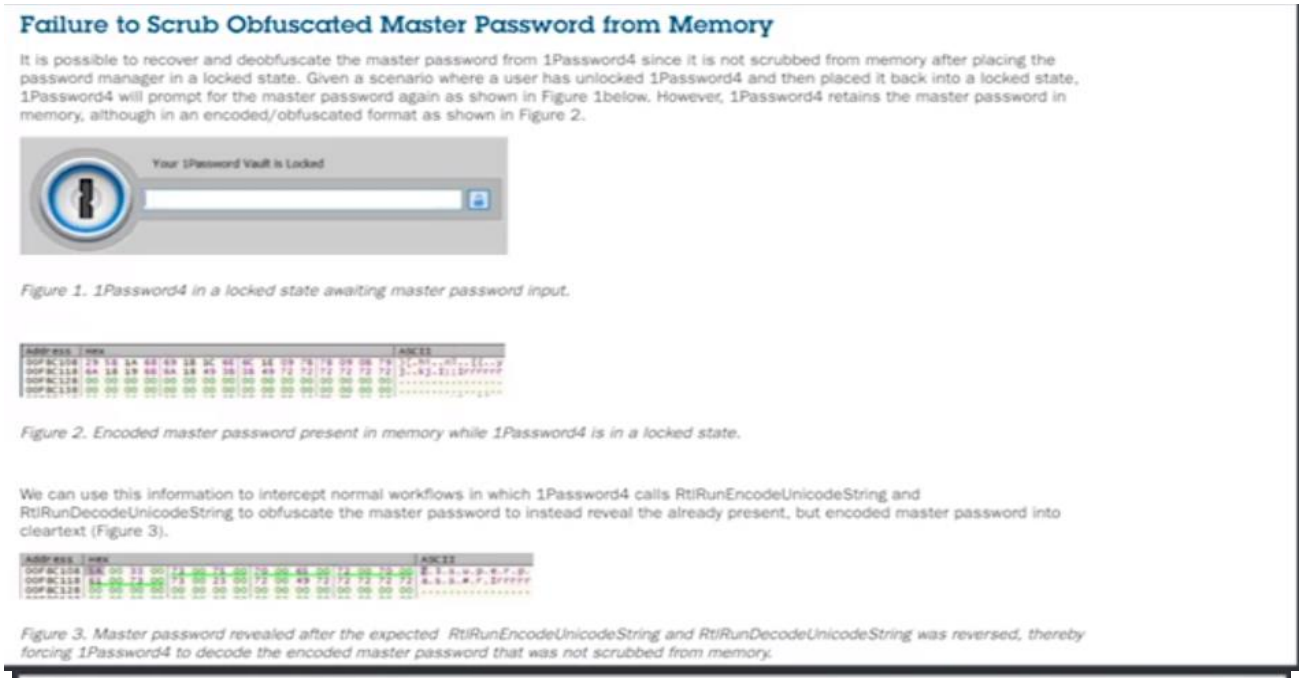
DAILY ASSESSMENT FORMAT

Date:	17th June 2020	**Name:**	Soundarya NA
Course:	Cyber security	**USN:**	4AL16EC077
Topic:	Cyber security	**Semester & Section:**	8th - B

FORENOON SESSION DETAILS

Image of session



Failure to Scrub Obfuscated Master Password from Memory

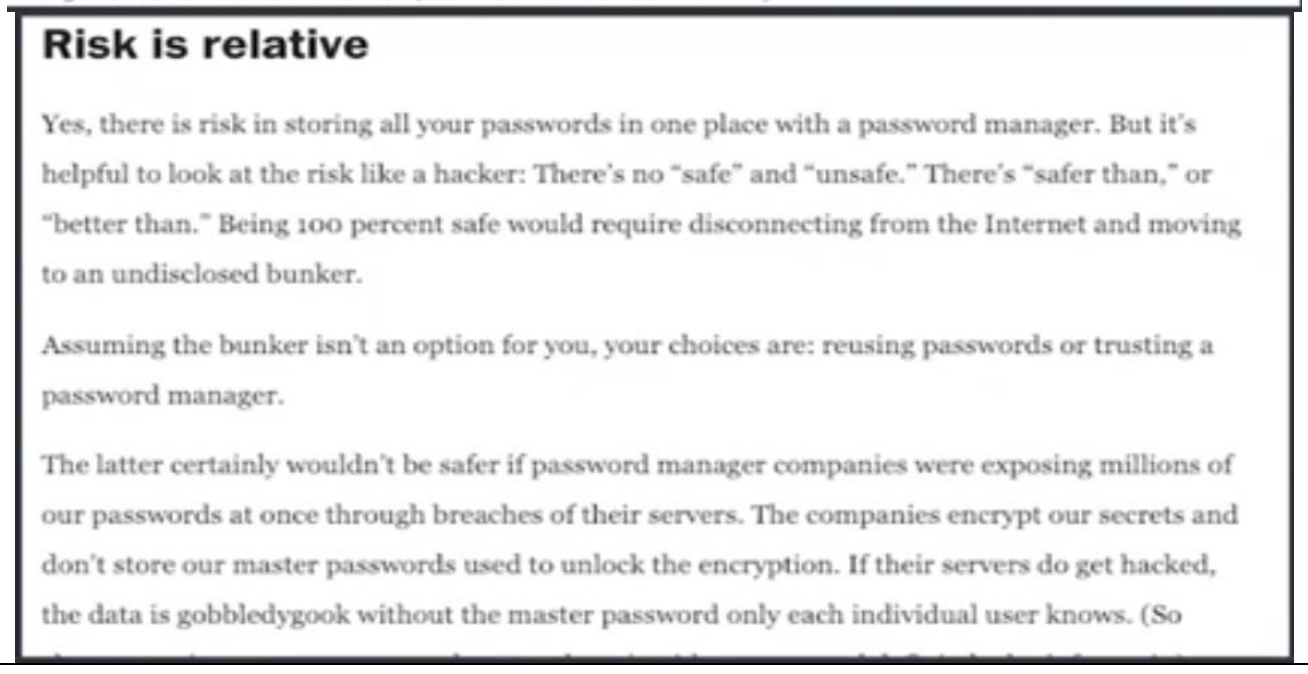
It is possible to recover and deobfuscate the master password from 1Password4 since it is not scrubbed from memory after placing the password manager in a locked state. Given a scenario where a user has unlocked 1Password4 and then placed it back into a locked state, 1Password4 will prompt for the master password again as shown in Figure 1below. However, 1Password4 retains the master password in memory, although in an encoded/obfuscated format as shown in Figure 2.

Figure 1. 1Password4 in a locked state awaiting master password input.

Figure 2. Encoded master password present in memory while 1Password4 is in a locked state.

We can use this information to intercept normal workflows in which 1Password4 calls RtlRunEncodeUnicodeString and RtlRunDecodeUnicodeString to obfuscate the master password to instead reveal the already present, but encoded master password into cleartext (Figure 3).

Figure 3. Master password revealed after the expected RtlRunEncodeUnicodeString and RtlRunDecodeUnicodeString was reversed, thereby forcing 1Password4 to decode the encoded master password that was not scrubbed from memory.



Risk is relative

Yes, there is risk in storing all your passwords in one place with a password manager. But it's helpful to look at the risk like a hacker: There's no "safe" and "unsafe." There's "safer than," or "better than." Being 100 percent safe would require disconnecting from the Internet and moving to an undisclosed bunker.

Assuming the bunker isn't an option for you, your choices are: reusing passwords or trusting a password manager.

The latter certainly wouldn't be safer if password manager companies were exposing millions of our passwords at once through breaches of their servers. The companies encrypt our secrets and don't store our master passwords used to unlock the encryption. If their servers do get hacked, the data is gobbledygook without the master password only each individual user knows. (So

Report:
Vulnerabilities and Password Security:

5 Most Common Password Vulnerabilities:

- Password Vulnerability due to Phishing
- Brute Force Attack puts your password at risk
- Dictionary or Wordlist Attack
- Social Engineering
- Malware attack on passwords increasing by the day

Password Vulnerability due to Phishing:

This type of attack causes victims to believe they are accessing legitimate content, usually e-mail or websites, when in fact they are accessing fake content produced by the attackers.

This type of content usually leads victims to fill in existing login and password data from other legitimate sites or services, such as Google and Facebook, which when filled in, allows the attacker to store the passwords before redirecting the victims to a legitimate site.

How to avoid:

Attackers often copy the image of sites almost perfectly, that are looking to steal passwords. But there are a few important items which cannot be copied, such as the site addresses and the links within it. Always check the links to make sure they belong to the desired location.

Brute Force Attack puts your password at risk:

A brute force attack is the name of the action performed on a website to test it with thousands of software, check against millions of passwords until you find the right one. It is a robot that randomly tries passwords to connect to the website.

How to avoid:

No one can really prevent a robot from doing these actions, but it is possible to reduce and discourage such hackers. One of the first solutions is to increase the security of the website by forcing its members

to create more complex passwords. For example, a minimum of 8 characters, containing a combination of numbers and letters. This will make the task of the robot much more complex.

Dictionary or Wordlist Attack:

The dictionary-based attack or wordlist attack is also considered a brute-force attack. The attacker uses files containing thousands or even millions of words of the most varied types and languages and software that allows this list to be tested quickly until the victim's password is found or until the dictionary finishes.

How to avoid:

Usually, the passwords present in dictionaries are not very extensive, that is, they have less than ten characters. To avoid becoming a victim of dictionary attacks use passwords that have more than 12 characters.

Like most attacks, the above attacks can be prevented by adopting some simple behavioral changes, and there are security solutions that can make this task even simpler.

Social Engineering:

Social engineering is somewhat similar to phishing attacks and is a widespread spying method aimed at gaining access to confidential data.

To extract confidential information, scammers very often exploit good faith, helpfulness, but also insecurity of people. Whether over the phone, pretending to be someone else, or the Internet, they are ready to do anything to get access to personal data.

How to avoid:

Reveal as little personal information as possible; social networks are real mines of information. Be suspicious when asked for email ID. Even emails from known senders can be falsified.

What is Cryptography?

Cryptography is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it. The prefix "crypt-" means "hidden" or "vault" -- and the suffix "-graphy" stands for "writing."

In computer science, cryptography refers to secure information and communication techniques derived from mathematical concepts and a set of rule-based calculations called algorithms, to transform messages in ways that are hard to decipher. These deterministic algorithms are used for cryptographic key generation, digital signing, verification to protect data privacy, web browsing on the internet, and confidential communications such as credit card transactions and email.

Cryptography Techniques:

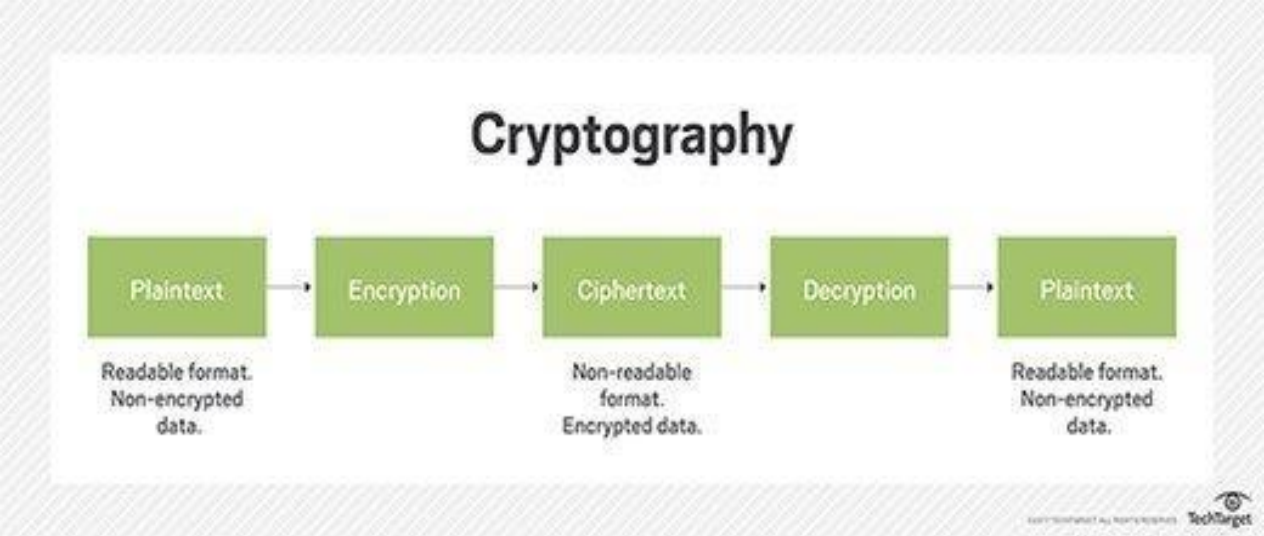
Cryptography is closely related to the disciplines of cryptology and cryptanalysis. It includes techniques such as microdots, merging words with images, and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with scrambling plaintext (ordinary text, sometimes referred to as cleartext) into ciphertext (a process called encryption), then back again (known as decryption). Individuals who practice this field are known as cryptographers.

Modern cryptography concerns itself with the following four objectives:

1. **Confidentiality:** the information cannot be understood by anyone for whom it was unintended
2. **Integrity:** the information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected
3. **Non-repudiation:** the creator/sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information
4. **Authentication:** the sender and receiver can confirm each other's identity and the origin/destination of the information

Procedures and protocols that meet some or all of the above criteria are known as cryptosystems. Cryptosystems are often thought to refer only to mathematical procedures and computer programs;

however, they also include the regulation of human behavior, such as choosing hard-to-guess passwords, logging off unused systems, and not discussing sensitive procedures with outsiders.



Cryptographic Algorithms:

Cryptosystems use a set of procedures known as cryptographic algorithms, or ciphers, to encrypt and decrypt messages to secure communications among computer systems, devices such as smartphones, and applications. A cipher suite uses one algorithm for encryption, another algorithm for message authentication, and another for key exchange. This process, embedded in protocols and written in software that runs on operating systems and networked computer systems, involves public and private key generation for data encryption/decryption, digital signing and verification for message authentication, and key exchange.

Message Integrity:

Message integrity means that a message has not been tampered with or altered. The most common approach is to use a hash function that combines all the bytes in the message with a secret key and produces a message digest that is difficult to reverse.

If Bob receives a message (either be encrypted or be in plaintext) from Alice, he needs to verify:

- The message indeed originated from Alice.
- The message was not tempered with on its way to Bob.

Before talking about how to achieve the above goals, we need to introduce cryptographic hash functions.

A hash function takes an input, m, and computes a fixed-size string H(m). A cryptographic hash function also needs to satisfy an additional property:

- It is computationally infeasible to find any two different messages x and y such that H(x) = H(y).

Common hash algorithms are MD5 [RFC 1321] and SHA-1 [FIPS 1995].

After we having the cryptographic hash function. A naive step to perform message integrity would be:

- Alice creates message m and send Bob (m, H(m)).
- After Bob receives (m, H(m)), he uses the same hash function to check whether after hashing m, the result is equal to the H(m) he received from Alice.

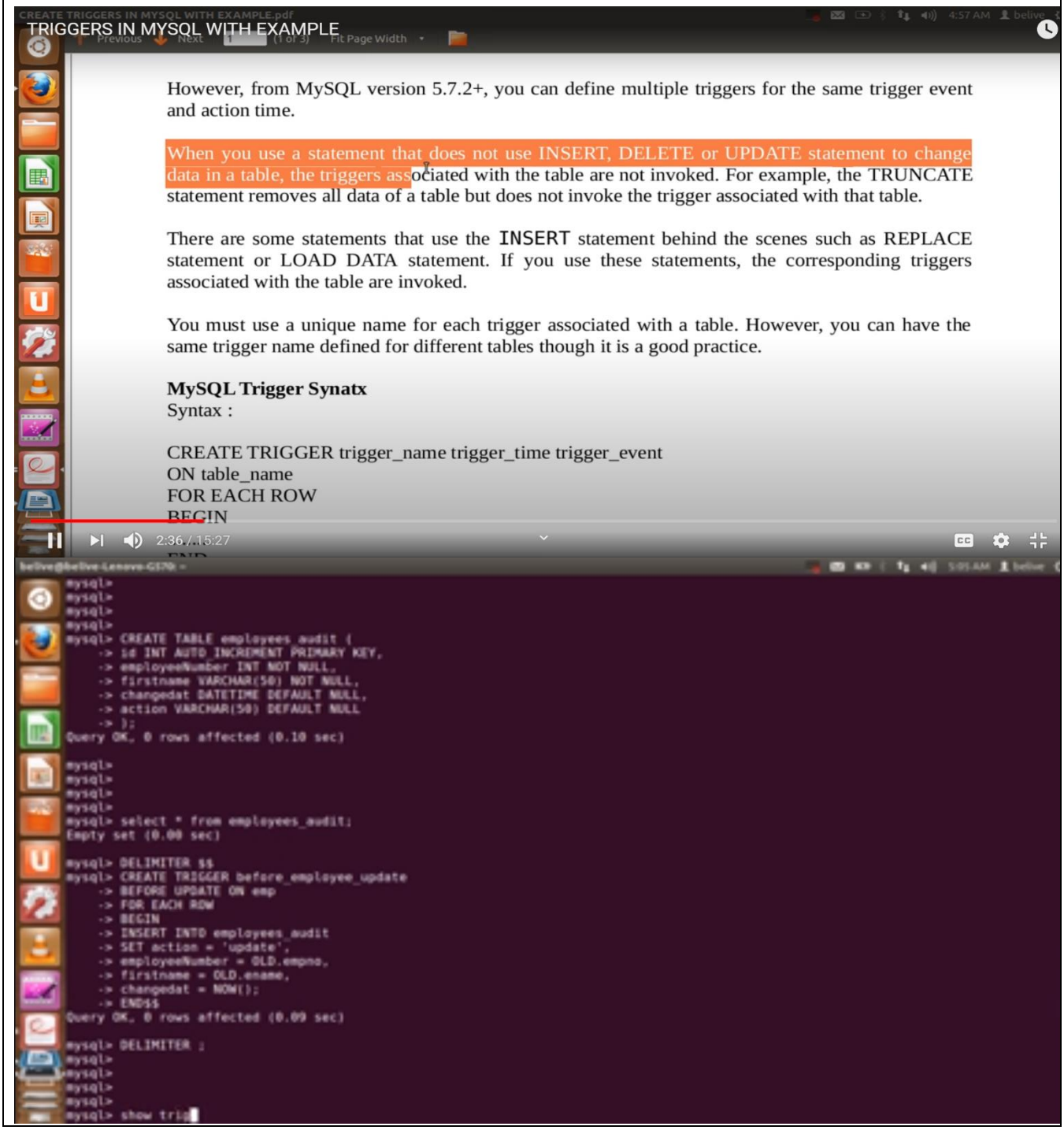
The approach is flawed because other party can also send message to bob (m', H(m')) claiming that this is from Alice, and Bob has no way to tell even the hash result is correct.

In order to prove the message is indeed from Alice, Alice and Bob should have a shared secret s. This shared secret is called authentication key. Using this shared secret, message integrity can be performed as:

- Alice creates message m, concatenates s to m, calculates H(m+s), sends Bob (m, H(m+s)). H(m+s) here is called the message authentication code (MAC).
- Bob receives (m, H(m+s)), calculates the MAC using his s, check whether it is the same H(m+s) he received from the message.

AFTERNOON SESSION			
Date:	17th June 2020	**Name:**	Soundarya NA
Course:	UDEMY	**USN:**	4AL16EC077
Topic:	Mysql	**Semester & Section:**	8th - B

Image:



Report:

Database Triggers:

The MySQL trigger is a database object that is associated with a table. It will be activated when a defined action is executed for the table. The trigger can be executed when you run one of the following MySQL statements on the table: INSERT, UPDATE and DELETE and it can be invoked before or after the event.

Code:

```
mysql> CREATE TABLE people (age INT, name varchar(150));
mysql> delimiter //
mysql> CREATE TRIGGER agecheck BEFORE INSERT ON people FOR EACH ROW IF NEW.age < 0 THEN SET NEW.age = 0; END IF;//
Query OK, 0 rows affected (0.00 sec)
mysql> delimiter ;
mysql> INSERT INTO people VALUES (-20, 'Sid'), (30, 'Josh');
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

Result:

```
mysql> SELECT * FROM people;
+——-+——-+
| age | name |
+——-+——-+
| 0 | Sid |
| 30 | Josh |
+——-+——-+
2 rows in set (0.00 sec)
```