ACADEMOR, BENGALURU

CYBERSECURITY FEBRUARY

BATCH-23

INTERNSHIP - MINOR PROJECT

MADE BY:

NAME: SOHAIM ASLAM AZMI

COURSE: M.C.A – INTEGRAL UNIVERSRITY

E-MAIL: sohaimaslam98@gmail.com

Minor Project

- Hash Cracking MD5
- DOS Attack
- Phishing Attack
- Steganography

Research on these topics make minimum ten pages and make a document in the PDF Format.

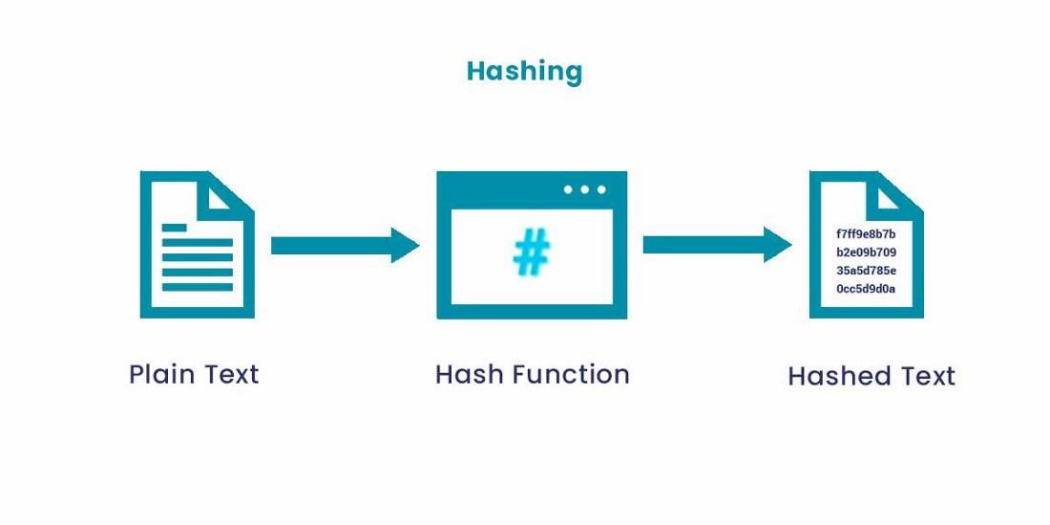
1. Hash Cracking MD5

Hashing is one of the pillars of cybersecurity. From securing passwords to sensitive data, there are a variety of use cases for hashing.

Hashing is often confused with encryption. A simple difference is that hashed data is not reversible. Encrypted data can be reversed using a key. This is why applications like Telegram use encryption while passwords are hashed.

What is Password Hashing?

Hashing is the process of converting an alphanumeric string into a fixed-size string by using a hash function. A hash function is a mathematical function that takes in the input string and generates another alphanumeric string.



There are many hashing algorithms like MD5, SHA1, and so on.

The length of a hash is always a constant, irrespective of the length of the input. For example, if we use the MD5 algorithm and hash two strings like “Password123” and “HelloWorld1234”, the final hash will have a fixed length.

Here is the MD5 hash for “Password123”.

42f749ade7f9e195bf475f37a44cafcb

If we use the input string as “HelloWorld1234”, this will be the result:

850eaebd5c4bb931dbb2bbcf7994c021

Now there is a similar algorithm called encoding. A popular encoding algorithm is base64. Here is how the same “Password123” will look if we encode it with base64:

What is Hashcat?

Hashcat is a fast password recovery tool that helps break complex password hashes. It is a flexible and feature-rich tool that offers many ways of finding passwords from hashes.

Hashcat is also one of the few tools that can work with the GPU. While CPUs are great for sequential tasks, GPUs have powerful parallel processing capabilities. GPUs are used in Gaming, Artificial intelligence, and can also be used to speed up password cracking.

Other notable features of Hashcat include:

- Fully open source.
- Support for more than 200 hashing algorithms.
- Support for Windows, Linux, and Mac.
- Support for cracking multiple hashes in parallel.

How to Install Hashcat

Hashcat comes pre-installed in Kali and Parrot OS. To install it in Ubuntu / Debian-based systems, use the following command:

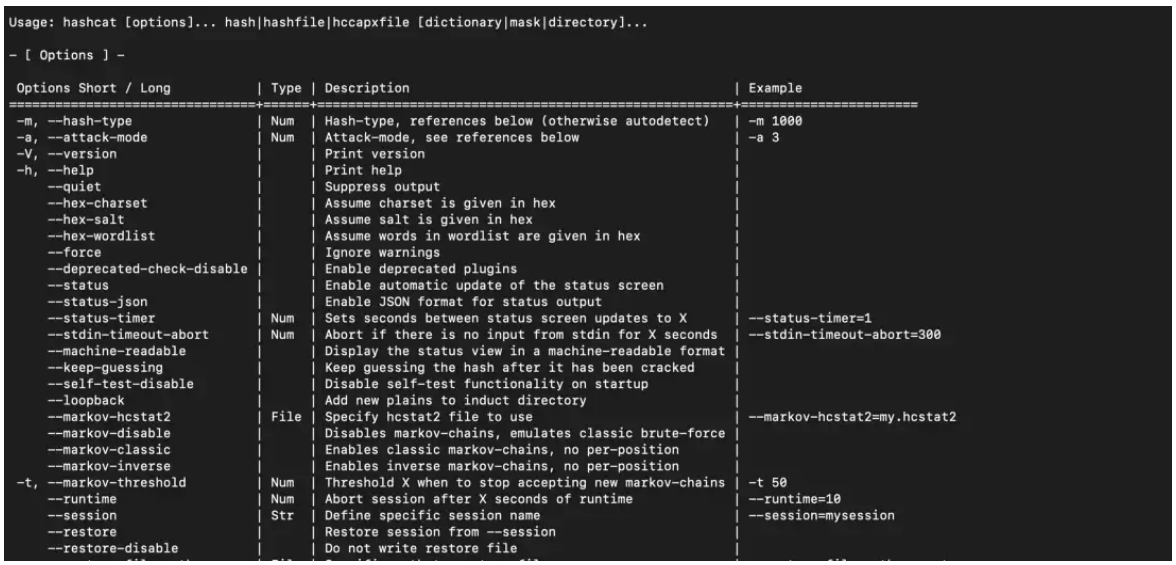
$ apt install hashcat

To install it on a Mac, you can use Homebrew. Here is the command:

$ brew install hashcat

For other operating systems, a full list of installation instructions can be found here.
Once the installation is done, we can check Hashcat's help menu using this command:

$ hashcat -h



```
Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long          | Type | Description                                          | Example
===============================+======+======================================================+=====================
 -m, --hash-type               | Num  | Hash-type, references below (otherwise autodetect)   | -m 1000
 -a, --attack-mode             | Num  | Attack-mode, see references below                    | -a 3
 -V, --version                 |      | Print version                                        |
 -h, --help                    |      | Print help                                           |
     --quiet                   |      | Suppress output                                      |
     --hex-charset             |      | Assume charset is given in hex                       |
     --hex-salt                |      | Assume salt is given in hex                          |
     --hex-wordlist            |      | Assume words in wordlist are given in hex            |
     --force                   |      | Ignore warnings                                      |
     --deprecated-check-disable |      | Enable deprecated plugins                            |
     --status                  |      | Enable automatic update of the status screen         |
     --status-json             |      | Enable JSON format for status output                 |
     --status-timer            | Num  | Sets seconds between status screen updates to X      | --status-timer=1
     --stdin-timeout-abort     | Num  | Abort if there is no input from stdin for X seconds  | --stdin-timeout-abort=300
     --machine-readable        |      | Display the status view in a machine-readable format |
     --keep-guessing           |      | Keep guessing the hash after it has been cracked     |
     --self-test-disable       |      | Disable self-test functionality on startup           |
     --loopback                |      | Add new plains to induct directory                   |
     --markov-hcstat2          | File | Specify hcstat2 file to use                          | --markov-hcstat2=my.hcstat2
     --markov-disable          |      | Disables markov-chains, emulates classic brute-force |
     --markov-classic          |      | Enables classic markov-chains, no per-position       |
     --markov-inverse          |      | Enables inverse markov-chains, no per-position       |
 -t, --markov-threshold        | Num  | Threshold X when to stop accepting new markov-chains | -t 50
     --runtime                 | Num  | Abort session after X seconds of runtime             | --runtime=10
     --session                 | Str  | Define specific session name                         | --session=mysession
     --restore                 |      | Restore session from --session                       |
     --restore-disable         |      | Do not write restore file                            |
```

Hashcat help menu

How to Work with Hashcat

Now that we know what hashing and Hashcat are, let's start cracking some passwords.

Before cracking a hash, let's create a couple of hashes to work with. We can use a site like Browserling to generate hashes for input strings.
Let's create two hashes: A MD5 hash and a SHA1 hash for the string "Password123". I'm using a weak password to help you understand how easy it is to crack these passwords.

Here are the generated hashes for the input strings.

MD5 hash -> 42f749ade7f9e195bf475f37a44cafcb

SHA1 hash -> b2e98ad6f6eb8508dd6a14cfa704bad7f05f6fb1

We can store these hashes under the names md5.txt and sha1.txt to use them when working with Hashcat.

To crack a password using Hashcat, here is the general syntax.

$ hashcat -m value -a value hashfile wordlist

Dictionary attack (-a 0)

As we saw in our example above, a dictionary attack is performed by using a wordlist. A dictionary attack is also the default option in Hashcat. The better the wordlist is, the greater the chances of cracking the password.

Combinator attack (-a 1)

The combinator attack will try different combinations of words from our wordlist. For example, if our wordlist contains the words “pass”, ”123", and ”hello”, Hashcat will generate the following wordlist.

Mask attack (-a 3)

The mask attack is similar to the dictionary attack, but it is more specific. Brute-force approaches like dictionary attacks can take a long time to crack a password. But if we have information regarding the password, we can use that to speed up the time it takes to crack the password.

Creating a list of MD5 hashes to crack

To create a list of MD5 hashes, we can use of md5sum command.

The full command we want to use is:

echo -n "Password1" | md5sum | tr -d " -" >> hashes

Here we are piping a password to md5sum so a hash is produced. Unnecessary output is then stripped and it is stored in a file in a file called “hashes”.

“echo -n ‘Password1’” is used to print the phrase “Password1”. The -n portion removes the new line added to the end of “Password1”. This is important as we don’t want the new line characters to be hashed with our password.

The part “tr –d ‘ -‘ “ removes any characters that are a space or hyphen from the output like so:

Before:

 # echo -n "Password1" | md5sum

2ac9cb7dc02b3c0083eb70898e549b63 -

After:

 # echo -n "Password1" | md5sum | tr -d " -"

2ac9cb7dc02b3c0083eb70898e549b63

For demonstration purposes, we'll create multiple MD5 hashes containing different strength passwords and output them to a file called hashes:

echo -n "Password1" | md5sum | tr -d " -" >> hashes

echo -n "HELLO" | md5sum | tr -d " -" >> hashes

echo -n "MYSECRET" | md5sum | tr -d " -" >> hashes

echo -n "Test1234" | md5sum | tr -d " -" >> hashes

echo -n "P455w0rd" | md5sum | tr -d " -" >> hashes

Once you have run these commands will look something like this:

 # cat hashes

2ac9cb7dc02b3c0083eb70898e549b63

eb61eead90e3b899c6bcbe27ac581660

958152288f2d2303ae045cffc43a02cd

2c9341ca4cf3d87b9e4eb905d6a3ec45

75b71aa6842e450f12aca00fdf54c51d

2. DoS Attack

A **Denial-of-Service (DoS) attack** is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resource they expected.

Victims of DoS attacks often target web servers of high-profile organizations such as banking, commerce, and media companies, or government and trade organizations. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and money to handle.

There are two general methods of DoS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and eventually stop. Popular flood attacks include:

- **Buffer overflow attacks** – the most common DoS attack. The concept is to send more traffic to a network address than the programmers have built the system to handle. It includes the attacks listed below, in addition to others that are designed to exploit bugs specific to certain applications or networks
- **ICMP flood** – leverages misconfigured network devices by sending spoofed packets that ping every computer on the targeted network, instead of just one specific machine. The network is then triggered to amplify the traffic. This attack is also known as the smurf attack or ping of death.
- **SYN flood** – sends a request to connect to a server, but never completes the handshake. Continues until all open ports are saturated with requests and none are available for legitimate users to connect to.

Other DoS attacks simply exploit vulnerabilities that cause the target system or service to crash. In these attacks, input is sent that takes advantage of bugs in the target that subsequently crash or severely destabilize the system, so that it can't be accessed or used.

An additional type of DoS attack is the Distributed Denial of Service (DDoS) attack. A DDoS attack occurs when multiple systems orchestrate a synchronized DoS attack to a single target. The essential difference is that instead of being attacked from one location, the target is attacked from many locations at once. The distribution of hosts that defines a DDoS provide the attacker multiple advantages:

- He can leverage the greater volume of machine to execute a seriously disruptive attack
- The location of the attack is difficult to detect due to the random distribution of attacking systems (often worldwide)
- It is more difficult to shut down multiple machines than one
- The true attacking party is very difficult to identify, as they are disguised behind many (mostly compromised) systems

Modern security technologies have developed mechanisms to defend against most forms of DoS attacks, but due to the unique characteristics of DDoS, it is still regarded as an elevated threat and is of higher concern to organizations that fear being targeted by such an attack.

Flood attacks

By saturating a targeted server with an overwhelming amount of packets, a malicious actor is able to oversaturate server capacity, resulting in denial-of-service. In order for most DoS flood attacks to be successful, the malicious actor must have more available bandwidth than the target.

What are some historically significant DoS attacks?

Historically, DoS attacks typically exploited security vulnerabilities present in network, software and hardware design. These attacks have become less prevalent as DDoS attacks have a greater disruptive capability and are relatively easy to create given the available tools. In reality, most DoS attacks can also be turned into DDoS attacks.

A few common historic DoS attacks include:

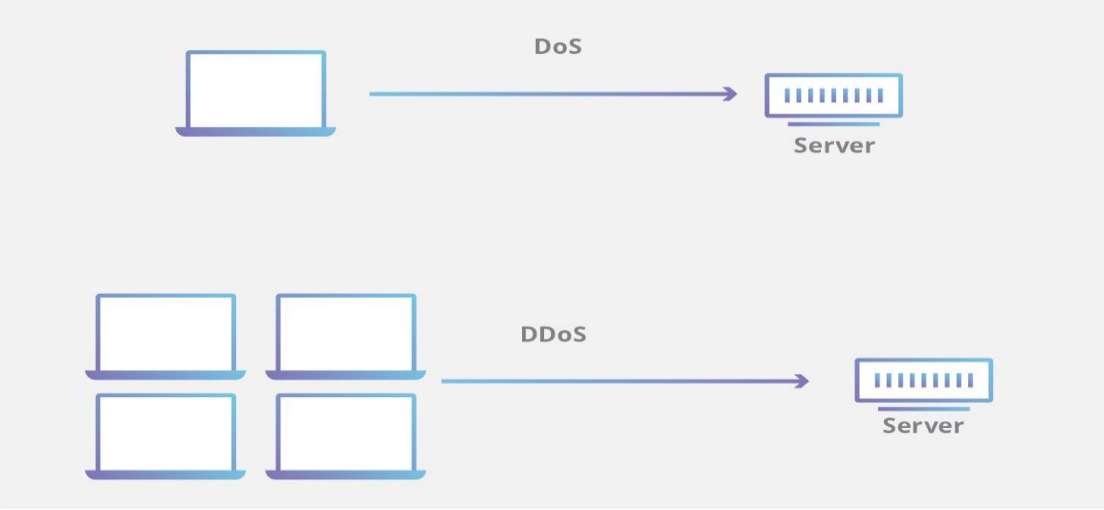
- **Smurf attack** - a previously exploited DoS attack in which a malicious actor utilizes the broadcast address of vulnerable network by sending spoofed packets, resulting in the flooding of a targeted IP address.

- **Ping flood** - this simple denial-of-service attack is based on overwhelming a target with ICMP (ping) packets. By inundating a target with more pings than it is able to respond to efficiently, denial-of-service can occur. This attack can also be used as a DDoS attack.

- **Ping of Death** - often conflated with a ping flood attack, a ping of death attack involves sending a malformed packet to a targeted machine, resulting in deleterious behavior such as system crashes.

What is the difference between a DDoS attack and a DOS attack?

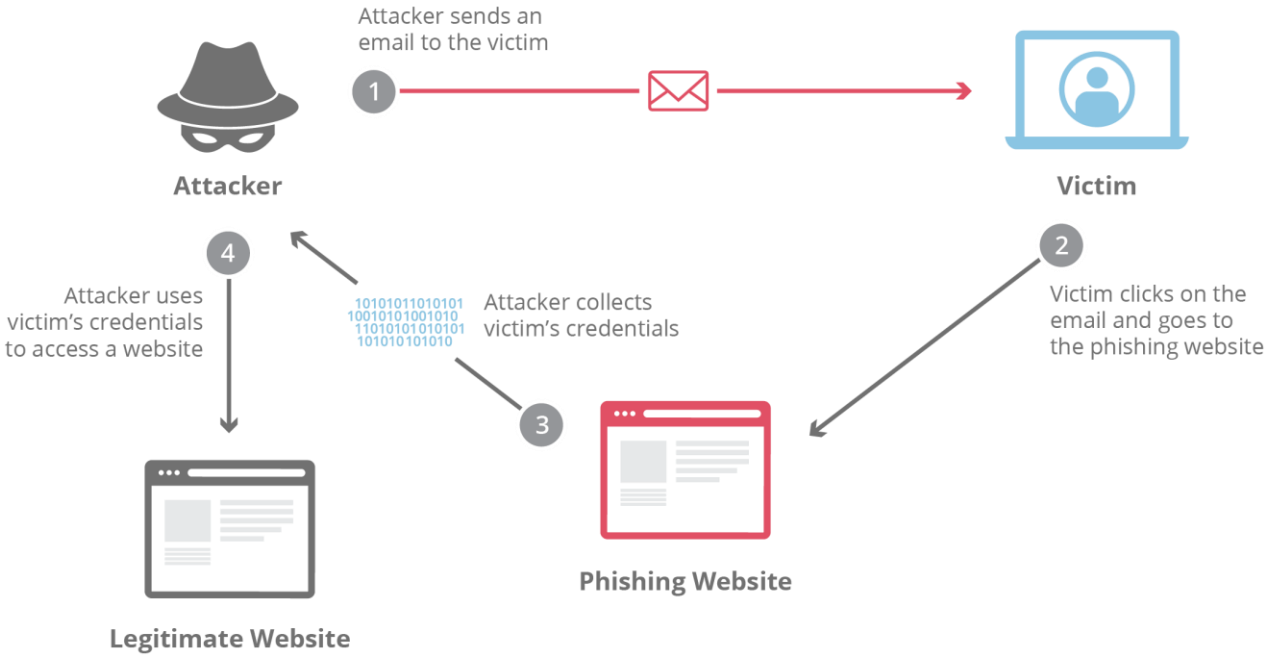
The distinguishing difference between DDoS and DoS is the number of connections utilized in the attack. Some DoS attacks, such as "low and slow" attacks like Slowloris, derive their power in the simplicity and minimal requirements needed to them be effective.



3. Phishing Attack

What is "Phishing" Attack?

"Phishing" refers to an attempt to steal sensitive information, typically in the form of usernames, passwords, credit card numbers, bank account information or other important data in order to utilize or sell the stolen information. By masquerading as a reputable source with an enticing request, an attacker lures in the victim in order to trick them, similarly to how a fisherman uses bait to catch a fish.



How is phishing carried out?

The most common examples of phishing are used to support other malicious actions, such as on-path attack and cross-site scripting attacks. These attacks typically occur via email or instant message, and can be broken down into a few general categories. It's useful to become familiar with a few of these different vectors of phishing attacks in order to spot them in the wild.

Advanced-fee scam

This common email phishing attack is popularized by the "Nigerian prince" email, where an alleged Nigerian prince in a desperate situation offers to give the victim a large sum of money for a small fee upfront. Unsurprisingly, when the fee is paid, no large sum of money ever arrives. The interesting history is that this type of scam has been occurring for over a hundred years in different forms; it was originally known in the late 1800s as the Spanish Prisoner scam, in which a con artist contacted a victim to prey on their greed and sympathy.

Account deactivation scam

By playing off the urgency created in a victim who believes an important account is going to be deactivated, attackers are able to trick some people into handing over important information such as login credentials. Here's an example: the attacker sends an email that appears to come from an important institution like a bank, and they claim the victim's bank account will be deactivated if they do not act quickly. The attacker will then request the login and password to the victim's bank account in order to prevent the deactivation. In a clever version of the attack, once the information is entered, the victim will be directed to the legitimate bank website so that nothing looks out of place.

This type of attack can be countered by going directly to the website of the service in question and seeing if the legitimate provider notifies the user of the same urgent account status.

Website forgery scam

This type of scam is commonly paired with other scams such as the account deactivation scam. In this attack, the attacker creates a website that is virtually identical to the legitimate website of a business the victim uses, such as a bank. When the user visits the page through whatever means,

be it an email phishing attempt, a hyperlink inside a forum, or via a search engine, the victim reaches a website which they believe to be the legitimate site instead of a fraudulent copy. All information entered by the victim is collected for sale or other malicious use.

In the early days of the Internet, these types of duplicate pages were fairly easy to spot due to their shoddy craftsmanship. Today the fraudulent sites may look like a picture-perfect representation of the original. By checking the URL in the web browser, it is usually pretty easy to spot a fraud.

What is spear phishing?

This type of phishing is directed at specific individuals or companies, hence the term spear phishing. By gathering details or buying information about a particular target, an attacker is able to mount a personalized scam. This is currently the most effective type of phishing, and accounts for over 90% of the attacks.

What is clone phishing?

Clone phishing involves mimicking a previously delivered legitimate email and modifying its links or attached files in order to trick the victim into opening a malicious website or file. For example, by taking an email and attaching a malicious file with the same filename as the original attached file, and then resending the email with a spoofed email address that appears to come from the original sender, attackers are able to exploit the trust of the initial communication in order to get the victim to take action.

What is whaling?

For attacks that are directed specifically at senior executives or other privileged users within businesses, the term whaling is commonly used.

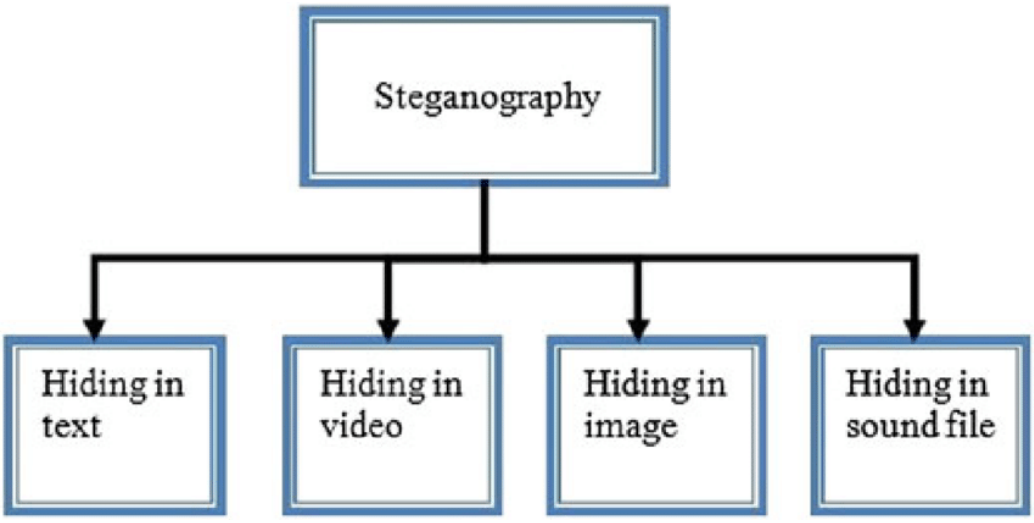
4. Steganography

A steganography technique involves hiding sensitive information within an ordinary, non-secret file or message, so that it will not be detected. The sensitive information will then be extracted from the ordinary file or message at its destination, thus avoiding detection. Steganography is an additional step that can be used in conjunction with encryption in order to conceal or protect data.

Steganography is a means of concealing secret information within (or even on top of) an otherwise mundane, non-secret document or other media to avoid detection. It comes from the Greek words steganos, which means "covered" or "hidden," and graph, which means "to write." Hence, "hidden writing."

You can use steganography to hide text, video, images, or even audio data. It's a helpful bit of knowledge, limited only by the type of medium and the author's imagination.

Although the technique is centuries old, it's still useful enough to make us justifiably pose the question, "What is steganography in cyber security?" But before we explore its uses in today's cyber security field, let's get more acquainted with the overall concept by looking at some steganography examples, then wrap things up with a fun little exercise.



Different Types of Steganography

1. Text Steganography − There is steganography in text files, which entails secretly storing information. In this method, the hidden data is encoded into the letter of each word.

2. Image Steganography − The second type of steganography is image steganography, which entails concealing data by using an image of a different object as a cover. Pixel intensities are the key to data concealment in image steganography.

Since the computer description of an image contains multiple bits, images are frequently used as a cover source in digital steganography.

The various terms used to describe image steganography include:

- **Cover-Image -** Unique picture that can conceal data.
- **Message -** Real data that you can mask within pictures. The message may be in the form of standard text or an image.
- **Stego-Image** − A stego image is an image with a hidden message.
- **Stego-Key -** Messages can be embedded in cover images and stego-images with the help of a key, or the messages can be derived from the photos themselves.

3. Audio Steganography − It is the science of hiding data in sound. Used digitally, it protects against unauthorized reproduction. Watermarking is a technique that encrypts one piece of data (the message) within another (the "carrier"). Its typical uses involve media playback, primarily audio clips.

4. Video Steganography − Video steganography is a method of secretly embedding data or other files within a video file on a computer. Video (a collection of still images) can function as the "carrier" in this scheme. Discrete cosine transform (DCT) is commonly used to insert values that

can be used to hide the data in each image in the video, which is undetectable to the naked eye. Video steganography typically employs the following file formats: H.264, MP4, MPEG, and AVI.

5. Network or Protocol Steganography − It involves concealing data by using a network protocol like TCP, UDP, ICMP, IP, etc., as a cover object. Steganography can be used in the case of covert channels, which occur in the OSI layer network model.

Steganography Examples Include

- Writing with invisible ink
- Embedding text in a picture (like an artist hiding their initials in a painting they've done)
- Backward masking a message in an audio file (remember those stories of evil messages recorded backward on rock and roll records?)
- Concealing information in either metadata or within a file header

Steganography Techniques Explained

Now that we have a better grasp on what steganography is, what forms it comes in, and who uses it, let's take a closer look at a sample of the available techniques.

- **Secure Cover Selection**

Secure Cover Selection involves finding the correct block image to carry malware. Then, hackers compare their chosen image medium with the malware blocks. If an image block matches the malware, the hackers fit it into the carrier image, creating an identical image infected with the malware. This image subsequently passes quickly through threat detection methods.

- **Least Significant Bit**

That phrase almost sounds like a put-down, doesn't it? However, in this case, it refers to pixels. Grayscale image pixels are broken into eight bits, and the last bit, the eighth one, is called the Least Significant Bit. Hackers use this bit to embed malicious code because the overall pixel value will be reduced by only one, and the human eye can't detect the difference in the image. So, no one is even aware that anything is amiss, and that the image is carrying something dangerous within.

- **Palette-Based Technique**

Like the Least Significant Bit technique, the Palette-Based Technique also relies on images. Hackers embed their message in palette-based images such as GIF files, making it difficult for cybersecurity threat hunters or ethical hackers to detect the attack.

Steganography Tools

Various tools or software that support steganography are now readily accessible. Though most hide information, some provide additional security by encrypting it beforehand. You can find the following free steganography resources online:

- **Steghide:** Steghide is a free tool that uses steganography to conceal information in other files, such as media or text.
- **Stegosuite:** It is a Java-based, free steganography tool. Stegosuite makes it simple to obfuscate data in pictures for covert purposes.
- **OpenPuff:** It is a high-quality steganographic tool that allows you to conceal data in other media types like images, videos, and Flash animations.

Advantages of Steganography

Steganography is a method that makes it easy to conceal a message within another to keep it secret. The result is that the hidden message remains hidden. A steganography approach can benefit images, videos, and audio files. Further advantages include:

- Unlike other methods, steganography has the added benefit of hiding communications so well that they receive no attention. However, in countries where encryption is illegal, sending an encrypted message that you can easily decipher will raise suspicion and may be risky.
- Steganography is a form of encryption that protects the information within a message and the connections between sender and receiver.
- The three essential elements of steganography—security, capacity, and robustness—make it worthwhile to covert information transfer via text files and develop covert communication channels.

References:

https://www.freecodecamp.org/news/hacking-with-hashcat-a-practical-guide/

https://www.4armed.com/blog/hashcat-crack-md5-hashes/

https://www.simplilearn.com/what-is-steganography-article

https://www.cloudflare.com/learning/access-management/phishing-attack/#:~:text=%E2%80%9CPhishing%E2%80%9D%20refers%20to%20an%20attempt,or%20sell%20the%20stolen%20information.

https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos

https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/