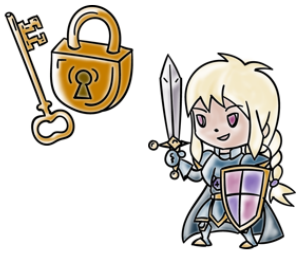
Intrusion Detection Systems, or IDS for short, are designed to counter these types of threats. They can be quite effective against known or less sophisticated attacks, such as large scale email phishing attacks.

However, as attack techniques become more sophisticated, IDS will become less effective. For example, attackers can blend attack traffic with normal activities so that it is very hard to detect such attacks.

And the most sophisticated attackers, in particular, the state-sponsored attackers, may use new or zero-day exploits to render IDS completely ineffective.

Intrusion Detection Systems (IDS)

Defense-In-Depth Strategies include:



- encryption
- detailed audit trails
- strong authentication and authorization controls
- active management of operating systems
- application security

Since IDS are not always effective, they need to be part of the defense-in-depth strategy for an organization. Typical defense-in-depth strategies should include encrypting sensitive information, provide detail of the trails of activities on systems and networks. Use strong authentication and access control. Actively manage the security of operating system and applications.

Intruder Behavior

The techniques and behavior patterns of intruders are constantly shifting in order to exploit newly discovered weaknesses, and to evade detection and counter measures. However, intruders typically use steps from a common attack methodology. Here are the common attack steps:

1. The first step is target acquisition and information gathering. That is, the attacker identifies the target systems using publicly available information, both technical and non-technical and use network tools to analyze target resources. For example, the attackers can find out what network services are accessible from the Internet. And the attacker may find out some email accounts of high level company executives.
2. The second step is initial access. This is accomplished by exploiting a remote network vulnerability. For example, a network service has a buffer overflow vulnerability that allows a remote attack to take over the service and gain access to a system. Or this can be accomplished by social engineering, whereby, for example, an attacker can send an email to a company executive with an attachment so that when the attachment is clicked, a malware is installed on the system.
3. The third step is privilege escalation. This is taken after the initial access. And the attacker would try to use local exploit to escalate the privilege from, for example, the normal user to root privilege on the target system.
4. The fourth step is information gathering or system exploit. That is, after an attacker has gained sufficient privilege on a system, he can then find out more about the network and the organization. And even move to another target system to further the exploit on the network.
5. The fifth step is maintaining access. This is important because an attack may not be a one-time action. That is, the attacker may choose to come back from time to time, or continue the exploit for a while. Therefore, the attacker may install backdoors or other malicious software on a target system so that he can continue to access this target system.

6. Lastly, the attacked wants to cover his tracks. For example, the attacker can disable or even edit the system audit logs to remove evidence of attack activities. Or the attacker may install root kits to hide the installed malware.

Intruder Quiz

Type True (T) or False (F) for each statement:

Decide whether each statement is true or false.

- ☐ An intruder can also be referred to as a hacker or cracker.
- ☐ Activists are either individuals or members of an organized crime group with a goal of financial reward.
- ☐ Running a packet sniffer on a workstation to capture usernames and passwords is an example of intrusion.
- ☐ Those who hack into computers do so for the thrill of it or for status.
- ☐ Intruders typically use steps from a common attack methodology.

- [T] An intruder can also be referred to as a hacker or cracker.
- [F] Activists are either individuals or members of an organized crime group with a goal of financial reward.
- [T] Running a packet sniffer on a workstation to capture usernames and passwords is an example of intrusion.
- [F] Those who hack into computers do so for the thrill of it or for status.
- [T] Intruders typically use steps from a common attack methodology.

The first statement, an intruder can also be referred to as a hacker or cracker. This is true. For example, we sometimes use hacker to refer to an intruder. Second, activists are either individuals or members of an organized crime group with a goal of financial reward. This is false. Instead of financial motives, activists typically have a social or political cause. Third, running a packet sniffer on a workstation to capture usernames and passwords is an example of intrusion. This is typically true, unless such packet sniffing is done with proper authorization. Fourth, those who hack into computers do so for the thrill of it or for status. This is false because it only describes some attackers. But there are many attackers who attack computers for other reasons, for example, for illicit financial gains. The last statement, intruders typically use steps from a common attack methodology. This is true.

Types of Backdoors Quiz

Choose the description that best fits each type of backdoor:

Choose the description that best describes the type of backdoors.

- ☐ Compiler Backdoors
- ☐ Object Code Backdoors
- ☐ Asymmetric Backdoors

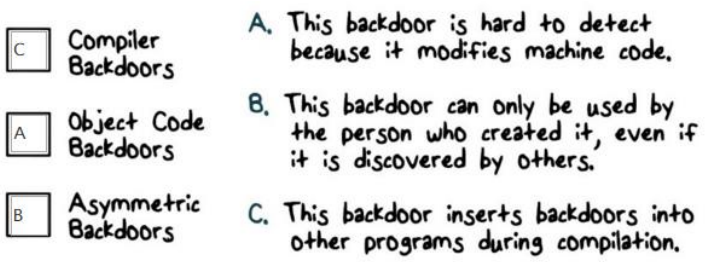
A. This backdoor is hard to detect because it modifies machine code.

B. This backdoor can only be used by the person who created it, even if it is discovered by others.

C. This backdoor inserts backdoors into other programs during compilation.

Instructor Notes:

- Biggest Baddest Boldest Software Backdoors of All Time
- Proprietary Backdoors
- Intrusion Detection System



First, compiler backdoors. Is it A, a backdoor that is hard to detect because it modifies machine code? Or B, this backdoor can only be used by the person who created it, even if it is discovered by others? C, this backdoor inserts backdoors into other programs during compilation. So it's obviously C. Second, object code backdoors. These are backdoors that are inserted into machine code. So that's A. Asymmetric backdoors. This backdoor can only be used by its creator, because it is protected or controlled by codographic schemes.

Elements of Intrusion Detection

- **Primary assumptions:**
  - System activities are observable
  - Normal and intrusive activities have distinct evidence

Now, let's discuss some details of intrusion detection.

First, what are the key design elements of an intrusion detection system? First, for intrusion detection to even be possible, we need to make some important assumptions. First, we can observe system and network and user activities.

Second, from the activities that we can observe, normal and intrusion actions have distinct evidence, that is, for intrusion detection to even be possible, we must be able to find evidence of intrusions by observing systems, networks, and user activities.

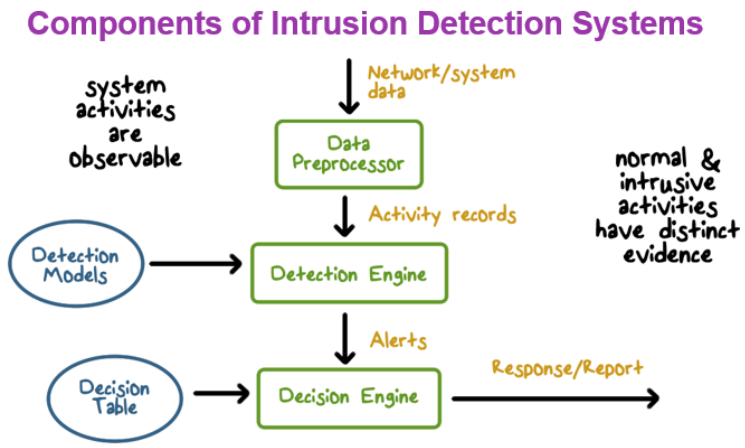
Elements of Intrusion Detection

- **Components of intrusion detection systems:**
  - From an algorithmic perspective:
    - **Features** - capture intrusion evidences
    - **Models** - piece evidences together
  - From a system architecture perspective:
    - Audit data processor, knowledge base, decision engine, alarm generation and responses

When it comes to designing an intrusion detection system, we must consider the following.

From the point of view of detection algorithm, we must consider the features, meaning how do we capture and represent intrusion or normal activity evidence? We must also decide the detection models. Meaning, how do we piece the evidence together, so that we can decide whether an activity is normal or intrusive.

The system architecture typically includes several modules including audit data processor, a knowledge base, a detection and decision engine, and alarm generation and response mechanisms.



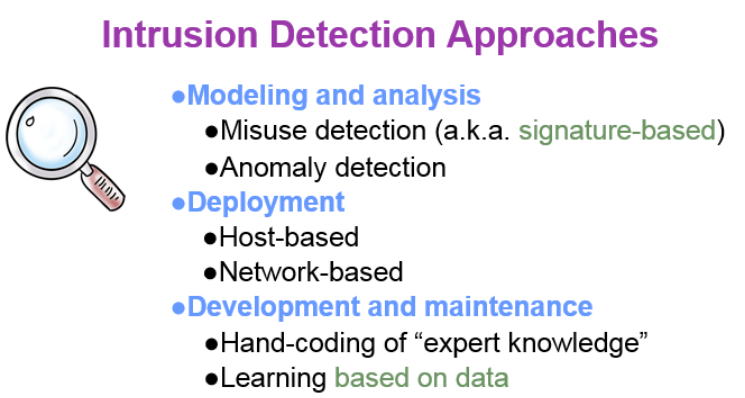
Here's an illustration of the workflow of intrusion detection as well as the main components of an intrusion detection system.

The input on an IDS is data that describes activities on systems and network. The data is processed by the data processor to extract activity records that are important for security analysis.

These activity data needs to be analyzed by the detection engine. The detection engine uses detection models that have been already constructed for the ideas that is these models are stored with ideas.

If a detection rule determines that there is an intrusion. The IDS produces an alert. The decision engine then decides the appropriate action according to decision table. For example, this can be a response that automatically blocks a network connection or report that is sent to the security admin.

Again, for the IDS to work properly, we'll assume that system activities are observable and are captured in the input data to the ideas. And when detection models are applied to the activity data, normal and intrusive activities have distinct evidence.



There are several ways to look at different intrusion detection approaches.

From the point of view of how data is analyzed, or how intrusion is detected, we have two approaches. One is misuse detection, also known as signature-based detection. The other one is anomaly detection. Misuse detection models all represents long intrusions. Anomaly detection models all represents normal activities.

From the point of view of deployment strategy, we can deploy an IDS on end host, or a network perimeter.

In terms of how the detection models are viewed and maintained, we can use manual encoding of expert knowledge. All hand written rules, or we can apply machine running algorithms to data to automatically learn the detection rules.

Analysis Approaches

•Anomaly Detection

• Misuse/ Signature Detection

An IDS typically uses these approaches to analyze data and detect intrusions.

Anomaly detection, by definition, it tries to detect what is not normal. Misuse or signature detection, you try to find a match of no intrusions.

Analysis Approaches

Anomaly Detection:

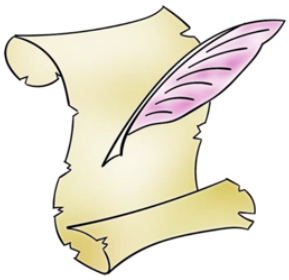
- Involves the collection of data relating to the behavior of legitimate users over a period of time
- Current observed behavior is analyzed to determine whether this behavior is that of a legitimate user or that of an intruder

There are mainly two elements in anomaly detection.

The first is training or profiling. The goal is to define or characterize the normal or expected behavior. This is accomplished by collecting data linking to normal activities, and applying data analysis algorithm to construct a model.

The second, is to evaluate whether an observed behavior fits the normal profile. This is accomplished by analyzing the observed data to see whether it fits the established model.

Analysis Approaches



Misuse/ Signature Detection

- Uses a set of **known malicious data** patterns or attack rules that are **compared with current behavior**
- Also known as **misuse detection**
- **Can only identify known attacks** for which it has patterns or rules

Misuse or signature detection, involves first encoding normal text into patterns or rules, and then comparing the current behavior with these rules or patterns. To see whether there's a match or not. Obviously, this approach can only detect known intrusions or known attacks.

Anomaly Detection Quiz

Check all answers that are true regarding Anomaly detection systems:

- [ ] The longer the system is in use, the more it learns about network activity.
- [ ] If malicious activity looks like normal traffic to the system, it will not detect an attack.
- [ ] False positives can become a problem, normal usage can be mistaken for an attack.

Check all statements that are true regarding anomaly detection systems.

First. The longer the system is in use, the more it learns about network activity. This statement's true, because anomaly detection involves first learning or profiling what is normal. The longer the system is in use, the better it can learn what is normal.

- [x] The longer the system is in use, the more it learns about network activity.
- [x] If malicious activity looks like normal traffic to the system, it will not detect an attack.
- [x] False positives can become a problem, normal usage can be mistaken for an attack.

Second, if malicious activity looks like normal traffic to the system, it will not detect an attack. This is true, because anomaly detection, detects what looks not like normal. Therefore, if an attack managed to look like normal, then the anomaly detection system will not be able to detect this attack.

The third statement, false positive can become a problem, normal usage can be mistaken for an attack. This is true. Because the definition of false positive is that, a normal activity is mistaken as an attack. At the minimum, false positives can waste systems time, because the system needs to investigate whether there's truly an intrusion or not.

Signature Detection Quiz

Check all answers that are true regarding Signature Based detection:

- [ ] New threats can be detected immediately.
- [ ] When a new virus is identified, it must be added to the signature databases
- [ ] Can only detect an intrusion attempt if it matches a pattern that is in the database

Check all statements that are true regarding misuse detection systems.

Instructor Notes:

- New Dark-Web Market is Selling Zero-Day Exploits to Hackers
- Signature Based and Anomaly Based Intrusion Detection Systems

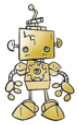
- [ ] New threats can be detected immediately.
- [x] When a new virus is identified, it must be added to the signature databases
- [x] Can only detect an intrusion attempt if it matches a pattern that is in the database

The first statement, new threats can be detected immediately, this is false because a misuse detection of signature-based detection system can only detect attacks that match patterns or rules of known intrusions. Second, when a new virus is identified, it must be added to the signature database. This is true. Because a misused detection system detects attacks based on signatures of known intrusions, therefore when a new attack is discovered, its signature needs to be added to the signature database. Third, can only detect an intrusion attempt if it matches a pattern that is in the database, this is true. This is essentially the definition of a signature-based detection system.

A Variety of Classification Approaches

Statistical: Analysis of the observed behavior using univariate, multivariate, or time-series models of observed metrics.

Knowledge Based: Approaches use an expert system that classifies observed behavior according to a set of rules that model legitimate behavior.



Machine Learning: Approaches automatically determine a suitable classification model from the training data using data mining techniques.

The anomaly detection approach involves first developing a model of normal or legitimate behaviors by collecting data from the normal operations of the monitored system and network. This is called a training phase.

Once this model exists, the observed behavior is compared with this model in order to classify it as either legitimate or anomalous and this is the detection phase.

A variety of approaches are used to construct such depression model. The statistical approach, process data into metrics or measures and then apply univariate, multivariate or time series analysis to view a model.

The knowledge based approach uses expert knowledge to encode legitimate behavior into a set of rules as the detection model.

A machine learning approach uses data mining and machine learning techniques to automatically learn a model from the training data.

A Variety of Classification Approaches

Issues Affecting Performance:

- **Efficiency**

- **Cost of Detection**

When we compare these approaches, we need to consider both efficiency and cost.

Efficiency here means how fast we can learning a model from training data, and how fast we can apply the model to the observed data to determine whether it is anomalous or normal.

The cost of detection here means, how much data do we need in order to view

the model and apply the model, and how much completion of power it is required? For example, typically a machine learning based approach required a lot of data.

Anomaly Quiz

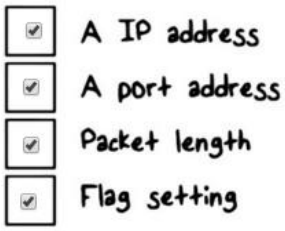
Which of the following **could be considered an anomaly** to typical network traffic?

- ☐ A IP address
- ☐ A port address
- ☐ Packet length
- ☐ Flag setting

Let's make sure we understand what we are looking for when we talk about anomalies with regards to intrusion detection. Which of the following is considered anomaly in a typical network?

Instructor Notes:

- Anomaly Detection
- Malicious Traffic Present on 100% of Sampled Networks
- Nations Buy as Hackers Sell



First, an IP address. Can this be an anomaly? If the IP address is not the one that normally accessed by users or is not well known, it can be anomaly. So this is anomaly. Second, a port address. Similar to the IP address, if the port address is not normally accessed, then this is an anomaly. How about packet length? Again, if the length is unusually long, for example, then this is an anomaly. How about flag setting on a packet? Again, if these flags are not normally seen under the same traffic conditions, then this is an anomaly. That is, all of these can be anomalies if they are not normally seen in normal operations of the network.

Statistical Approaches

Characteristics:

- Use captured sensor data
- Multivariate models using time of and order of the event

Advantages:

- their relative simplicity
- low computation cost
- lack of assumptions about expected behavior

Disadvantages:

- difficulty selecting suitable metrics
- not all behaviors can be modeled using these approaches.

Let's discuss the statistical approaches in more details.

Statistical approaches use the captured sensor data as training data, to build up a model of normal behavior. The earliest approaches used univariate models, where each metric, or measure, such as a CPU utilization of a program, was treated as an independent, random variable. However, this was to crude to effectively detect intrusions. The later approaches use multivariate models to consider the correlations between the measures. For example, CPU utilization and memory size of the program are considered together. Some later approaches also use time series models to consider the order and time we can observe events. For example, user logging into a workstation and an email from the workstation are two events that consider in a time order.

The main advantages of statistical approaches are that they are relatively simple, easy to compute and they don't have to make a lot of assumptions about the behavior. These approaches don't rely on assumptions about the expected or normal behaviors, and they're simple and efficient.

On the other hand, the effectiveness of these approaches rely on selecting the right set of measures, which is a very difficult task. In addition, they have complex behaviors that are beyond the capabilities of these models.

Knowledge Based Approaches

- Developed during training to **characterize data into distinct classes**

Advantages:
- Robust
- Flexible

Disadvantages:
- The difficulty and time required to develop knowledge from the data
- Human experts must assist with the process

Knowledge based approaches rely on experts to develop a set of rules that describe the normal and legitimate behaviors observed during training.

For example, these rules can classify activities into different classes.

For example, a rule can say that a secretary typically only uses office productivity programs such as web browser, email, calendar, and Microsoft Office.

These rules are quite robust and flexible because it's relatively easy to update and improve them. On the other hand, these approaches rely on manual efforts by the experts, and the experts must have very good knowledge of the data.

Statistical & Knowledge Based Approaches Quiz

Which of these characteristics describes the **statistical approach** and which describe a **knowledge based** approach? **Write S or K in the box:**

☐ Any action that does not fit the normal behavior profile is considered an attack.

☐ Any action that is not classified as normal is considered to be an attack.

Which of the following describes the statistical approach, versus the knowledge based approach?

Instructor Notes:

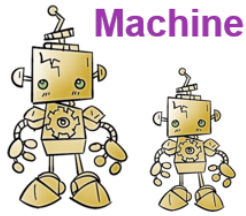
- Statistical Intrusion Based Detection
- Knowledge Based Intrustion Detection

☐ S Any action that does not fit the normal behavior profile is considered an attack.

☐ K Any action that is not classified as normal is considered to be an attack.

First, any action that does not fit the normal behavior profile is considered an attack. Since it talks about behavior profile, this is the statistical approach. Second, this means that any action that's not classified as one of the normal behaviors according to set of rules is considered to be an attack. So, this is the knowledge-based approach.

Machine Learning Approaches



- Use **data mining techniques** to develop a model that can classify data as normal or anomalous

Advantages:
- Flexibility
- Adaptability
- Ability to capture interdependencies between observed metrics

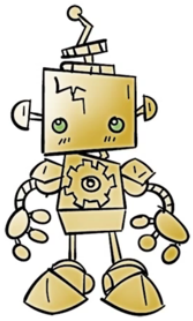
Disadvantages:
- Dependency on assumptions about accepted behavior
- High false alarm rate
- High resource cost
- Significant time and computational resources

Let's discuss the machine learning based approaches. Machine learning approaches can automatically build a model using the labeled normal training data. That is, a machine learning algorithm fixes input examples of normal data and outputs a model that is then able to classify subsequently observed data as either normal or anomalous.

Machine learning base models can automatically handle small changes or variations in the observed data. It can also capture the inter-dependencies or the deeper connections between the measures of the observed data.

On the other hand for these approaches to be effective the normal training data must be representative of normal behaviors otherwise when we apply the model to detection there could be a lot of false positives. In addition in the training phase it required a lot of data and a lot of computational power. However, once the model is produced subsequent analysis is typically, fairly efficient.

Machine Learning Intruder Detection Approaches



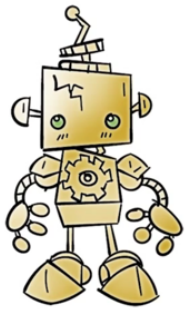
- Bayesian networks: Encode probabilistic relationships among observed metrics.
- Markov models: Develop a model with sets of states

A variety of machine learning algorithms can be used for intrusion detection.

These include Bayesian networks. Bayesian networks encode the conditional probabilities among observed events. For example, how likely an email is sent by a user if the current time is 2:00 AM. If a low probability activity takes place it is an anomaly.

Markov models, a Markov model is a set of states that are connected by transitional probabilities. For example, Markov models can be used to model legitimate website names because, the traditional probabilities from one letter to the next should be similar to that of real dictionary words because, users need to type the website names. On the other hand, a randomly spelled website name is anomalous and may be used by botnets for C and C, or command control, because bots don't have to type the website names.

Machine Learning Intruder Detection Approaches



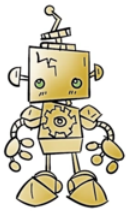
- Neural networks: Simulate human brain operation with neurons and synapse between them
- Clustering and outlier detection: Group the observed data into clusters then identify subsequent data as either belonging to a cluster or as an outlier.

Neural networks. A neural network simulates how human brain perform reasoning and learning. They are one of the most powerful machine learning approaches.

Clustering and outlier detection. Clustering groups the observed data into clusters based on some similarity or distance measure. And then identify subsequently observed data as either belonging to a cluster or as an outlier. For example, traffic from the internal network to the company's internal web server have common characteristics that can be grouped into clusters based on the webpages visited. On the other hand, an attack may access data on the web server that's rarely visited, which makes it an outlier.

Machine Learning Quiz

Which description best describes the Machine Learning **approach for Intruder Detection**:



- ☐ detects new and novel attacks
- ☐ detects attacks similar to past attacks

Let's take a moment to think about the machine learning base approaches to intrusion detection. Which of these statements best describes the machine learning base approach for intrusion detection? Is it, detecting new and novel attacks? Or, detecting attacks similar to past attacks?

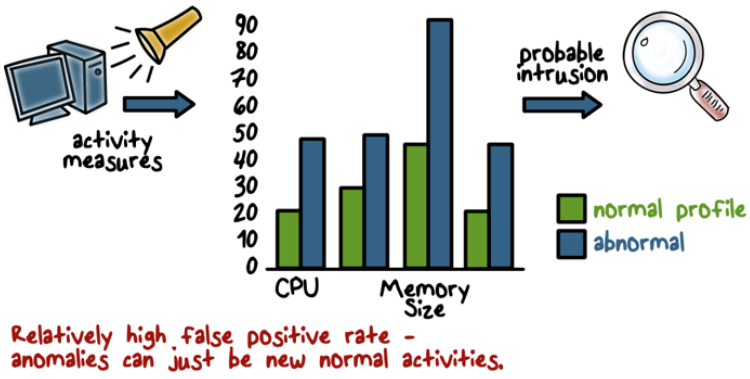
- ○ detects new and novel attacks
- ◉ detects attacks similar to past attacks

Limitations of Anomaly Detection

- They are generally trained on **legitimate data**
- This **limits the effectiveness** of some of the techniques discussed.

A key limitation of Anomaly Detection is that it only uses normal or legitimate data in training. That is, there's no intrusion data involved. As a result, when an Anomaly Detection model detects an anomaly, it may not be limited to an intrusion.

Anomaly Detection Example



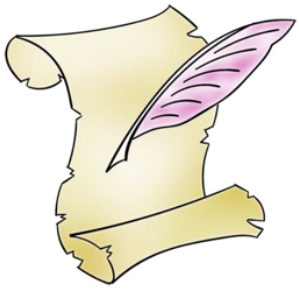
Here's an example of a very simple anomaly detection approach.

First, it establishes the normal statistical runtime profile of a program. For example, in terms of CPU utilization, memory size, etc. This can be accomplished by running the program many times. And at each time, record the values of these measurements and then compute the means and variance.

Once the profile or detection model is built, when the IDS observes that when the program is run, its measures deviate from the means beyond the allowed thresholds or the variance, meaning that the values are outside of their normal ranges. The ideas outputs an alert.

Again, the main drawback of anomaly detection approach is that you can produce relatively high false positive rate because an anomaly can just be a new or observed normal activity.

Misuse or Signature Detection



Detect intrusion by:

- observing events in the system
- applying a set of patterns or rules to the data
- determining if the is intrusive or normal

Now let's discuss misuse or signature detection.

These techniques detect intrusions by looking at activity data, seeing if there's a match of known intrusion patterns, and if there's a match this activity is intrusion. Otherwise it is normal.

Anomalous Behavior Quiz

One of the weaknesses of anomalous intruder detection is that a system must learn what is normal behavior. While it is learning this, the network is vulnerable to attack. What can be done to mitigate this weakness?

Write your answer in the textbox:

One of the issues with anomaly detection is that a system must learn what is “normal behavior”? And in the training phase, when it is learning what is “normal behavior”, the network is vulnerable to attack. Meaning that before the IDS is deployed, the network is vulnerable. What can we do about this?

uses a Firewall.

Signature Approaches

- **Match a large collection of known patterns** of malicious data against data stored on a system or in transit over a network
- The signatures need to be **large enough to minimize the false alarm rate**, while still detecting a sufficiently large fraction of malicious data
- **Widely used** in anti-virus products, network traffic scanning proxies, and in NIDS

Many misused detection approaches are signature based.

A signature is essentially a known pattern of malicious activity data. For example, this can be option settings of flags in a packet header, or it can be a particular string in the packet pay load. The set of signatures needs to be as large as possible in order to have a high detection rate and a low false-positive rate. Signature-based approaches are most widely used in anti-virus software and network intrusion detection systems.

Signature Approach Advantages & Disadvantages

Advantages:
- Low cost in time and resource use
- Wide Acceptance

Disadvantages:
- Significant effort to identify and review new malware to create signatures
- inability to detect zero-day attacks

Signatures are typically very easy to understand. And symmetry matching is very efficient. Therefore, symmetry based approaches are widely used. On the other hand, every time a new malware or a new attack method appears, significant menu effort must be spent in order to create new signatures. In addition, these approaches can not detect zero-day attacks, because these attacks are new and they do not have known signatures yet.

Zero Day Market Place Quiz

In the thriving zero day attack marketplace hackers sell information on software vulnerabilities. **Can you guess some of the buyers**?

- [ ] Apple
- [ ] Google
- [ ] Microsoft
- [ ] U.S. Government

Perhaps, you do not think that it is important to detect zero day attacks. There are buyers of zero day attack information, can you guess some of the buyers? Is Apple a buyer, Google a buyer, Microsoft, or the US Government?

Instructor Notes:

- **Zero Day Marketplace**

The answer is, that they're all buyers of zero day attack information. For example, a zero day vulnerability in the Linux operating system was sold for $50,000.

- [x] Apple
- [x] Google
- [x] Microsoft
- [x] U.S. Government

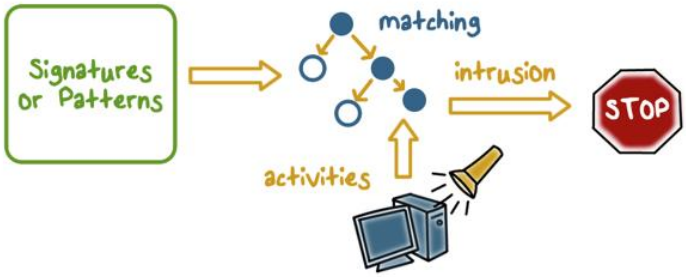
Rule-Based Detection

- Involves the **use of rules for identifying known penetrations** or penetrations that would exploit known weaknesses
- Rules can also be defined that **identify suspicious behavior**
- Typically rules used are **specific**
- **SNORT** is an example of a rule-based NIDS

In addition to signature based approaches, a misused detection system can also use a more sophisticated Rule-Based Approach.

A Rule-Based Approach uses rules to represent known intrusions. These rules, typically match multiple signatures or patterns in the activity data. These rules are not only specific to the known intrusions, they can also be specific to the target of protected network. Because the same intrusion may leave different evidence in a different network depending on the network set ups. SNORT is a widely known example of a Rule-Based Network Intrusion Detection System.

Misuse Signature Intruder Detection



Example: if (src_ip == dst_ip && src_prt == dst_prt) then "land attack"

Can't detect new attacks

Here's a simple example of misuse detection approach.

The IDS matches the observer activities using a set of text signatures or patterns. If there's a match, the IDS outputs an alert. For example, here is the signature of the so called land attack. That is, the source IP, it's the same as destination IP and the source port is the same as destination port. A machine that received this packet may crash. Again, such a simple approach cannot detect new attacks because they don't have signatures.

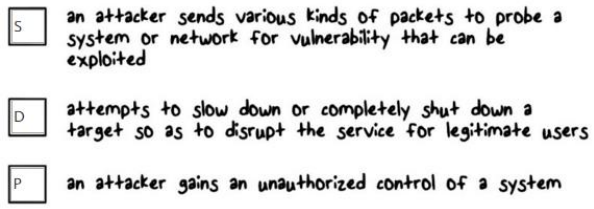
Attacks Quiz

Write the name of each attack next to it's definition. The choices are **Scanning Attack (S), DOS(D)**, and **Penetration Attack(P)**.

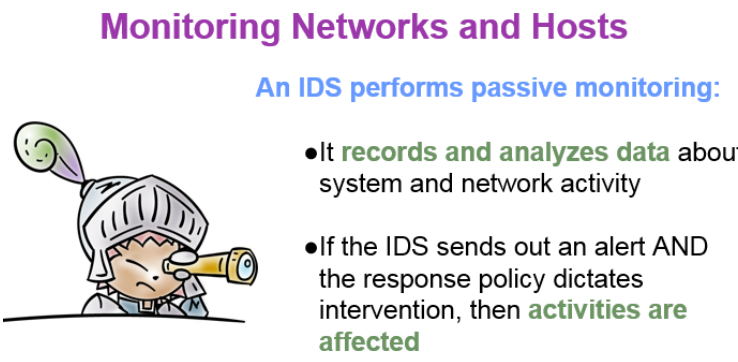
Let's check our understanding of intrusion detection. Here's a list of attacks that an intrusion detection system can detect. Match the definition of the attack with its name.

- ☐ an attacker sends various kinds of packets to probe a system or network for vulnerability that can be exploited
- ☐ attempts to slow down or completely shut down a target so as to disrupt the service for legitimate users
- ☐ an attacker gains an unauthorized control of a system

First, an attacker sends various kinds of packets to probe a system or network for vulnerability that can be exploited. This is scanning the network in order to find weaknesses for attacks. Second, attempts to slow down or completely shut down a target, so is to disrupt the service for legitimate users.

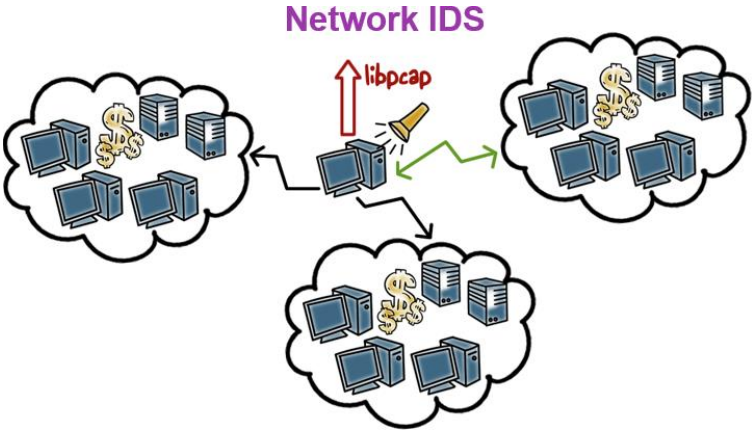


Disrupting the service is the same as Denial of Service. Third, an attacker gains an unauthorized control of a system. That is to say, the attacker has penetrated into the system.



Now let's discuss some details of the systems and deployment aspects of intrusion detection.

An IDS typically performs what we call passive monitoring. That is, it is recording and analyzing data about systems and network activities while these activities continue to take place. Only when the IDS outputs an alert and the response policy dictates a direct intervention, such as blocking a connection or terminating a program, are the activities affected.



An IDS can be deployed at the perimeter of a network, or subnet, to monitor traffic going in and out of the network.

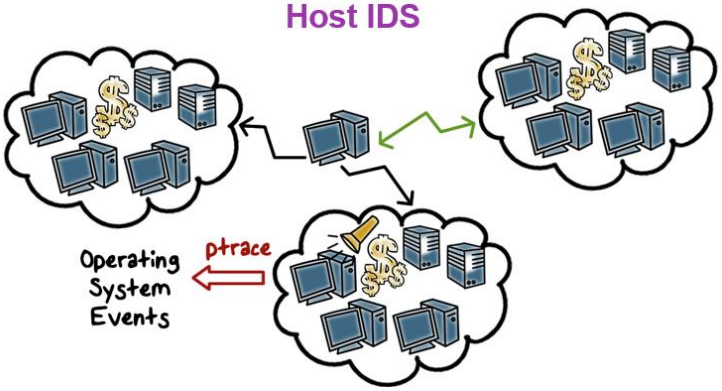
Such an IDS is a network IDS. It typically uses a packet capturing tool, such as the libpcap tool to obtain network traffic data. The packet data contains the complete information about network connections. For example, if a user connects to a website using his browser, the packet data will contain all the TCP handshake information between the browser and the web server, and all the URL requests from the browser and the return page contents from the server. That is, by examining the packet data, the IDS has all the data sent and received by the user's browser. If the user's machine is infected by a bot malware, for example, any attempts to connect to a website for command and control, the network IDS will be able to see that the traffic looks like CNC activities and output an alert.

Network Based IDS (NIDS)

- **Monitors traffic at selected points** on a network in real or close to real time
- May examine network, transport, and/or application-level protocol activity
- **Comprised of a number of sensors**, one or more servers for NIDS management functions, and one or more management consoles for the human interface
- **Analysis of traffic patterns** may be done at the sensor, the management server or a combination of the two

An IDS can be deployed to monitor a network or subnet. It monitors traffic in real or close to real time so that it can react to intrusions in a timely manner. It can analyze traffic in multiple layers of the network stack. A network IDS can include a number of sensors. For example, each of them is deployed to monitor a subnet. In such case, it may include one or more serves in the backend that manages these censors and interface with a human analyst. Typically, a censor monitors traffic of a subnet and reports its finding to the backend server. Then, the backend management server can correlate evidence across multiple sensors to provide protection or detection of the whole network.

Host IDS



An IDS can also be deployed in an end host. And such an IDS is a host-based IDS.

It can use a variety of data on system activities. For example, most host-paced IDS use ptrace to obtain the system calls made by the program to monitor the behaviors of the program.

System called data is very useful to security monitoring because whenever a program requests a resource such as memory allocation, access to the file system, networks, and IO devices, it needs to make a system call to the operating system because the operating system manages the system resources. That is, most of the interesting or useful activities by program are carried out through system calls. For example if the users browser receives a page with a malicious JavaScript that is able to break the protection in the browser and attempts to overwrite the Windows registry file, the IDS will observe a write system call to the registry file. And can decide that this is an anomaly.

NIDS QUIZ

Can you think of a way to reduce the impact of excessive reporting on a system's administrator?

Write your answer in the textbox:

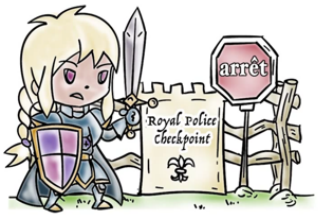
One issue, with network IDS, in particular, for large network, is that it may produce a large number of alerts of possible intrusions, and these alerts need to be examined by the systems administrator. How do we reduce the impact of excessive reporting on an administrator?

Instructor Notes - Anomaly Detection based NIDS

Prioritize the alerts by adding a security level that is associated with each report, so that the system admin can focus on the top priority alerts first.

Prioritize the alerts

Inline Sensors



- Used to block an attack when one is detected, **performing both intrusion detection and prevention functions**
- An inline sensor is inserted into a network segment so that the **traffic that it is monitoring must pass through the sensor**.

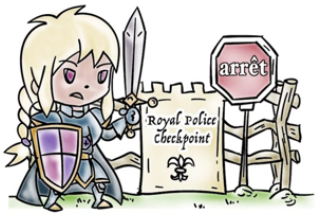
One type of network IDS configuration, is inline sensors.

The primary motivation for using inline sensors is to enable them to block an attack, when an attack is detected. That is, in such a case, an inline sensor performs both intrusion detection and intrusion prevention. And of course, for an inline sensor to be effective, it must be placed at a network point where traffic must pass through it.

We can deploy an inline sensor as a combination of network ideas and a firewall, in a single piece of hardware. Or, we can deploy the inline sensor as a stand-alone, inline network IDS.

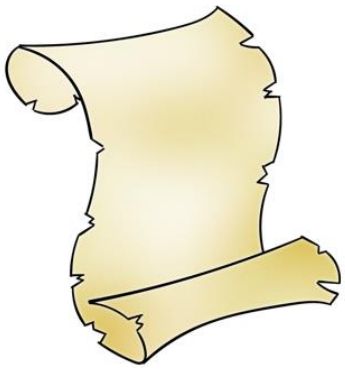
Inline Sensors

Can be achieved by:



- **Combining NIDS sensor logic with a firewall or LAN switch**. This has the advantage of no additional hardware is needed
- Using a **stand-alone inline NIDS sensor**

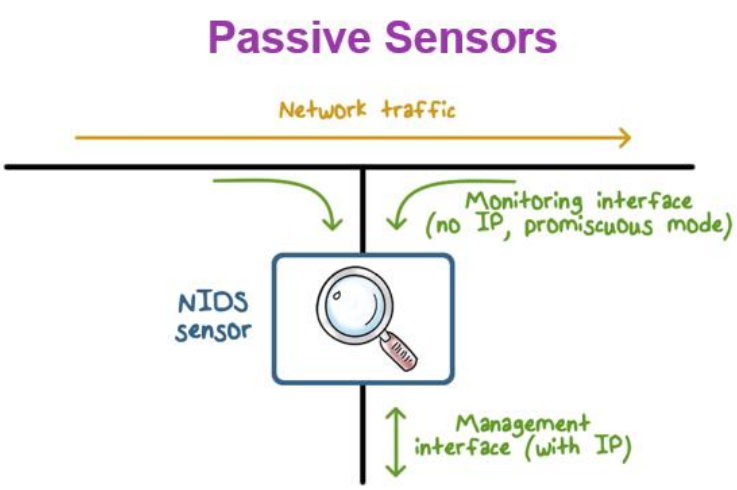
Passive Sensors



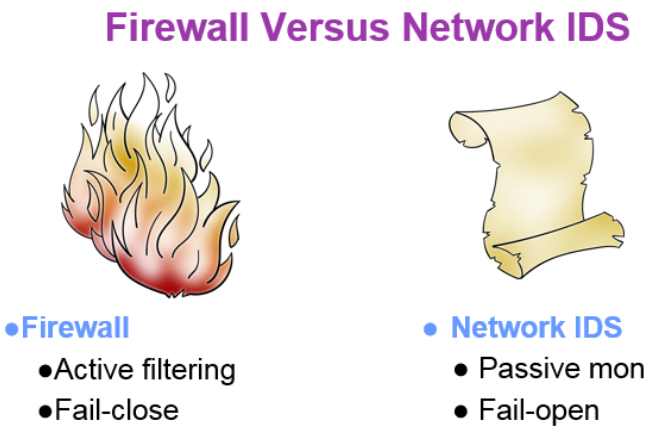
- A passive sensor **monitors a copy of network traffic**; the actual traffic does not pass through the device
- Passive sensors are more efficient

The more common deployment strategy for network IDS is to deploy them as passive sensors.

A passive sensor only takes a copy of the traffic. That is, the traffic continues to reach its destination without passing through the device. Therefore, from the point of view of network performance, a passive sensor does not add any overhead to network traffic.



This illustrates a typical passive sensor configuration. The sensor connects to the network transmission medium, such as an Ethernet cable, through a direct physical tap. The tap provides the sensor with a copy of all network traffic being carried by the medium. The network interface card for this tab usually does not have an IP address configured for it. Therefore, all traffic into this network interface card is simply collected with no protocol interaction with the network. The sensor has a second network interface card that connects to the network with an IP address so that it can communicate with a backhand management server.



Let's compare firewalls with IDS. A network IDS performs passive monitoring. That is, while the IDS is copying and analyzing the network traffic, the traffic is continuing to reach its destination. Traffic analysis can take a lot of computing power, and therefore the IDS can be overloaded by large body morph traffic. When an IDS is overloaded, it cannot detect intrusion in a timely manner. That is, it fails to adequately protect a network. We call this situation fail open, meaning that when the IDS fails, the network is open to intrusions. On the other hand, a firewall performs active filtering. That is, all traffic must pass through the firewall and the firewall performs relatively simpler and more efficient analysis. However, it can still be overloaded by large volume of traffic. When this happens, it will simply not let the traffic go through. We call this fail close, meaning that when a firewall fails, the internal network is closed to the external network, and it is safe.

IDS Quiz

Put a (T) for True next to each true statement and a (F) for False next to each false statement.

Now let's do a quiz on IDS. Decide whether each statement is true or false.

- ☐ Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified.
- ☐ The primary purpose of an IDS is to detect intrusions, log suspicious events, and send alerts.
- ☐ Signature-based approaches attempt to define normal, or expected, behavior, whereas anomaly approaches attempt to define proper behavior.
- ☐ An network IDS sensor monitors a copy of network traffic; the actual traffic does not pass through the device.
- ☐ Network-based intrusion detection makes use of signature detection and anomaly detection.

The first statement, intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified. This is true. This is the primary assumption of IDS.

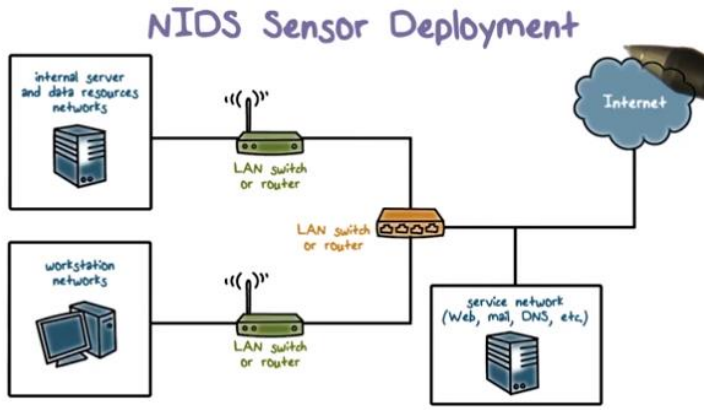
- [T] Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified.
- [T] The primary purpose of an IDS is to detect intrusions, log suspicious events, and send alerts.
- [F] Signature-based approaches attempt to define normal, or expected, behavior, whereas anomaly approaches attempt to define proper behavior.
- [T] An network IDS sensor monitors a copy of network traffic; the actual traffic does not pass through the device.
- [T] Network-based intrusion detection makes use of signature detection and anomaly detection.

The second statement, the primary purpose of an IDS is to detect intrusions, log suspicious events and send alerts. This is true statement, because these are the basic functions of an IDS.

The third statement, signature-based approaches attempt to define normal, or expected, behavior, whereas anomaly approaches attempt to define proper behavior. This is false, because a signature based approach is typically used to represent known intrusion patterns.

The fourth statement, a network IDS sensor monitors a copy of network traffic. The actual traffic does not pass through the device. This is true because a network ID typically performs passive monitoring by copying the network traffic.

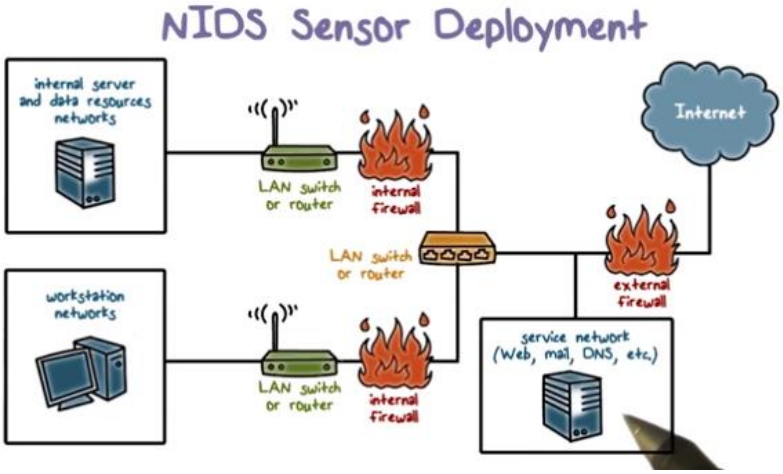
The last statement. Network-based intrusion detection makes use of signature detection and anomaly detection. This is true. You can indeed use both approaches.



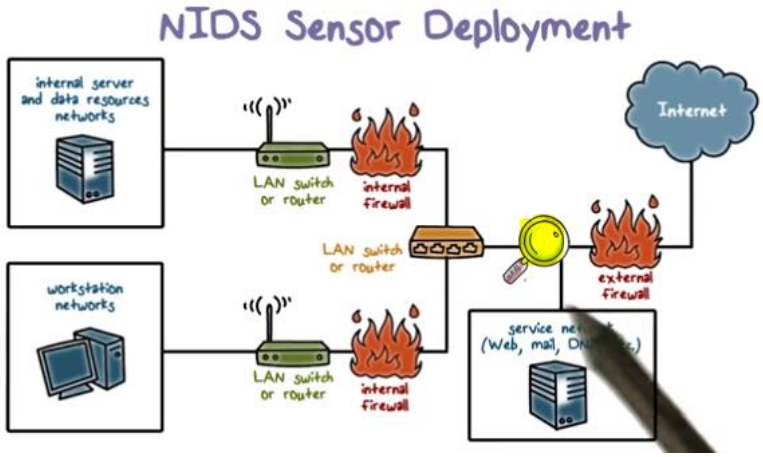
Let's discuss more about network IDS deployment.

Here's an example enterprise network configuration. The internal network has multiple subnets. And the enterprise has public-facing services, such as a public web server.

Recall our lecture on firewalls. We typically want to place an external firewall to protect the entire enterprise network. In addition, we want to protect the internal network from the public-facing servers. These servers are put in what we call a DMZ and we use internal firewalls to monitor traffic between the internal subnet and a DMZ. The internal firewalls also



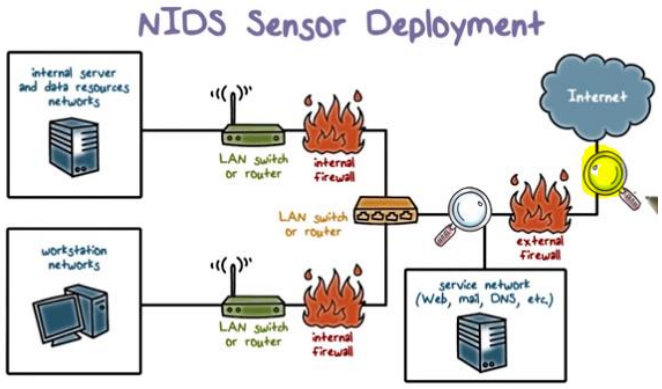
monitor traffic between its subnets.



So that's the deployment of firewalls, but what about IDS? A common location for an IDS sensor is just inside the external firewall.

This position has a number of advantages. Obviously, you can see attacks from the outside world. By comparing the logs of the firewall and the IDS, we can also find out whether the firewall had missed an attack that it should have prevented. The IDS at this location can also detect attacks that are targeted at the public facing servers. In addition, because it can analyze all outgoing traffic of the entire enterprise network, it can also detect attacks from a compromised server, either from DMZ or the internal network.

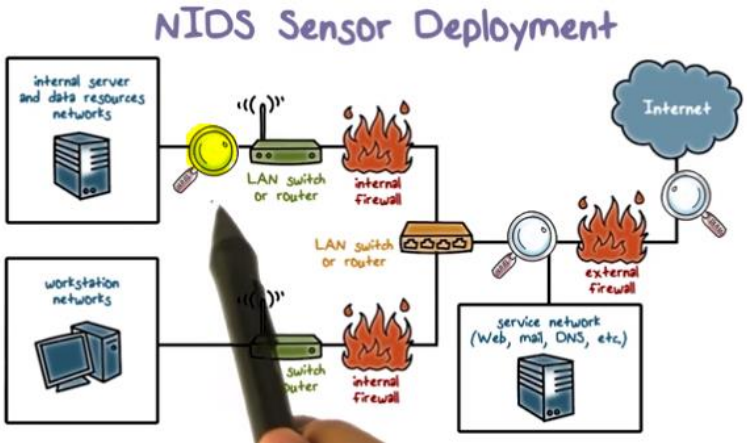
Advantages:
- Sees attacks, originating from the outside world
- Highlights problems with the network firewall policy or performance.
- Sees attacks that might target the Web server or ftp server.
- Even if the incoming attack is not recognized, the IDS can sometimes recognize the outgoing traffic that results from the compromised server.



A network IDS can also be placed between the external network and the Internet.

The main advantage of this location is that the network IDS can see all attempted attacks to the enterprise network, including those attacks that have been filtered by the firewalls. For example, if the firewall is overloaded, you would not only drop the incoming packet, it may not even have resource to log this packet. But an IDS at this location can see the packet and log it. Therefore, the IDS can see all attempted attacks.

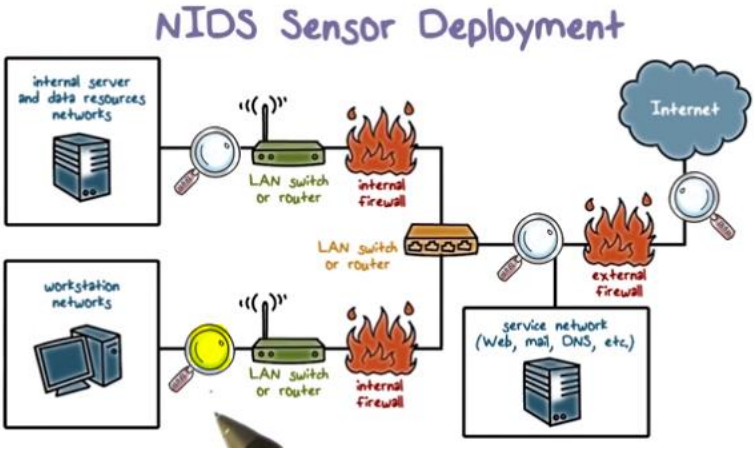
Advantages:
- Documents number of attacks originating on the Internet that target the network
- Documents types of attacks originating on the Internet that target the network



In addition to deploying network IDS at the perimeter, we can also deploy a network IDS to protect a subnet or set of servers.

Advantages:
- Monitors a large amount of a network's traffic, thus increasing the possibility of spotting attacks
- Detects unauthorized activity by authorized users within the organization's security perimeter

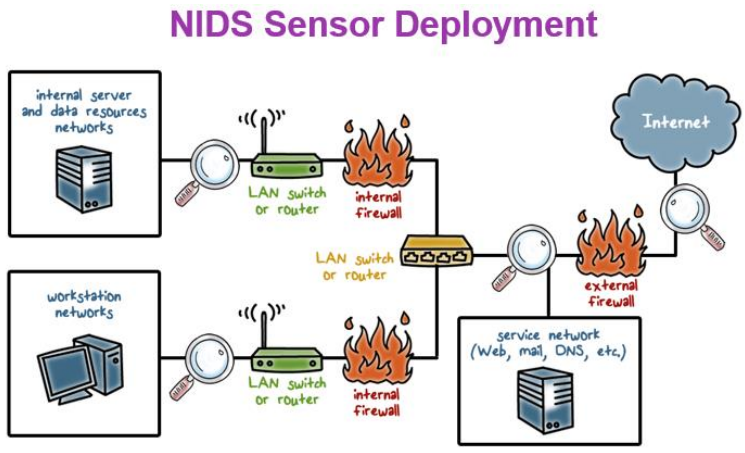
A network IDS at this location can perform more detailed analysis of traffic data. Because compared with a network IDS at the perimeter, it has smaller amount of traffic volume, due to the fact that it only monitors traffic to a subnet and the servers.



In addition, it can also detect intrusions from inside the network. In addition to protecting the servers, a network IDS can also be placed to protect the workstations or networks of important personnels or departments.

Advantages:
- Detects attacks targeting critical systems and resources
- Allows focusing of limited resources to the network assets considered of greatest value

A network IDS at this location can focus on targeted attacks, for example, attacks that are targeted at financial transaction systems. Compared with an IDS at the network perimeter, which must examine traffic to the whole network, an IDS at this location can instead focus on traffic to these high-value systems.



NIDS Sensor Deployment Quiz

When using sensors which of the following is considered good practice? Check all the **true** statements:

- ☐ Set the IDS level to the highest sensitivity to detect every attack
- ☐ Monitor both outbound and inbound traffic
- ☐ Use a shared network resource to gather NIDS data
- ☐ NIDS sensors are turnkey solutions, system administrators can interpret alerts.

Let's do a quiz on network IDS deployment. Which of these are good practice?

- ☐ Set the IDS level to the highest sensitivity to detect every attack
- ☑ Monitor both outbound and inbound traffic
- ☐ Use a shared network resource to gather NIDS data
- ☑ NIDS sensors are not turnkey solutions, system administrators must interpret alerts.

The first statement, set the IDS level to the highest sensitivity to detect every attack. This may appear to be a good idea, but in practice, this may lead to a large number of false alarms. Second, monitor both outbound and inbound traffic. This is a good idea. Because there will be a tech traffic in both directions. Third, use a shared network resource to gather NIDS data. This is not a good idea, because an attacker can disable the IDS or modify the alerts that sent. Fourth, NIDS sensors are not turnkey solutions. System admins must interpret alerts. This is true, because network IDS can produce false positives. Therefore, the system admins must interpret the alerts and take the appropriate actions.

SNORT

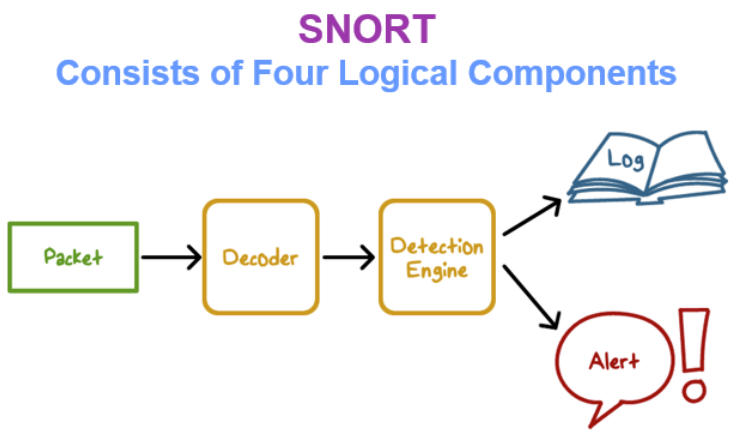
- Open source
- Highly configurable
- Lightweight IDS

Let's discuss a network ideas example, SNORT. SNORT is open source, very easy to configure and very efficient.

SNORT can be easily deployed on most nodes of a network, including end host, server, or even a router. It uses a small amount of memory and processor time. SNORT is very easy to learn, and very easily configured by sys admins. SNORT can perform real time packet captures, particle analysis, and content searching on the packet. SNORT can detect a variety of intrusions based on the rules that are configured by a sys admin. In fact, there's a community who maintains a very large set of SNORT rules that can be further configured by sys admin for his network.

SNORT

- Characteristics:
  - Easily deployed on most nodes
  - Efficient operation
  - Easily configured by system administrators
- Performs real-time packet capture
- Detects a variety of attacks and probes



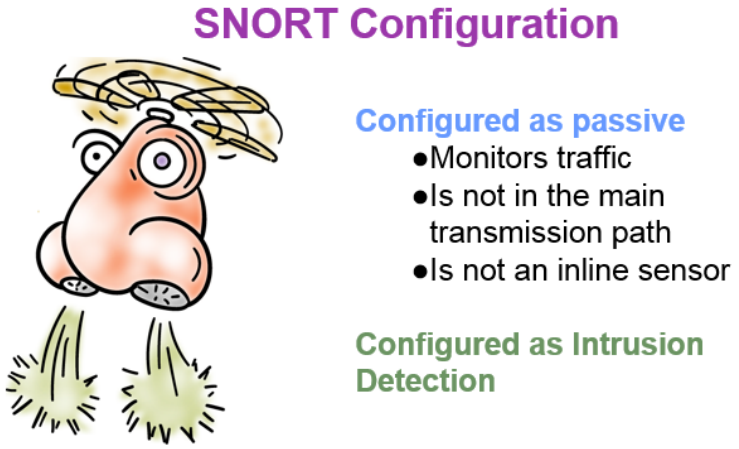
The SNORT system has four logical components.

The packet decoder possesses each capture packet to identify the protocol headers at it's data link network transport and application layers.

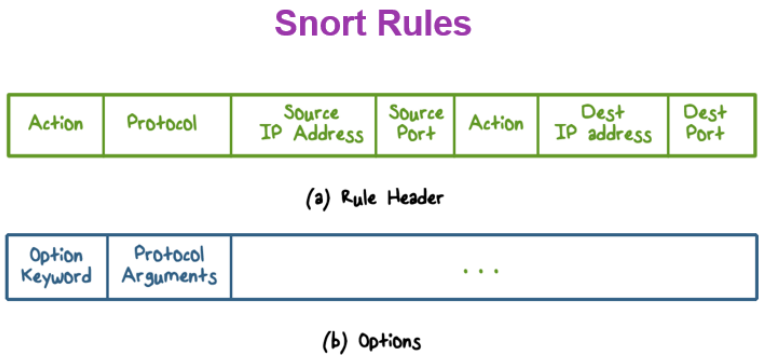
The detection engine does the actual work of intrusion detection. The detection engine checks each packet against a set of rules. That is, each packet is checked against all rules to determine if the packet matches the characters defined by a rule. A rule that matches the decoded packet would trigger the action specified by the rule. If no rule matches the packet, the detection engine will discard the packet.

For each packet that matches a rule, the rule specifies what logging or alerting options are to be taken. When the logging selection is selected, the logger stores the detected packet in human readable format. This can then use the log files for later analysis.

For each detected packet, an alert can be sent. An alert can be sent to a file, a database, or an email, etc.

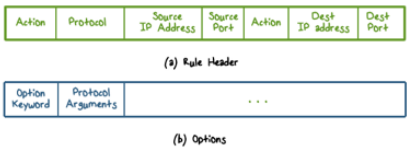


SNORT can be configured as in-line or passive. In the passive mode, it simply copies and monitors traffic, and the traffic does not pass through SNORT. That is, with the passive mode, SNORT is configured for intrusion detection.



SNORT uses a simple and very flexible rule definition language. Each rule consists of a Rule Header and number of Options.

Snort Rule Options



- **Meta-data**: provides information about the rule but do not have any effect during detection
- **Payload:** look for data inside the packet
- **Non-payload**: Look for non-payload data
- **Post-detection**: rule-specific triggers that happen after a rule has matched a packet

The options are the places where we can specify the intrusion detection logic. There are four main categories of rule options.

The first is meta data. This does not actually affect intrusion detection because it contains information such as revision number to the rule.

Payload, this is where we can specify the logic to examine the packet payload.

Non-payload, this is where we specify the logic to examine the packet headers.

Post-detection, this is where we can specify triggers, such as storing the packet information in a table so that we can correlate it with other packets.

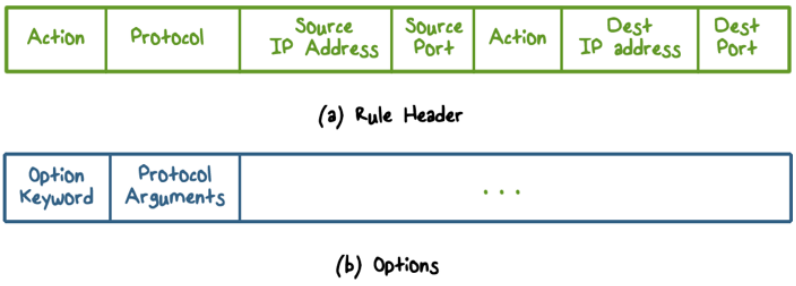
Snort Rule Actions

Action	Description	Inline Mode Only
alert	Generate an alert using the selected alert method, and then log the packet.	
log	Log the packet.	
pass	Ignore the packet.	
activate	Alert and then turn on another dynamic rule.	
dynamic	Remain idle until activated by an activate rule, then act as a log rule.	
drop	Make iptables drop the packet and log the packet.	X
reject	Make iptables drop the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.	X
sdrop	Make iptables drop the packet but does not log it.	X

The most important field in a Snort rule header is the action field. It tells Snort what to do when it finds a packet that matches the rule. Here are the possible actions. The last three are only applicable when Snort is configured as an inline sensor.

To summarize, a Snort rule has a header and multiple options. Each option consists of an option keyword which defines the option, followed by arguments which specify the details of the option.

Snort Rule Actions



Snort Rule Example:

Here's an example of Snort rule.

Typically the root user account is used only for specific privileged operations such as picking out file systems and setting up subnetworks. It is quite uncommon to send email using the root account. And such an event should trigger an alert.

alert tcp any any -> 192.168.1.0/24 25 (content: "mail from: root"; msg: "root users attempts to send an email";)

Here's an example of how Snort can capture this event. It looks for traffic to the SMTP port on any host in the /24 network, and checks if the content of the email contains, mail from: root, which indicates that a root user is sending email. It then sends an alert with the following message, root users attempt to send email.

The content keyword here is one of the more important features of Snort. It allows the sys admin to set rules that search for specific content in the packet payload and trigger response based on that data.

SNORT Quiz

Check all those who can write rules for SNORT:

Let's do a quiz on SNORT. The question is, who can write rules for SNORT?

- ☐ Users of SNORT
- ☐ The SNORT Community
- ☐ Talos Security Intelligence and Research Team

Can users of SNORT write rules? Yes. Can the community who maintains a large set of SNORT rules? Of course. Can the security experts write rules? Of course. As an open source software, everyone can write rules for SNORT. The rules can then be submitted and improved by security experts, and shared with the community.

- ☑ Users of SNORT
- ☑ The SNORT Community
- ☑ Talos Security Intelligence and Research Team

Honeypots

Honeypots are **decoy systems designed to lure attackers** away from critical systems.

Honeypots are designed to:

- divert an attacker
- collect information about an attacker
- encourage an attacker to stay long enough for administrators to respond

Honeypots is another component of the intrusion detection technologies.

Honeypots are decoy systems to attract the attackers away from the critical systems. By diverting attackers from valuable systems to honeypots, we can observe what the attackers are trying to do to our systems and networks. And based on that information, we can develop strategies to respond to the attacks.

Typically a honeypot system is filled with fabricated information to make it appear to be a valuable system on the network. A honeypot system is instrumented with monitors and event loggers so that any access, or any activity on the honeypot system is logged. In order to attract attackers to a honeypot and keep them there, so that we can gather more information about attacks. Any attack against a honeypot is made to seem successful.

Honeypots

- Honeypots are filled with **fabricated information**
- **Any accesses** to a honeypot trigger monitors and event loggers
- An attack against a honeypot is made to **seem successful**

Honeypots

- A honeypot has **no production value**
- There is **no legitimate reason to access** a honeypot
- Any attempt to communicate with a honeypot is **most likely a probe, scan, or attack**
- If a honeypot **initiates outbound traffic**, the system is most likely compromised

Most importantly, a honeypot is not a real system used by any real user. Therefore any access to honeypot is not legitimate. Most likely, any inbound connection to honeypot is a network scan or direct attack. In addition, any outbound traffic from the honey pot means that they system is most likely compromised.

Honeypot can be low or high interaction.

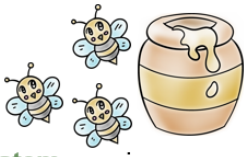
A low interaction honeypot typically, emulates some network services, such as the web server. For example, you can speak the HTTP protocol. On the other hand, it is not a full version of the service. For example, the emulated web server

Honeypot Classification

- **Low interaction honeypot:**
  - Emulates particular IT services or systems well enough to provide a realistic initial interaction, but **does not execute a full version** of those services or systems
  - Provides a **less realistic target**
  - Often **sufficient for use as a component** of a distributed IDS to warn of imminent attack

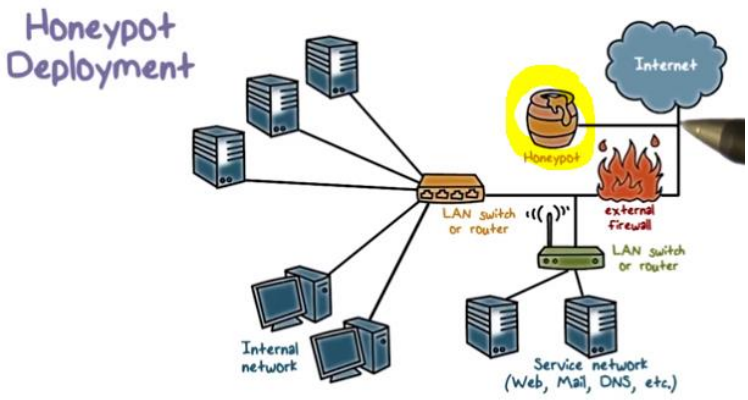
does not have all the web content and server side programs. A low interaction honeypot is typically sufficient to detect network skin and probe and imminent attacks. On the other hand, a sophisticated attacker may realize that these services are not full version and probably are not real.

Honeypot Classification



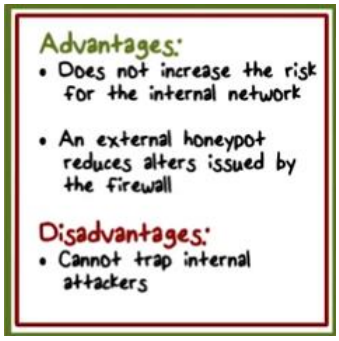
- **High interaction honeypot**
  - A **real system, with a full operating system**, services and applications, which are instrumented and deployed where they can be accessed by attackers
  - **More realistic target** that may occupy an attacker for an extended period
  - However, it **requires significantly more resources**

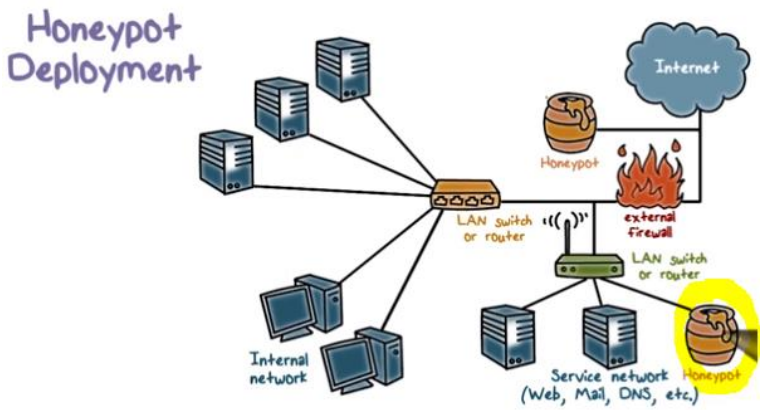
A high interaction honeypot, essentially replicates what what a real server or work station has in terms of operating systems, services and applications. In other words, they look really realistic and they can be deployed alongside with the real servers and work stations. Since a high induction honeypot mimics a real server and workstation, an attacker may be attacking it for a long time without knowing it is a honeypot. Therefore, we can learn more about the attacks. On the other hand, it is also quite challenging to make a honeypot look like a real server and workstation. For example, we must emulate user activities and never traffic on honeypot and this requires a significant amount of programming effort as well as data storage.



Honeypots can be deployed in a variety of locations on a network. A honeypot outside the external firewall is useful for tracking attempts to scan or attack the internal network.

The main advantages of placing the honeypot at this location are that. First of all, it does not have any side effect. Second, since it attracts and traps attacks to the honey pot, it reduces the amount of traffic, in particular the attack traffic to the firewall. Therefore, it reduces the amount of alerts produced by the external firewall. On the other hand, honeypot at this location does not trap internal attackers.

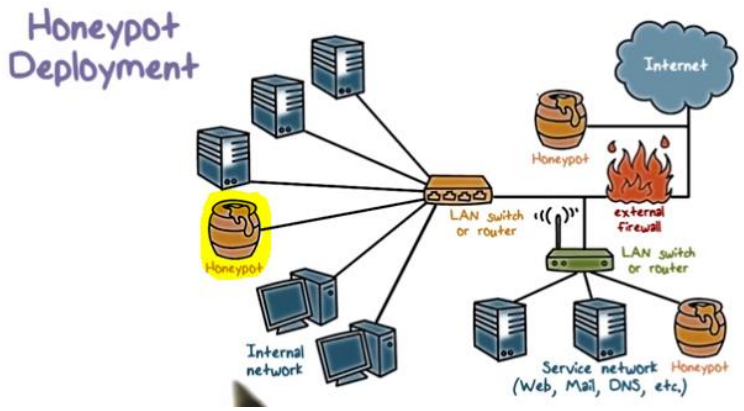




A honeypot can also be placed in a DMZ to trap attacks to the public facing service.

Disadvantages:
- The DMZ is not fully accessible

On the other hand, a honeypot at this location may not be able to trap interesting attacks. This is because a DMZ is typically not fully accessible. That is, other than the well-defined public facing services, no other services are supposed to be available in DMZ. That is, if an attacker is attempting to access the honeypot. And the service is not one of these well-defined, public facing services, the firewall is going to block the traffic. What if you say, let the firewall allow the traffic to the honeypot. But this would mean that we're opening up the firewall. And this is a security risk.



We can also place the honeypot in the internal network alongside with servers and workstations.

Advantages:
- Can also detect a misconfigured firewall

Disadvantages:
- A compromised honeypot can attack other internal systems
- The firewall must adjust its filtering to allow traffic to the honeypot

The main advantages here are that it can catch internal attacks. It can also detect a misconfigured firewall that forwards impermissible traffic from the internet to the internal network. On the other hand, unless we can completely trap the attacker within the honeypot. The attack may be able to reach other internal systems from the honeypot. In addition, in order to continue to attract and trap the attackers to the honeypot, we must allow his attack traffic from the internet to their honeypot. This means that we must open up the firewall to allow the attack traffic to come from the Internet to the internal network, and this carries a huge security risk.

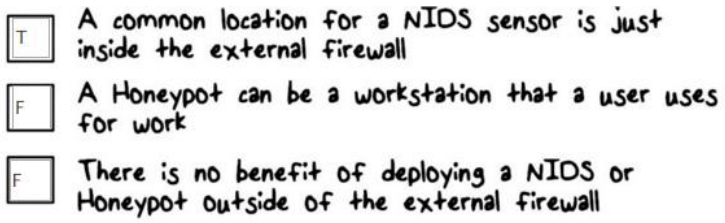
Honeypot Quiz

Put **True (T)** next to each true statement and **False (F)** next to each false statement.

And now let's do quiz on Honeypot and network intrusion detection. Decide whether each statement is true or false.

- ☐ A common location for a NIDS sensor is just inside the external firewall
- ☐ A Honeypot can be a workstation that a users uses for work
- ☐ There is no benefit of deploying a NIDS or Honeypot outside of the external firewall

First, a common location for a network intrusion detection system sensor is just inside the external firewall. This is true. This is a very typical deployment strategy of network IDS.



Second, a Honeypot can be a workstation that a user uses for work. This is false, because a Honeypot is not a real system used by any real user. Third, there's no benefit of deploying a network IDS or Honeypot outside of the external firewall. This is false.

Using a network IDS or Honeypot outside of the external firewall will allow us to see what attacks are coming from Internet to the enterprise network. In the case of Honeypot, because attacks are trapped in the Honeypot, it reduces the amount of traffic that the firewall has to process. In other words, the firewall does not need to produce as many alerts.

Evaluating IDS

Detection rate or True Positive(TP) rate: given that there is an intrusion, how likely will the IDS correct output an alert.

False Negative Rate: FN = 1 - TP

How do we evaluate an intrusion detection system? We typically use accuracy metrics to measure the detection algorithm.

We use detection rate or true positive rate to measure how well an IDS can detect intrusions. That is, given that there is an intrusion, how likely would the IDS correctly output an alert? We can also use force negative rate. That is how many intrusions did we miss?

Another aspect of detection accuracy is the false alarm rate. That is, given that there's no intrusion, how likely is the IDS is going to falsely output an alert? We can also use true negative rate. That is, how likely normal activities are currently classified as normal?

Evaluating IDS

False alarm or False Positive (FP) rate: given that there is no intrusion, how likely is the IDS to falsely output an alert.

True Negative Rate: TN = 1 - FP

Evaluating IDS

Bayesian detection rate: given that the IDS produces an alert, how likely is it that an intrusion actually occurs?

And if you are a sysadmin, you may want to know about the Bayesian detection rate of an IDS. That is, given that the IDS has already produced an alert. How likely is it that an intrusion actually occurs in your network?

Evaluating IDS

Algorithm

- Alarm/positive: A; Intrusion: I
- Detection (true positive) rate: P(A|I)
  - False negative rate P(¬A|I)
- False alarm rate: P(A|¬I)
  - True negative rate P(¬A|¬I)
- Bayesian detection rate: P(I|A)

We can more formally summarize these metrics. Here, we use:

- A to represent alarm or positive.
- I to represent intrusion.

- Detection (true positive) rate: P(A|I)

Detection rate or true positive rate is the probability that given there's an intrusion, how likely the IDS will produce an alert.

- False negative rate P(¬A|I)

And false negative rate is 1 minus true positive rate.

- False alarm rate: P(A|¬I)

False alarm rate is the probability that, given that there's no intrusion, not I, meaning it's normal. How likely an IDS would incorrectly output an alert?

- True negative rate P(¬A|¬I)

And true negative rate is 1 minus false alarm rate.

- Bayesian detection rate: P(I|A)

Bayesian detection rate is the probability that given there's an IDS alert, how likely there's likely an intrusion.

In addition to the detection algorithms, you can also evaluate IDS in terms of its system architecture.

We want the IDS to be scalable. Meaning that, it can function at high speed networks.

We also want the IDS to be resilient to attacks. Meaning that, it is not easily disabled by attacks that target the IDS.

Evaluating IDS

System should be:

- Scalable

- Resilient to attacks

Bayesian Detection Rate

$$P(I \mid A) = \frac{P(I)P(A \mid I)}{P(I)P(A \mid I) + P(\neg I)P(A \mid \neg I)}$$

P(I) is prior probability of attacks: this is the probability of intrusion evidences in the data.

Let's discuss more about the Bayesian Detection Rate.

We can use the base theorem to expand this. And we got this formula here P(I) is the prior probability of attacks. Meaning this is the probability of intrusion evidences in the data. An intuitive example is that on a typical day, what's the percentage of packets that contain intrusion activities in our network.

There's an interesting phenomenon about Bayesian detection rate called the base-rate fallacy.

Bayesian Detection Rate

- **P(I) is base rate:** prior probability of attacks
- **Base-rate fallacy**
  - Even if false alarm rate $P(A|\neg I)$ is very low, Bayesian detection rate $P(I|A)$ is still low if base-rate $P(I)$ is low
  - E.g. if $P(A|I) = 1$, $P(A|\neg I) = 10^{-5}$, $P(I) = 2\times10^{-5}$, $P(I|A) =$ 66%

Even if the false alarm rate is very low, as long as it is not zero, then the Bayesian detection rate is still low even if the base rate is also very low. For example, using the formula in the previous slide, if we plug in these numbers, meaning the detachment rate is 100%, false alarm rate is 10 to the -5, and the base rate is 2 times 10 to the -5, then the Bayesian Detection Rate is only 66%. In other words, one-third of the time when the ideas produces an alert, there is no intrusion.

If you look at these numbers more carefully, 100% detection rate is perfect. False alarm rate of 10 to the -5 is also great. This is not 0, but is very, very low. So you may ask, is this low base rate realistic? This is 2 x 10 to the -5. It depends on where do you measure the base rate. For example, if you measure base rate at the network packet level meaning that, the number of packets that contain intrusion activities which can be hundreds or thousands to a total number of packets in the network which can be tens and hundreds of millions. Then this base-rate can be quite realistic.

Bayesian Detection Rate

When the IDS produces an alert, the probability that an intrusion has actually occurred is low.

- **Implications to IDS**
  - Design algorithms to reduce false alarm rate
  - Deploy IDS to appropriate point/layer with sufficiently high base rate
  - Multiple independent detection models

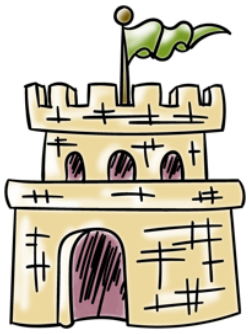
So the base-rate fallacy says that, as long as the false alarm rates' not zero, then when the IDS produces an alert the probability that an intrusion has actually occurred is also low. So how do we address this problem?

We can reduce the false alarm rate to be zero, or as much as possible. In fact, that's what the vendors of IDS have been trying to do.

Or we can deploy the IDS to the appropriate layer so that, at that layer, the base rate is sufficiently high. Modern day ideas use a hierarchal architecture to achieve this.

We can also use multiple independent models. This is similar to medical diagnosis where multiple tests are used to reduce the overall false positive rate and increase the base indication rate.

Architecture of Network IDS



- Packet data **volume can be huge**
- Base rate at the packet level **is typically low**
- Applying detection algorithms at this level **may result in a low bayesian detection rate**

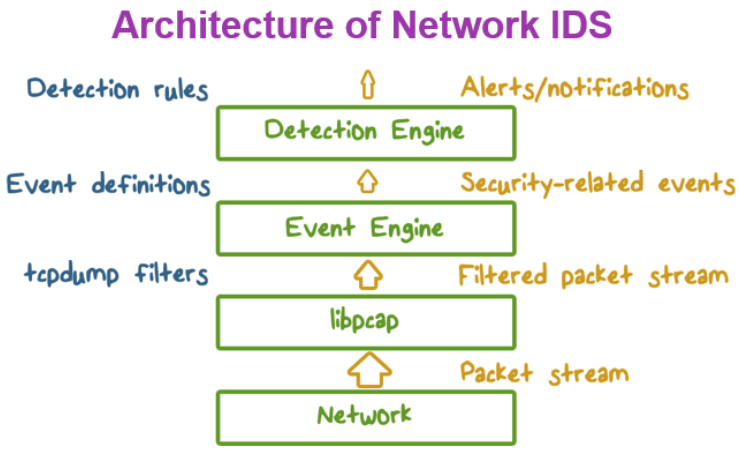
With a Bayesian detection rate and base rate fallacy in mind, lets discuss the system architecture of network IDS.

First, typically the volume of packet data in the network can be huge.

Which means the base rate at the packet level is typically low. For example, there can be tens of millions of packets per day in a network, but only a few involved the intrusion activities.

Therefore, according to the base rate fallacy, if we apply detection algorithms at the packet level, this may result in very low Bayesian detection rate.

Instead, we should apply detection models to data that has higher base rate. This can be accomplished using hierarchical architecture.



First, we can apply filters to the packet data. For example, by inserting libpcap to capture only packets to certain services.

Second, the event engine analyses the filtered packet data, and summarizes them into security related events such as failed log in's.

Finally, detection models are applied to the security related event data.

As we can see, the volume of data is decreased first by a packet filter and then the event engine. Therefore, as long as we can keep the intrusion evidence in the event data the base rate is going to be a lot higher than the original packet data. As a result the IDS model applied to the event data will yield a higher Bayesian detection rate.

IDS Quiz

What should an IDS should in order to improve detection performance? Check any statement that is true.

Check any item that is true. **To improve detection performance**, an IDS should:

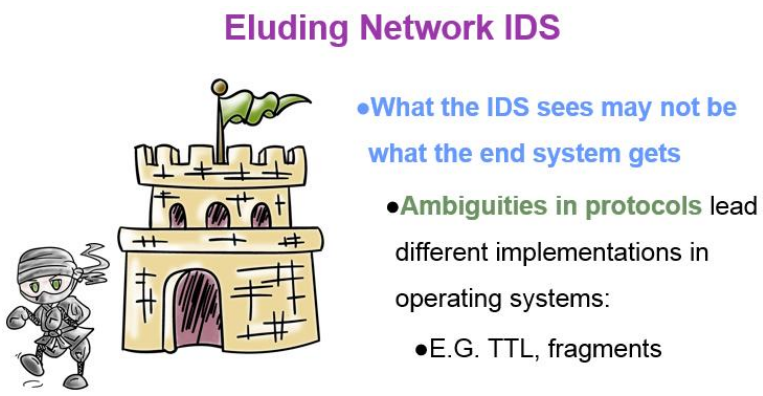
- ☐ Reduce false alarm rate while detecting as many intrusions as possible
- ☐ Apply detection models at all unfiltered packet data directly
- ☐ Apply detection models at processed event data that has higher base rate

First, reduce false alarm rate, while detecting as many intrusions as possible. This is true. Obviously, we want to detect as many intrusion as possible. We also want to reduce the false alarm rate so that we don't burden the system admins with false alarms.

- ☑ Reduce false alarm rate while detecting as many intrusions as possible
- ☐ Apply detection models at all unfiltered packet data directly
- ☑ Apply detection models at processed event data that has higher base rate

Second, apply detection models at all unfiltered packet data directly. This is false, because most likely, the base rate at this level is very low. Therefore, the IDS will have a low Bayesian detection rate.

Third, apply detection models at processed event data that has higher base rate. This is true, because as long as we can keep the intrusion evidence in the event data, the event data is going to have a higher base rate and therefore, the ideas will have a higher Bayesian detection rate.

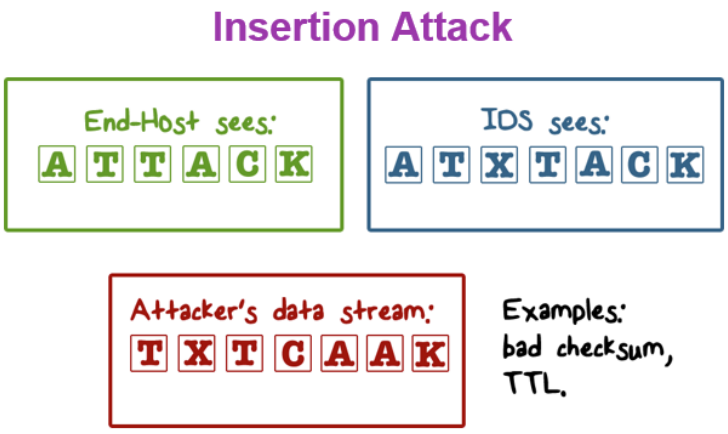


Now let's look at how an attacker can defeat the IDS so that his attack can go undetected.

Recall that an IDS performs passive monitoring. That is, a network IDS takes a copy of the network traffic and analyzes it while the traffic is reaching the end host. Therefore, in order for the IDS to detect the intrusion that's happening at the end host, he must see the same traffic as the end host. However, this is not always the case.

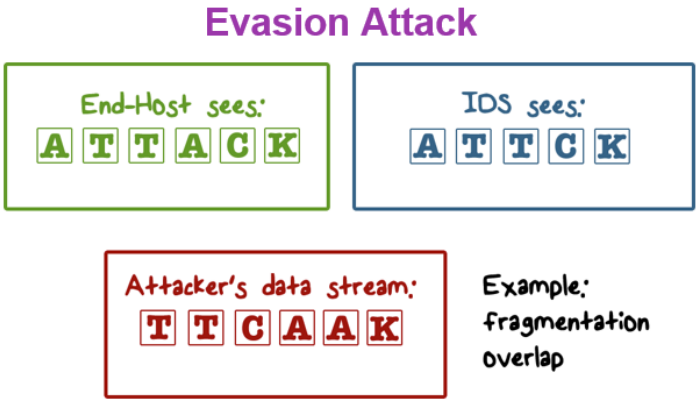
An attacker can exploit this in order to evade the IDS. The reason that the IDS and the end host are seeing different traffic is because they're using two different operating systems that process traffic in different ways. In particular, TCP/IP protocol specifications have ambiguities that lead to different implementations in different operating systems. As a result, if the IDS runs on Unix and the end host runs on Windows, they may not process certain packets exactly the same way. For example, options such as time to live or error conditions associated with fragments and checksums are handled in different ways in different operating systems. By exploiting these evidences, the attacker hopes that the IDS would not see the attack traffic, yet the end host will be affected by the attack traffic as intended by the attacker.

For example, attacker can insert data into the packet stream, to cause the IDS to miss detecting the attack. For example, by including a packet with bad checksum value, the end host may reject this packet, and yet, the IDS may accept it. As a result, the end host gets the attack, and yet, the IDS misses detecting it.

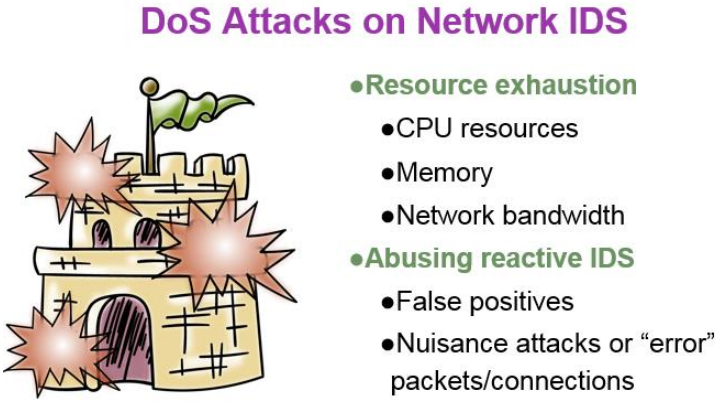


Here's an illustration of the Insertion Attack. For example, the attacker sends these packets. Although out of order, both of the IDS and the end host will assemble them according to the sequence numbers. One of the packets, X, has a bad checksum value. The IDS will accept it. Therefore, the IDS sees ATXTACK. On the other hand, the end host rejects this packet with a bad checksum value. In other words, the end host gets attacked by the traffic, yet, the IDS misses the attack.

As another example, the attacker can also hide part of the attack and cause the IDS to miss detecting the attack. For example, by sending fragments that overlap, the IDS may discard a fragment that overlaps with a previous fragment. While the end host may accept both. Again, the result is that, the IDS will miss the attack.



Here's an illustration of this evasion attack. Again, the attackers sends a packet although out of order, both the end host and the IDS will assemble them in order. However, the two As here are overlapping fragments. The IDS drops the second A, so therefore the IDS only sees A, T, T, C, K. On the other hand, the end-host accepts both fragments even though they overlap, therefore the end-host sees A, T, T, A, C, K. In other words, the end-host gets attacked by the traffic, yet the IDS misses the attack.



Attackers can also use denial of service attacks to disrupt the network intrusion detection process.

Similar to denial of service attacks on a network server such as a web server, an attacker can send a lot of traffic to the IDS to process. The result is resource exhaustions, for example, in CPU memory and network bandwidth. In other words, the network IDS may not be able to analyze traffic and such traffic may contain actual attacks. That is, the attacker can first denial service the IDS and then launch the real attack.

Another attack approach is to abuse the reactive nature of intrusion detection. The intrusion detection process is reactive because when the IDS outputs an alert, the security admin must analyze the alert. Therefore, the attacker can send a lot of traffic that would trigger the alerts, for example, by crafting packets that on purpose contain signatures of attacks. The goal is to overwhelm the response system and the security admins. And then the attacker can send the real attack traffic that even if it triggers an alert, the alert will not be acting upon in time. That is because the security admins are so busy analyzing alerts of fake attack, the real attack is not analyzed until it's too late.

Intrusion Prevention Systems (IPS)

- Also known as **Intrusion Detection and Prevention System** (IDPS)
- Is an **extension of an IDS** that includes the capability to attempt to block or prevent detected malicious activity
- Can be host-based, network-based, or distributed/hybrid
- Can use **anomaly detection to identify behavior** that is not that of legitimate users, or signature/heuristic detection to identify known malicious behavior can block traffic as a firewall does, but makes use of the types of algorithms developed for IDSs to determine when to do so

Attackers can also use analysis of his attacks to disrupt the network intrusion detection process.

In addition to intrusion detection systems, there's also intrusion prevention systems. Instead of simply sending alerts like and IDS. And IDS would try to block the attack when it detects malicious activities.

For example, it can block network traffic that involves malicious activities.

Similar to an IDS, an IPS can be deployed at the end host and the network perimeter or a combination of different locations.

An IPS also uses the similar detection algorithms of an IDS. For example, it can use anomaly detection algorithm to detect abnormal behavior. And stop such abnormal behavior.

And this is the main difference between an IPS and a firewall. A firewall typically only use simple signatures of attacks to stop traffic. Whereas, an IPS can use very sophisticated detection algorithms.

IDS Attack Quiz

Check any item that is true. **To defeat an IDS, attackers can**:

- ☐ Send a huge amount of traffic
- ☐ Embed attack in packets what cause non-uniform processing by different operating systems, e.g., bad checksum, overlapping fragments
- ☐ Send traffic that purposely matches detection rules
- ☐ Send a packet that would trigger a buffer-overload in the IDS code

How can an attacker defeat an IDS? Check any statement that's true.

First, send a huge amount of traffic. This is true. This can cause denial of service of the IDS and cause it to not be able to analyze traffic that contains attacks.

- ☑ Send a huge amount of traffic
- ☑ Embed attack in packets that cause non-uniform processing by different operating systems, e.g., bad checksum, overlapping fragments
- ☑ Send traffic that purposely matches detection rules
- ☑ Send a packet that would trigger a buffer-overflow in the IDS code

Second, embed attack in packets that cause non-uniform processing by different operating systems, for example, bad checksum and overlapping fragments. This is true because the result of this is that the IDS is seeing different traffic as the end host, and as a result, the end host may be attacked by the traffic, yet the IDS will miss it.

Third, this is true because this will result in a lot of alerts that need to be analyzed by the sysadmins. And when the sysadmins are overwhelmed, then the attacker can send his attack that although the attack is detected and an alert is produced, the sysadmin will not have time to look at the alert until it's too late.

Fourth, send a packet that would trigger a buffer-overflow in the IDS code. This is true because the buffer-overflow is a typical exploit method used to attack a program. For example, the attacker can inject his own code using buffer-overflow into a program. In other words, if the attacker can buffer-overflow an IDS, that means the attacker can now control the IDS.

Intrusion Detection
Lesson Summary

- Anomaly detection and misuse/signature detection
- Network IDS, IPS, and honeypots
- True positive, false positive, and the base-rate fallacy
- Insertion, evasion, and DoS attacks on IDS

The main intrusion detection approaches include anomaly detection and misuse detection. The main department strategies include Network IDS, IPS, and honeypots. True positive, and false positive are the main performance metrics. In the effect of false positive is highlighted by the base rate policy. IDS can be bypassed by insertion and evasion attacks. And it can be disabled by the now service attacks.