Topic 1: Analogy

Analogy: Military Espionage

Covert Channels and Data Leakage
CSEC 640 – Module 6

Analogy: Military Espionage
Both government and private organizations employ measures to guard against data theft, but attackers still manage to subvert communication channels in those organizations. How can classified information be leaked from inside an organization without being detected by firewalls or intrusion detection systems (IDSs)? How is an illicit communication channel that facilitates data leakage established between two entities?

To understand how such covert communication channels are built, imagine two enemy spies, a general and a soldier, who are not allowed to share confidential information with each other but do so by creating a simple code that cannot be detected by observers.

The Who

Step 1
The general and the soldier are spies working for an enemy camp. The general has access to confidential information that he is not allowed to share with the soldier.

Step 2
The military cybersecurity specialist monitors any communication between the general and soldier to make sure that no classified information is passed from one to the other.

The Plot
The general wants to transmit a secret key for a military network device to the soldier. The key is 101011010001. The general and the soldier agree on a code consisting of two gestures, each of which signifies 0 or 1.

The Signal

Step 1
To transmit 1, the general brushes his hair. To the military cybersecurity specialist, the general brushing his hair is a normal gesture. However, the soldier who is aware of the code knows that the general is transmitting 1.

Step 2
To transmit 0, the general touches his glasses. To the military cybersecurity specialist, the general touching his glasses is an ordinary action, but when the soldier sees the general making that motion, he knows that the general is transmitting 0.

Step 3
Using a series of these two gestures, the general is able to transmit the secret key, 101011010001, to the soldier.

Analysis
The analogy of the general and the soldier demonstrates that it is possible to build covert channels of communication and avoid detection by using existing innocent gestures. Similarly, security policies of protected networks can be bypassed to build covert channels using systems resources and processes.
Topic 2: Module Introduction

A covert channel transmits information between two entities in a network using system resources—such as Internet Protocol (IP) header fields or device status bits—that are not intended for communication. To leak information, the sender accesses sensitive information and covertly passes the information to the receiver by manipulating these system resources.

This module covers classification of covert channels and important characteristics of covert communication. This module also discusses how a covert channel can be implemented using the Internet Control Message Protocol (ICMP), the Transmission Control Protocol (TCP/IP), and the Hypertext Transfer Protocol (HTTP).
A covert channel is an unintended communication path through which two entities in the system transmit information.

Covert channels pose threats to organizations or applications in which the main security concern is to prevent illicit information flow or data leakage.

To counter the threat posed by covert channels, many organizations use multilevel security (MLS) systems that allow data at different sensitivity levels to be simultaneously stored and processed in a system.

What Is MLS?
The purpose of MLS is to avoid the unauthorized disclosure of information at a higher security level to users assigned a lower security clearance.

Who Uses MLS?
Organizations such as military services, government agencies, and related defense industries, which are privy to classified information, are the most interested in unearthing covert channels.

How Is MLS Used?
The different types of data are labeled with security levels such as unclassified, classified, secret, and top secret. Users can access data according to the security clearance levels assigned to them.
Two communication entities on two ends of a covert channel can be considered an information sender and an information receiver.

In a MLS environment, a communication entity at a higher security level, referred to as High, acts as an information sender. An entity at a lower security level, referred to as Low, acts as an information receiver.

**How Covert Channels Work**

![Diagram of a covert channel](image)

**Trojan Horse**
A typical scenario in an MLS system is that High has access to confidential information, and it tries to leak the information to Low through a covert channel. For example, a Trojan horse in an infected system tries to send confidential information to an outside adversary.

**Covert Channel**
The Trojan horse sends confidential information through the covert channel.

**Information Receiver**
Low receives the confidential information from the covert channel.
Topic 3: Covert Channels

Types of Covert Channels

There are two kinds of covert channels: covert storage channels and covert timing channels.

Covert Storage Channel
A covert storage channel implicitly discloses information through the manipulation of one or more resources in a storage location. Take the example of an organization that has implemented a security policy specifying that High cannot communicate with Low in an MLS system. In other words, Low cannot read the contents of the files owned by High.

However, the MLS system allows both to share a directory. High can take advantage of this feature to transmit confidential information to Low.

Step 1

To transmit a bit 1, High creates a file called 1.txt in the shared directory.

Step 2

Low tries to create a file with the same name as the one created by High that is 1.txt in the shared directory.
Step 3

If the operating system displays an error message such as “the same file exists,” then Low can deduce that High has transmitted bit 1 and that High’s intention is to transmit bit 1.

Step 4

If no error message appears, Low can deduce that High has transmitted bit 0 and that High’s intention is to transmit bit 0.

**Covert Timing Channel**

A covert timing channel is an illicit communication path that a sender uses to signal information to the receiver. This communication violates an existing security policy by using system resources in such a way that this manipulation affects the response time observed by the receiver.

Take the example of an organization in which three entities exist in a network environment: High, Low, and the firewall on the High side. The TCP/IP packets are exchanged between High and Low through the firewall. The goal of the firewall is to prevent a leak of confidential information by making sure that High cannot send any TCP/IP packet with payload to Low.

The security policy of the organization mandates:
1. TCP/IP packets with payload cannot be sent from High to Low. The only exception to the rule is that High can send an acknowledgment (SYN-ACK) to Low.
2. Any packets can flow from Low to High.
The images here show how a covert timing channel can be established while the above security policy is being enforced by the firewall.

**Step 1**

The security policy ensures that High can send an acknowledgment (SYN-ACK) only when a SYN packet sent by Low reaches High. The SYN-ACK packet does not have any application payload. In order to leak confidential information to Low, High adds some delay before executing transmission of the SYN-ACK packet.

**Step 2**

To transmit 1, High waits for some time and sends the SYN-ACK packet to Low.

**Step 3**

Low observes the delay in receiving the SYN-ACK packet from High and interprets that 1 has been transmitted.
Step 4

When Low observes no delay, it knows that 0 has been transmitted.
**Topic 4: Covert Channel Characteristics**

**Transmission Cycle**

The main characteristics of a covert channel include the transmission cycle, noise, and capacity.

The transmission cycle of a covert channel includes the sender-receiver synchronization (S-R) period, transmission period, and feedback period.


![Diagram of Transmission Cycle](image)

**S-R Period**

During the S-R period, a sender needs to notify a receiver that it is ready to transmit a new symbol. For example, High may send a special packet to indicate that it is ready to start the transmission.

However, no S-R period may be needed if a sender and a receiver have some previous agreement that a new symbol will be transmitted after a predetermined interval of time. For example, in the case of the general and the soldier, the two could have a prior agreement that the general will start sending information to the soldier at 2 p.m.

**Transmission Period**

In the transmission period, the channel of communication is open and the symbols are transmitted. For example, the general makes a gesture to transmit 1 or 0, and the soldier observes his behavior.

**Feedback Period**

The feedback period is essential to ensure the continuous flow of reliable communication. During this period, the soldier acknowledges that he has understood the message sent by the general by making another gesture. The feedback period, however, can be omitted if the agreement says that the symbol can be sent every one minute.

After the general receives the acknowledgment, he is ready to send the next symbol.
Topic 4: Covert Channel Characteristics

Noise

The communication channel, including a covert channel, is typically noisy in a real-world application. In a noisy channel, messages sent by High to Low are mixed with messages sent by other legitimate entities sharing the same resource.

An information sender cannot reliably transmit a symbol to a receiver through a noisy channel. For instance, bit 1 can be converted to bit 0 during transmission because of noise.

Therefore, attackers such as Trojan horses try to build less noisy covert channels to reliably transfer data to external adversaries.


Noiseless Channel

High uses a set of available input symbols—X1 and X2—to transmit data through the covert channel. Low observes a set of output symbols—Y1 and Y2—that are transformed from the set of input symbols through the covert channel. In a noiseless channel, Low can easily decode the message sent by High because only High and Low use the channel.
Noisy Channel

Noise can alter and affect the output observed by Low during the transmission of a symbol. The behavior of a noisy channel may be nondeterministic in the sense that the output observed by the receiver is no longer a function of the input symbol transmitted.

For example, in the diagram above, when the receiver observes the output symbol Y1, it cannot reliably deduce which value—X1 or X2—was the input symbol transmitted by the sender. Therefore, a noisy channel reduces the reliability of leaked data and is useful for system security.
Topic 4: Covert Channel Characteristics

Activity: Deducing Message Content

Introduction
Consider the protected network of a company that has been compromised by a Trojan horse hiding in a computer. The intention of the Trojan horse, High, is to send a message to an outside adversary, Low, if it is able to obtain classified information.

High can transmit a set of input symbols—X1, X2, X3, X4—to Low. Low can receive a set of output symbols—Y1, Y2, Y3, Y4—from High. These output symbols are transformed from the set of input symbols through the covert channel.

In order to show that there is no covert flow between High and Low, it should be demonstrated that Low is not able to deduce with certainty anything about the activities of High.


Workspace

Question 1: In the noisy channel shown in the image, which of the symbols transmitted by High can be reliably deduced by Low? Note that “reliably deduced” means that Low does not have to make any guess or inference about which input symbol has been sent or not sent when it receives an output symbol.

Options:
- a. X1
- b. X2
- c. X3
- d. X4
- e. All of the above
- f. None of the above

Correct answer: Option f

Feedback:
Low cannot reliably deduce which input symbol has been transmitted by High. For example, when Low receives Y1, it cannot deduce exactly which symbol has been transmitted by High since any one of the input symbols—X1, X2, X3, and X4—could be a cause of Y1. This reasoning applies to all output symbols: Y1, Y2, Y3, and Y4.

This type of noisy channel, from which no information can be deduced, is called a nondeducible channel.

**Question 2:** Upon receiving an output symbol, which of the symbols transmitted by High can be reliably deduced by Low? Note that “reliably deduced” means that Low does not have to make any guess or inference about which input symbol has been sent or not sent when it receives an output symbol.

<table>
<thead>
<tr>
<th>Options:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. X1</td>
</tr>
<tr>
<td>b. X2</td>
</tr>
<tr>
<td>c. X3</td>
</tr>
<tr>
<td>d. X4</td>
</tr>
<tr>
<td>e. All of the above</td>
</tr>
<tr>
<td>f. None of the above</td>
</tr>
</tbody>
</table>

**Correct answer:** Option c

**Feedback:**
Upon receiving Y3, Low can deduce that X3 has been transmitted by High. Upon receiving Y1, Y2, or Y4, the receiver cannot deduce which symbol has been transmitted by High since X1, X2, X3, and X4 could all be possible input symbols. However upon receiving Y3, the receiver can pinpoint that X3 has been transmitted by High.

This also means High can transfer information through this noisy channel. This type of noisy channel is called a positive-deducible channel.
Question 3: Upon receiving an output symbol, which of the symbols transmitted by High can have its identity reliably deduced by Low? Note that “reliably deduced” means that Low does not have to make any guess or inference about which input symbol has been sent or not sent when it receives an output symbol.

Options:
- a. X1
- b. X2
- c. X3
- d. X4
- e. All of the above
- f. None of the above

Correct answer: Option a

Feedback:
Upon receiving Y3, Low can reliably deduce that High has transmitted either X2, X3, or X4. Equivalently, this means that High has not transmitted X1. Thus, Low, upon receiving Y3, can reliably deduce that X1 has not been transmitted by High. Therefore, this type of a channel is called a negative-deducible channel.
Question 4: Upon receiving the output symbol Y4, which of the symbols transmitted by High can be reliably deduced by Low? Note that “reliably deduced” means that Low does not have to make any guess or inference about which input symbol has been sent or not sent when it receives an output symbol.

Options:
- a. X1 or X4
- b. X2 or X3
- c. X3 or X4
- d. All of the above
- e. None of the above

Correct answer: Option a

Feedback:
What Low can deduce from this channel is somewhat limited. For instance, upon receiving Y4, Low can reliably deduce that either X1 or X4 has been transmitted. As shown in the diagram, there are two arrows originating from X1 and X4 and ending at Y4.
Review
An analysis of noisy channels reveals that if the channel between the Trojan horse and
the adversary is nondeducible, then the adversary cannot reliably deduce the intention of
the Trojan horse. On the other hand, if the channel is positive-deducible, the adversary
can easily deduce the intention of the Trojan horse.

In this example, the Trojan horse and the outside adversaries can adopt strategies to
transmit data.

The normal mode of operation for the Trojan horse is to transmit either X1, X2, or X4.
When the Trojan horse is able to access some classified data and needs to signal the
adversary, it immediately changes its mode of operation and continues sending X3.

Meanwhile, the adversary ignores other symbols and waits until it observes Y3. Upon
observing Y3, it collects the classified information.

Further Challenges
What possible strategy can the Trojan horse use to transmit the information to the
adversary using a negative-deducible channel?
Topic 4: Covert Channel Characteristics

Channel Capacity

How can the vulnerability of a covert channel be measured? Is there any security metric that can capture the severity of the vulnerability? Security researchers commonly use Shannon’s information theory to quantify the amount of information transferred from the information sender to the information receiver. The quantity of transmitted information is called channel capacity.

Channel capacity can be defined as the maximum rate of reliable and accurate information transmission through the channel.

The unit of channel capacity is bits/channel usage. For instance, 4 bits/channel usage means that senders can transmit four bits through a channel every time they use the channel.

From an information theory viewpoint, this also means that the sender can select one symbol from 16 (= 2^4) different available input symbols and transmit the symbol to the receiver.

The formula for Shannon’s channel capacity is:
\[ C = \log_2 n \text{ (bits/channel)} \]
where \( C \) is the channel capacity and \( n \) is the number of symbols available. Note that the base of log function is also 2.

Try This

Question 1: A sender transmits a symbol from two different character sets, \( x_1 \) and \( x_2 \), to a receiver through a channel without any error. What is the capacity of such a channel?

Options:
- a. \( C = 6 \text{ bits/channel} \)
- b. \( C = 2 \text{ bits/channel} \)
- c. \( C = 5 \text{ bits/channel} \)
- d. \( C = 1 \text{ bit/channel} \)

Log Table

<table>
<thead>
<tr>
<th>( x )</th>
<th>( \log_2 x )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
</tr>
<tr>
<td>2</td>
<td>1.000000</td>
</tr>
<tr>
<td>3</td>
<td>1.584963</td>
</tr>
<tr>
<td>4</td>
<td>2.000000</td>
</tr>
<tr>
<td>5</td>
<td>2.321928</td>
</tr>
<tr>
<td>6</td>
<td>2.584963</td>
</tr>
<tr>
<td>7</td>
<td>2.807355</td>
</tr>
<tr>
<td>8</td>
<td>3.000000</td>
</tr>
<tr>
<td>9</td>
<td>3.169925</td>
</tr>
<tr>
<td>10</td>
<td>3.321928</td>
</tr>
<tr>
<td>16</td>
<td>4.000000</td>
</tr>
</tbody>
</table>
Correct answer: Option d

Feedback:
Because the sender has two input symbols to choose from, \( n = 2 \). If we use the channel capacity formula with \( n = 2 \), \( C = \log_2 n \) (bits/channel) = 1 bit/channel. This answer is intuitively obvious if we assume that \( x_1 = 0 \) and \( x_2 = 1 \). The sender can transmit only 0 or 1.

Question 2: A sender accurately transmits a symbol from four different character sets—\( X_1, X_2, X_3, X_4 \)—to a receiver through a channel. What is the capacity of the channel?

Options:
- a. \( C = 3 \) bits/channel
- b. \( C = 2 \) bits/channel
- c. \( C = 6 \) bits/channel
- d. \( C = 7 \) bits/channel

Correct answer: Option b

Feedback:
The correct answer is \( C = \log_2 4 \) (bits/channel) = 2 bits/channel. The sender has four input symbols to choose from; therefore, \( n = 4 \). If we use the channel capacity formula given above with \( n = 4 \), \( C = \log_2 n \) (bits/channel) = \( \log_2 4 \) = 2 bits/channel.
Topic 5: Covert Channel Application

ICMP Covert Channel

How are covert channels established in real-life applications? Open Systems Interconnection (OSI) layers, such as the application layer and TCP/IP and ICMP protocols, can be exploited to establish a covert channel. This channel can bypass packet filters, firewalls, and network sniffers.

The data field in an ICMP echo request or reply message is intended to record router information or store timing records to calculate round-trip time. However, a covert channel can be established by using the data field in an ICMP packet to carry confidential data to an adversary.

Some operating systems and firewalls do not inspect the data field of an ICMP packet. An ICMP packet can bypass packet filters or firewalls undetected. This data-carrying capability of ICMP can be used to establish a covert channel. The length of the data field is normally 24 or 56 bytes long. However, the protocol allows the data field to be much longer, yielding a high channel capacity compared to that of a TCP/IP-based covert channel.

Data

- Some operating systems and firewalls do not inspect the data field of an ICMP packet. Therefore, an ICMP packet can bypass packet filters or firewalls undetected.
Topic 5: Covert Channel Application

IP Covert Channel

Many fields in an IP header that are optional (not used in active connections) can be used to covertly transfer data. However, the fields that are modified by some network devices, such as routers, are not appropriate for hiding and transferring data covertly.

One of the most appropriate choices for hiding data is a 16-bit identification (ID) field. The ID field gives a unique identification number to identify the fragmented packets during reassembly. The 16-bit ID field can be replaced by a mathematical product of the ASCII value of the character to be encoded and transmitted.

![The IP Header Format](image)

Example of How Data Can Be Sent Using the IP Header

Step 1

Assume that a sender, High, wants to transmit P, which has an ASCII value of 80.

Step 2
The value 20480, which is the product of 80 and 256, is put in the ID field in an IP header instead of 80 since the value 80 is too small for a 16-bit field and may look suspicious to firewalls or network filters.

**Step 3**

High sends a SYN packet with the ID value of 20480 to Low.

**Step 4**

Low scans the SYN packet and derives the value of P by dividing 20480 by 256 without engaging in a three-way TCP/IP handshake. In this way, a covert channel is established.

Topic 5: Covert Channel Application

TCP Covert Channel

The TCP header offers more possibilities for covert communication channels than the IP header. For example, TCP fields such as the sequence number, acknowledgment number, source port, flags, and the TCP timestamp option can be used to establish a covert channel.

There are two types of covert channels that exploit TCP fields to transfer data illicitly.

Sequence Number Field
Here is an example of how the sequence number field of TCP can be used to establish a covert channel.

1. The client is a Trojan horse and the server is an outside adversary. The client who wants to initiate a reliable TCP connection with a server selects an initial sequence number (ISN). Note that the client is an information sender and the server is an information receiver.

2. Now assume the client wants to send a character P, which has an ASCII value of 80, to the server. The client encodes P by inserting 5242880, the product of 80 and 65536, in the sequence field. The value 65536 is chosen to make the ISN large and realistic.

3. In the first step in a three-way handshake, the client sends the SYN packet with the ISN to the server. The ISN serves as a medium for transmitting covert data.

4. The server receives the SYN packet and decodes P by dividing the value of the ISN (5242880) by 65536. To send more characters, the client needs to transmit more SYN packets with the encoded ISNs. The server just receives the SYN packets and never engages in the three-way handshake process.

Acknowledgment Number Field

Step 1

The ACK bounce method is used to establish a covert channel using the acknowledgment field of the TCP protocol. For example, an information sender, High, wants to send data to an information receiver, Low. High can use a third party, a bounce server, to send the data without detection.

Step 2

High encodes the data stream into the ISN.
Step 3

High spoofs the source IP address of the intended receiver and sends the SYN packet to the bounce server.

Step 4

The bounce server responds to High with the acknowledgment number that is one greater than the ISN the sender chooses.
Step 5

In the final step, Low decodes the data sent by High.
Introduction
The application layer offers many opportunities for creating a covert channel. Covert data can reside either in the application protocol header or in the application payload field.

The HTTP protocol gives an attacker much more freedom to create a covert channel than the TCP/IP protocol suite does. Attackers can pass messages by using the CRLF field of the HTTP protocol and by reordering the HTTP fields.

CRLF Field
In the HTTP header, carriage return and line feed (CRLF) represents a sequence of characters, CR and LF. These two special characters are used as the end-of-line (EOL) marker for many Internet protocols, including HTTP. A parser of a Web server or client browser splits the headers based on where the CRLF is found.

HTTP treats any number of consecutive linear white space characters, such as [CRLF], space [Space], tab [Tab], as a single-space character.
Here is an example of what a typical HTTP header looks like when a Web browser sends an HTTP request to a Web server. An attacker can encode information using these nonprintable characters and modify the header.

Modify HTTP Header

The attacker uses [Space] and [Tab] to represent 0 and 1. Thus, 0101 is encoded in the second line of the HTTP header. Typically, when a firewall scans an HTTP packet and inspects its header, it ignores any white space.

When the Web browser receives the packet, it parses the white space from the header and decodes it to 0101. Thus, information is covertly transferred from the attacker to the Web server.

**Reordering of HTTP Header Field**
An attacker can covertly transmit data to an outside adversary by modifying the order of HTTP header fields. Here is an example of how HTTP header fields can be reordered.

Both the HTTP headers are legitimate and the **GET / HTTP/1.1** line cannot be reordered.

Topic 5: Covert Channel Application

Try This!

Question: Which of the following HTTP headers can be used to create a 2-bit covert channel?

Options:
- a. Option A
  ```
  GET/HTTP/1.1
  ```
- b. Option B
  ```
  GET/HTTP/1.1
  Host: www.umuc.edu
  User-Agent: Mozilla/5.0
  ```
- c. Option C
  ```
  GET/HTTP/1.1
  Host: www.umuc.edu
  User-Agent: Mozilla/5.0
  Accept-Encoding: gzip
  ```

Correct answer: Option c

Feedback for Option a:
Not quite.
To find out a number of input symbols to create a 2-bit covert channel, we can use the channel capacity formula: \( C = \log_2{n} \)

Because \( C = 2 \), \( n \) is 4. This means that four input symbols are required to create a 2-bit covert channel.

Only one symbol, either 0 or 1, can be generated from this HTTP header. Therefore, only a half-bit covert channel can be constructed using this header.

Feedback for Option b:
Not quite.
To get the number of input symbols required to create a 2-bit covert channel, we can use the channel capacity formula: \( C = \log_2{n} \)

Because \( C = 2 \), \( n \) is 4. This means four input symbols are required to create a 2-bit covert channel.

Two symbols, 0 and 1, can be generated from this HTTP header. Therefore, only a 1-bit covert channel can be constructed using this header.
The following diagram shows one example of how the HTTP header is used to represent two different input symbols to create a 1-bit covert channel.

Feedback for Option c:
That's correct.
To get the number of input symbols required to create a 2-bit covert channel, we can use the channel capacity formula: \( C = \log_2 n \)

Because \( C = 2 \), \( n \) is 4. This means four input symbols are required to create a 2-bit covert channel. The following diagram shows one example of how the HTTP header is used to represent four different input symbols to create a 2-bit covert channel.
Topic 6: Summary

We have come to the end of Module 6. The key concepts covered in this module are listed below.

- A covert channel transmits information between two entities in a network using system resources that are not intended for communication.

- In a multilevel security (MLS) environment, a communication entity at a higher security level, referred to as High, acts as an information sender, and an entity at a lower security level, referred to as Low, acts as an information receiver.

- Covert storage channels implicitly disclose information through the manipulation of one or more objects. A covert timing channel manipulates system resources to modify the response time observed by the receiver.

- The transmission cycle of a covert channel comprises the sender-receiver (S-R) period, transmission period, and feedback period.

- Channel capacity can be defined as the maximum rate of reliable and accurate information transmission through the channel. The formula for Shannon’s channel capacity is: \( C = \log_2 n \) (bits/channel), where \( n \) is the number of symbols available.

- Open Systems Interconnection (OSI) layers such as the Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), Internet Protocol (IP), and the application layer can be exploited to establish a covert channel.

- Attackers can pass messages using the carriage return and line feed (CRLF) field of the HTTP protocol and by reordering the HTTP fields.
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Capacity</td>
<td>Channel capacity can be defined as the maximum rate of reliable and accurate information transmission through the channel.</td>
</tr>
<tr>
<td>CRLF</td>
<td>CRLF represents a sequence of characters, carriage return and line feed. CRLF is used as an end-of-line (EOL) marker in the HTTP protocol.</td>
</tr>
<tr>
<td>Feedback Period</td>
<td>During the feedback period, the receiver of a message acknowledges the receipt of the message with a signal to the sender.</td>
</tr>
<tr>
<td>Firewall</td>
<td>A firewall is the hardware or software that prevents unauthorized users from accessing a computer or a network.</td>
</tr>
<tr>
<td>Hypertext Transfer Protocol</td>
<td>Hypertext Transfer Protocol (HTTP) transmits Web pages to clients.</td>
</tr>
<tr>
<td>Internet Control Message Protocol</td>
<td>The Internet Control Message Protocol (ICMP) integrates with the Internet Protocol (IP). It reports error, control, and informational messages between a host and a gateway.</td>
</tr>
<tr>
<td>Internet Protocol</td>
<td>Internet Protocol (IP) address is a numeric label that identifies each device within a computer network that communicates over the Internet.</td>
</tr>
<tr>
<td>MLS Systems</td>
<td>Multilevel security (MLS) systems allow data at different sensitivity levels to be simultaneously stored and processed in a system.</td>
</tr>
<tr>
<td>Parsing</td>
<td>Parsing is the process in which an interpreter or compiler checks the code for correct syntax and then builds a data structure.</td>
</tr>
<tr>
<td>Shannon's Information Theory</td>
<td>Shannon’s information theory mathematically deals with the fundamental limits of representation and transmission of information.</td>
</tr>
<tr>
<td>Security Policy</td>
<td>A security policy states in writing how a company plans to protect its physical and information technology assets.</td>
</tr>
<tr>
<td>S-R period</td>
<td>During the sender-receiver (S-R) period, a sender notifies a receiver that it is ready to transmit a new symbol.</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol/Internet Protocol (TCP/IP) is the communication protocol suite for the Internet.</td>
</tr>
<tr>
<td>Transmission Period</td>
<td>During the transmission period, the channel of communication between a sender and receiver is open to transmit symbols.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Trojan Horse</td>
<td>A Trojan horse is a program in which malicious or harmful code is hidden inside apparently harmless programming or data in such a way that the Trojan horse can get control and do its damage.</td>
</tr>
</tbody>
</table>