Intrusion Detection & Prevention

CISC 3325 — CUNY Brooklyn College
Lecture Notes

Intrusion Detection & Prevention

In this chapter, we will:

1. Define "Intrusion" and "Intrusion Detection System" (IDS) are, and explain why we should use the services of an IDS on our device.
2. Have a short discussion about what False Positive and False Negative mean.
3. Present the two types of Intrusion Detection Systems.
4. Bring two detection methods: (a) Signature-Based and (b) Anomaly Based.
5. Analyze the responses that an IDS may issue when it detects intrusion.

Intrusion Detection & Prevention

Intrusion is unwanted or unauthorized interference with data, a device, or a system. Being such, intrusion normally comes with bad intentions, such as disrupting a system's activity, stealing sensitive data, or modifying critical information.

An **Intrusion Detection System** (**IDS**) is a hardware and/or software solution that detects intrusion into a system or network. An IDS usually complements the activity of a firewall installed in the system.

An IDS inspects each and every packet by peeling it all the way down to its "data content" part, which is inspected for any malicious code. Afterwards, the packet is reassembled back to its original form and sent on its way.

Why do we use an IDS? Just as you don't want to sit the whole day searching youself for malware in a program's code, you'd also wish to 'automate' the process of searching for malware in packets, a goal that an IDS achieves.

Why not just using a firewall? Being considered a 'basic guard', a firewall is a necessary of a network security system but is insufficient on its own. Most of the modern networks have IDS as an essential part of the security architecture.

False Positives and False Negatives

A few more definitions:

- **Detection** is the action of finding out whether something undesirable happenned.
- **Prevention** is the action of not allowing something undesirable to happen in the first place.
- A **Response** is an action that we (users, data owners, system admins, organizations, etc.) take to address the consequences of the undesirable event.

The *detection* results of a security system (e.g., a firewall or IDS) fall into the following four cases:

- **True Positive**: When a system labels an item as 'infected', and it was indeed infected.
- **False Positive**: When a system labels an item as 'infected', but it was actually uninfected/benign.
- **True Negative**: When a system doesn't detect anything bad, and it was indeed uninfected/benign.
- **False Negative**: When a system doesn't detect anything bad, but the item was unfortunately infected/malicious.

False Positive cases lead to loss of data, efficiency, and time, while False Negative cases can harm data and the system.

False Positives and False Negatives

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Detection Cases. Taken from an article on Machine Learning by Cem Dilmegani.

When choosing between security tools on the market for your system, the goal is to reduce False Positive and False Negative cases as much as possible. This can be done by:

1. Using services of several tools in parallel (e.g., 2 firewalls, 3 anti-viruses, etc.) Challenges: (a) Might be costly (b) Might increase False Positive cases.
2. [To reduce False Negative cases:] Increasing the sensitivity/detection threshold of the security system.
3. [To reduce False Positive cases:] Decreasing the sensitivity/detection threshold of the security system.

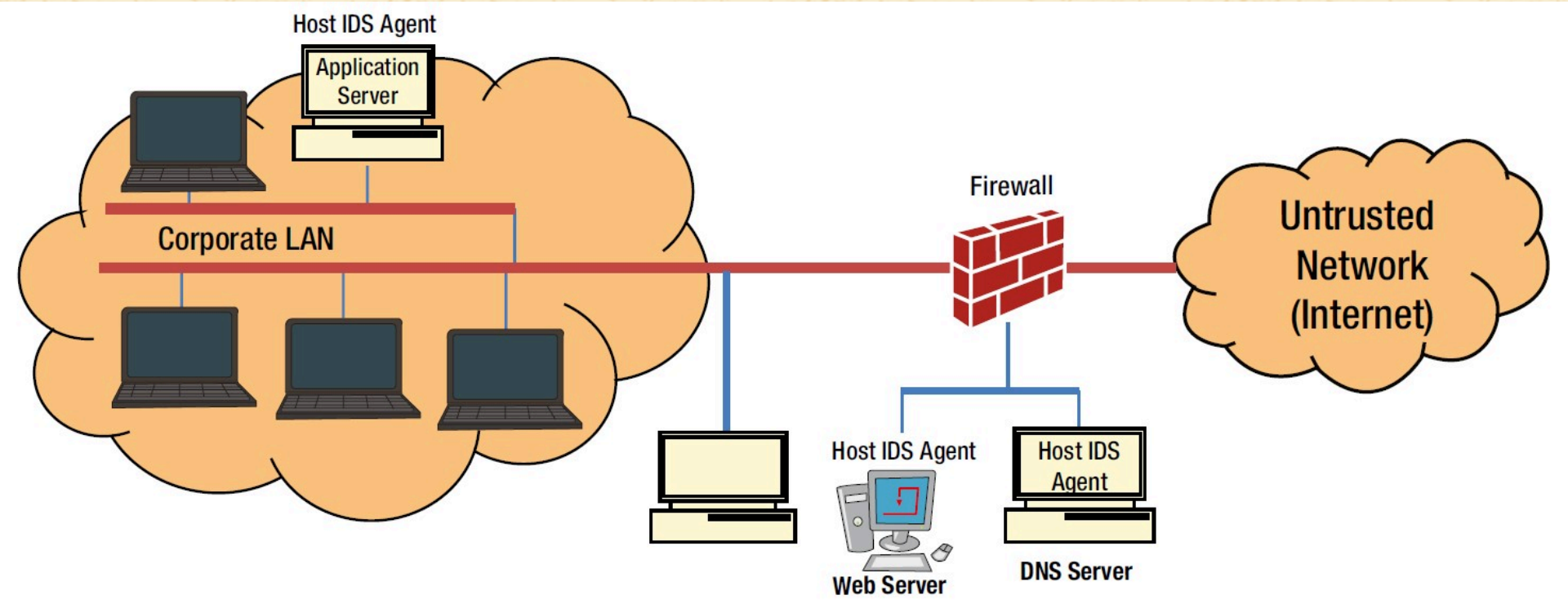
Types of Intrusion Detection Systems

There are two types of IDS:

1. **Host-based IDS (HIDS)**: Protects the end-system or the end-network resources. This is normally a software-based deployment where an agent is installed on the local host that monitors and reports the application activity.
    - HIDS monitors the access to the system and its application and sends alerts for any unusual activities.
2. **Network-based IDS (NIDS)**: Monitors network traffic for attacks. A Network IDS is deployed on the network near a firewall or even inside the trusted internal network. It checks each and every packet that is entering the network to make sure it does not contain any malicious content which would harm the network or the end system.
    - A NIDS sniffs the network traffic continuously. The traffic is matched against known signature profiles and if there are any abnormalities found in the traffic, then a NIDS triggers an alarm to the management console.

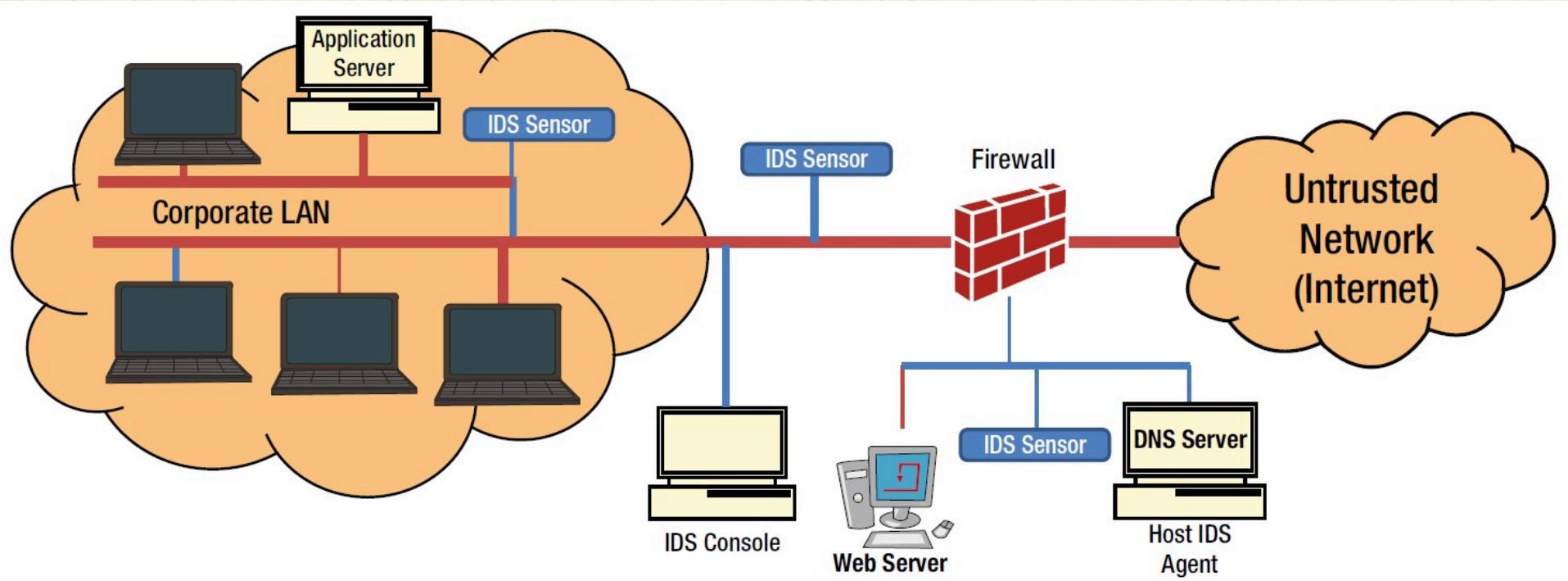
A **Intrusion Prevention System (IPS)** is an extension of an IDS that is used to *prevent* (and, sometimes, correct) intrusion. IDS only performs *detection* whereas IPS protects the network from intrusion by dropping a packet, denying entry to a packet, or blocking the connection. Together, an IPS and an IDS monitor the network traffic for malicious activities.

Types of Intrusion Detection Systems



Host-based IDS. "Figure 11-2: Host-Based Intrusion Detection System" (page 227), Nayak, U., & Rao, U. H. (2014). *The InfoSec handbook: An introduction to information security* (1st ed.). APRESS.

Types of Intrusion Detection Systems



Network-based IDS. "Figure 11-3: Network-Based Intrusion Detection and Prevention System" (page 228), Nayak, U., & Rao, U. H. (2014). *The InfoSec handbook: An introduction to information security* (1st ed.). APRESS.

Types of Intrusion Detection Systems

Let's compare Host-Based and Network-Based IDS:

Table 11-2. *Pros and Cons of H-IDS and N-IDS*

	Pros	Cons
Host IDS	Protects from attacks at the host level No Bandwidth Impact	Impacts host resources – CPU, memory Operating System dependent One agent can protect one host only
Network IDS	Protects network and network resources Protects against DoS attacks	Sensor hardware is process intensive Prone to false positives.

"Table 11-2: Pros and Cons of HIDS and NIDS" (page 229), Nayak, U., & Rao, U. H. (2014). *The InfoSec handbook: An introduction to information security* (1st ed.). APRESS.

Detection Methods

What are commonly-used detection methods by an IDS?

1. **Signature-Based Detection** searches for a known patterns of threats called **signatures** in a packet. Such patterns are stored in the **signature database** of the IDS. Signature-based detection is the simplest form of detection because it just compares the traffic with the database. If a match is found, an alert is generated; if no match is found, traffic flows without any problem. Example: a virus code in a packet's body.
2. **Anomaly-Based Detection** protects against unknown threats. An **anomaly** is anything that is abnormal. Anomalies include protocol issues, such as erroneous source and destination port numbers, illegal protocol commands, and corrupt checksums, and Denial of Service and Buffer overflow attacks. Example: HTTP traffic on a non-standard port.
    - We'll focus on checksums in the next slides.
3. **Stateful Protocol Analysis Detection** is similar to the anomaly-based detection, except that statistics on what constitutes 'normal data flow' are created by the vendors who supply the detection sensor equipment. "Stateful" means that it has the capability to keep track of the state of the protocol both in network and application layers.

Checksums

A packet's content (data) might not only get modified intentionally by hackers, but might also suffer from naturally-occurring transmission errors, such as the change of a bit (from `0` to `1`, or from `1` to `0`.)

Whether the change is due to intrusion or due to an error, the goal of an Anomaly-Based IDS is to detect that it happened and let the system, or sometimes the user, know it happened.

To detect changes to data, an IDS expects to receive and use at least one of the following along with the packet:

1. **A Parity Bit**. This bit, which is sent along with the packet, tells whether the number of `1` bits in the data is even (in which case the parity bit is `0`) or odd (in which case the parity bit is `1`).
    - **Advantage**: only one extra bit is sent with the packet, so not much space is consumed.
    - **Disadvantage**: if an even number of changes/errors happened, the parity bit won't detect them. Example: The parity bit of each of `10101` and `10011` is `1`, despite these string being different.
        - Parity bits are, therefore, not highly-reliable for detecting intrusion/errors.

Checksums

2. **A Checksum**. This number is the total of all the bits (or bytes) of a data. For example, it could be:
   - The sum of all the `1` bits.
   - The sum of all the bytes' values (e.g., by converting the characters/bytes into their corresponding ASCII or Unicode values.) Here are a Text to ASCII and a Text to Binary converter web apps.
   - [The Checksum in a TCP packet's header] The 16-bit ones' complement of the ones' complement sum of all 16-bit bytes in the header and content of a packet.

   The checksum calculation method is provided to the system ahead of time.

   Example: the ASCII-byte (decimal) checksum of `01100101`, the letter `e` in ASCII, is 101, while the checksum of `01001110`, the letter `N` in ASCII, is 78, The parity bit for each of these letters, however, is the same: `0` (***Why is the parity bit 0?***).

   Although a checksum is more reliable than a parity bit, the chance that a transmission error would result in obtaining the same checksum for a modified data isn't small.

Checksums

3. **A Hash Digest**. Recall that we spoke about Hash Functions in Topic 4: Cryptography as an effort to ensure data and password confidentiality (e.g., when deciding to save the hash digest of a password, instead of the password itself, in a database) and the non-repudiation of messages (when the hash is used as part of the digital signature, which is a proof that the message was indeed sent by the sender.)

   A hash has another wonderful use: it acts as a strong checksum to detect if an error occurred in the data.

   Unlike a checksum, which has a relatively-high chance of False Negative, a hash function returns an almost-unique string called **digest**, which is a 64-byte alpha-numerical string. The chances for a **collision** (= getting the same digest for two different texts) is very small. As such, a hash function makes a great tool for detecting changes in data!

   Many strong (and not so strong anymore) hashing algorithms have been created. One distinguishable example of strong hashing algorithms is SHA-256 (which we already got to use in Topic 4.) Here is a SHA-256 calculator.

Checksums

Fun Class Activities:

1. What is the parity bit of each of:
   a. `01010111`
   b. `01001000 01101001`
   c. `01001100 01001111 01001100`
2. What is the ASCII-byte checksum of each of the messages:
   a. `Hi`
   b. `LOL`

   [*Hint*: Find the decimal ASCII value of each of the letters in the data, and sum the values up.]
3. What is the ASCII-byte checksum of each of the messages:
   a. `2KB`
   b. `6GB`

   What are their SHA-256 digests?

Checksums

Any questions?

Intrusion Detection & Prevention: Sources

Rao, Umesh Hodeghatta, and Umesha Nayak. "Chapter 11: Intrusion Detection and Prevention Systems." *The InfoSec handbook: An introduction to information security.* Springer Nature, 2014, pp. 225-243. URL: https://link.springer.com/content/pdf/10.1007/978-1-4302-6383-8.pdf#chapter+11:+intrusion+detection+and+prevention+systems