Covert Channels

2025-03-26

Table of contents

A **Covert Channel** in cybersecurity is a method used to secretly transfer information in a way that violates security policies. These channels operate by exploiting legitimate communication or computing mechanisms in unintended ways, allowing unauthorized access or data exfiltration without detection.

A **Covert Channel** typically implies the existence and use of an **overt channel**. An Overt channel is a valid or legitimate channel of communication that is being utilized to send the *covert message*. Examples of these channels include but are not limited to websites, FTP sites, phone conversations, chats, Facebook newsfeeds, TCP/IP packets, etc.

> **!** Challenge Yourself
>
> Can you think of ways that the above mentioned overt channels can be used to communicate covertly? Suggestions include code words, preset conversation systems, puzzles and codes, ...

Because typical covert channels allow the transfer of information between entities that are supposed to be allowed to communicate, as set by some access control policy, covert channels are typically

- hidden

- hard to detect
- hard to set up because they might require administrative access to machines
- low bandwidth i.e. they require a lot of resources (time and/or space) to send a very small message.

> **Note**
>
> Note that **steganography** (hiding data within data) is not a covert channel. There are a few reasons why the two are distinct but the main one is that Steganography requires a prior communication channel e.g. sending a picture, while covert channels create a channel to transmit a message potentially using the unintended side effects of a system's operation. In the end, both are surreptitious ways to send a message.

Covert Channels can typically be put into two categories.

1 Storage Covert Channels

With this category, the communication takes place using a system's storage location and will involve some subtle modification of it. The sender (or sending process) modifies a general resource that everyone has access to, and the receiver (or receiving process) reads that message.

Here are some examples of ways that this form **could** take

1. Abusing the print queue. The print queue keeps track of what jobs a printer has to work on and everyone has access to it. However, a person could send jobs to the queue, or ignore the queue to transmit a 1 or 0 respectively. Overtime, a receiver could put together the message that is being sent over the covert channel by repeatedly polling the print queue.
2. Abusing website log files. Log files are used to keep track of which webpages on a server were requested. A sender process could attempt to access a specific page (which is interpreted as a 0) and then attempt to access a different page (which is interpreted as a 1). A receiver would then have to access this log file in order to get the covert message.
3. FTP site file priviledges and permissions. FTP servers are easy to setup and can have any level of supervision. A sender could embed a message in the file permissions of all or some of the files on that FTP server. Without knowing about a hidden message, normal users would just assume that the files (which might or might not have important information) have nothing special in their permissions.

```
drwxrwxrwx 1 prof prof     0 Mar 26 17:52 'Some random file name'
```

Note that there are 10 positions in the file permissions above that can be used to transmit a covert message. This could take multiple forms but an easy one would be a 0 if there is no permission set, and a 1 if there is a permission set. This would translate to a maximum possible 10 bits of the covert message per file on the FTP server.

> **Note**
>
> FTP site file priviledges is the basis of the next Lab that you'll cover in this class

2 Timing Covert Channels

This category of covert channels relies on resource availability or system timing to transmit hidden messages. The recepient of the message would then have to pay attention to the timings of certain system tasks to decipher a message.

Here are some examples of ways that this form of covert channels **could** take.

1. File access. A sender could close and open a file for a given amount of time that the recipient would track. A short time for a 0 and a long time for a 1, or something similar.
2. Port Knocking. Later on in the quarter you'll discuss port knocking as an approach to blocking unwanted network traffic. However, a sender could use an incorrect port knocking pattern to send a message that a receiver inside the closed network could keep track of. So even though the ports to the outside would remain closed and presumably "safe", a recipient inside the network could still receive messages.
3. Hard Drive noises. Older hard drives produce specific sounds based on how the hard drive head is moving. There is an attack that was shown to have used this sound to send a covert message consisting of sensitive data from an "air-gaped computer" i.e. a computer that was not connected to the internet. The sounds were being recorded and interpreted/sent by a nearby device with a microphone. [1]

3 Network Covert Channels

Strictly speaking this isn't a sub-category of Covert channels as any attack in here could be placed in one of the two earlier categories. However, their use of network resources is a feature that these all share.

[1] https://arxiv.org/abs/1608.03431

An example is a sender placing information in the packet header of some network packets. The overt message would be the information in the actual payload of the packet and many security systems will check that payload to look for something suspicious. However, the actual content of the header isn't typically as supervised and so would allow one to send some message using it.

> **Note**
>
> We shall now go through **two** labs in which we shall set up one covert channel from each of the two main categories described above. By the end of the labs, you'll be able to send and receive messages using either channel.

The discussion in this chapter and the labs that follow it are not supposed to be a comprehensive discussion on all forms Covert Channels might take. Rather, they are designed to expose you to specific forms that have been used in the past. Note that Covert channels, as the name suggests, are difficult to identify. Unfortunately, or fortunately, they are also really easy to design. Once you see one, you should start to see ways that covert messages can easily be hidden in what looks like normal overt messaging.

For a more in-depth discussion of this topic, make sure to sign up for *CSC 446/CYEN 402: Access Control Logic and Covert Channels*. This class is offered every other year and was last offered in the Winter 2024-25 quarter.