Data Security

Module 2: 08

Objectives

1. SQL Injection
2. Hashing
    a. Salting
    b. Hashing Passwords
3. Encryption
    a. Data at Rest with Symmetric Encryption
    b. Data in Transit with Asymmetric Encryption
    c. Man In the Middle Attack

(https://www.coursera.org/articles/popular-cybersecurity-certifications)
(https://www.coursera.org/articles/cybersecurity-jobs)

SQL Injection

```
"SELECT * FROM app_user WHERE UPPER(user_name) = '" +userName.toUpperCase()
                + "' "+ "AND password = '" + password + "'"
```

What if I enter a valid username (Bill) and then the password as: `' OR 1=1--`

```
SELECT * FROM app_user WHERE UPPER(user_name) = 'BILL'
            AND password = '' OR 1=1--'
```

What is the result of this query?

It returns the row of data were user_name = “Bill” regardless of the password, because OR 1=1 is always TRUE.

The trailing -- changes the remainder of the SQL statement into a comment, ending the query after OR 1=1.

Examples of SQL Injection

1. Query Modification

   The attacker modifies the original query and then Ignores the rest of the original by adding -- at the end of their addition to comment it out.

2. Stacked Queries

   The attacker ends the original query with a ; and then appends their own query onto the original..

Preventing SQL Injection

1. **Parameterized Queries** - The single most effective thing you can do to prevent SQL injection is to use parameterized queries. ***If this is done consistently, First Order SQL injection will not be possible*****, however, second level attacks are still possible.**

2. **Input Validation** - Limiting the data that can be input by a user can certainly be helpful in preventing SQL Injection, but is by no means an effective prevention by itself. **If done consistently then Second Order SQL Injection will are also prevented.**

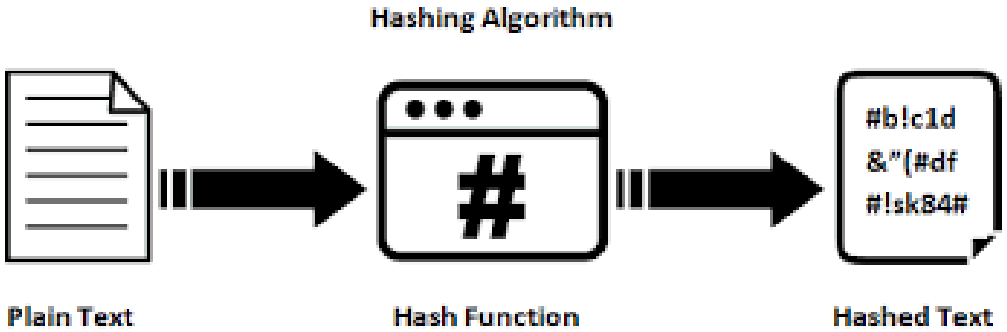
3. **Limit Database User Privileges** - A web application should always use a database user to connect to the database that has as few permissions as necessary.

Hashing

A Hash Function is one that can map input data of arbitrary size to a fixed size output.

Hashing is 1-way, meaning that once data is hashed, the hash cannot be reversed back into the original data.

Commonly used to store passwords.

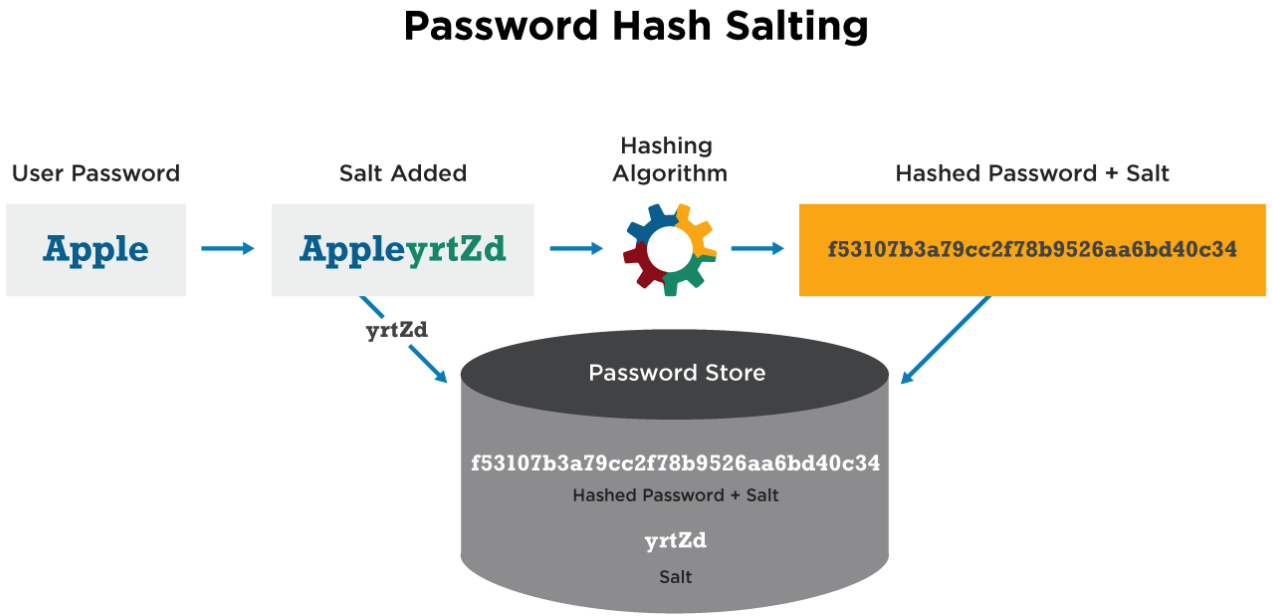


MD5 Hash Generator

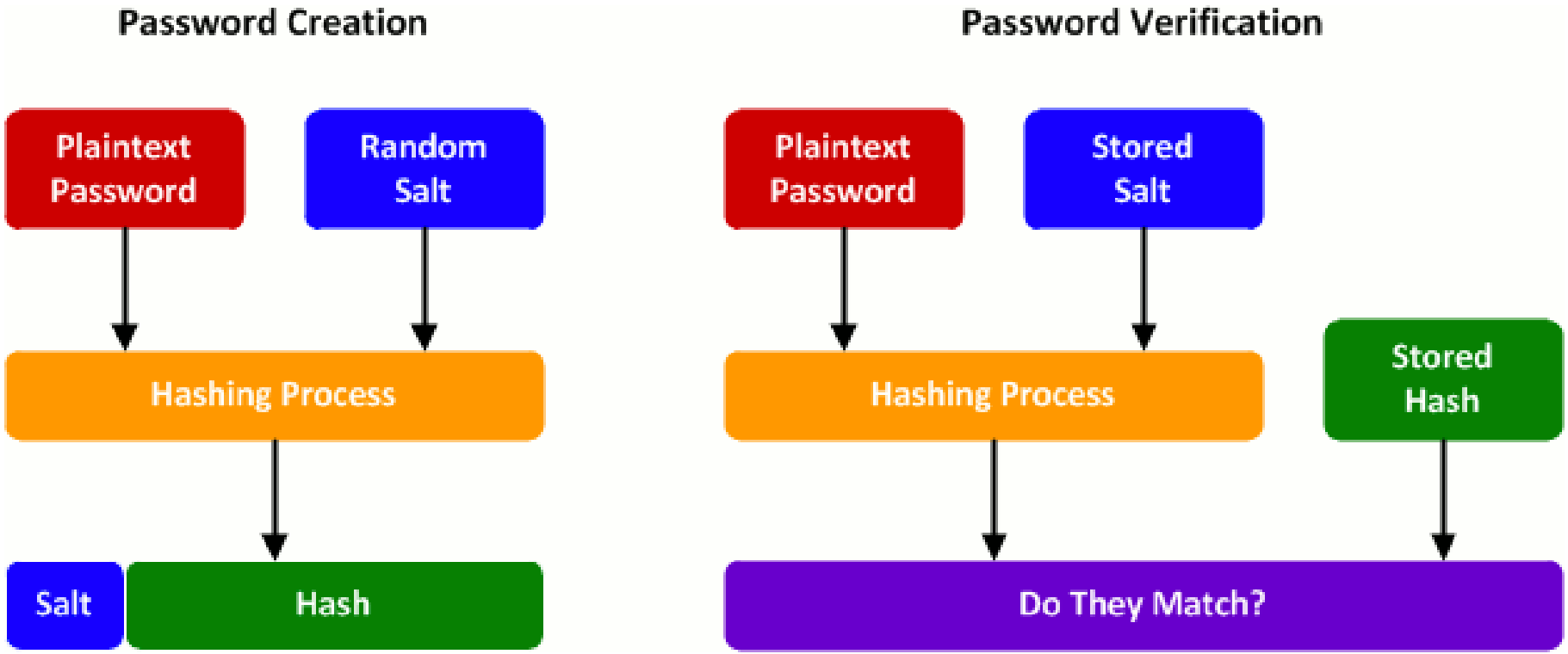
Salting

A Salt is a fixed-length cryptographically-strong random value that is added to a password as input to a hash function.

Dictionary attacks make passwords hashed with common algorithms vulnerable, salting reduces the effectiveness of dictionary attacks by making all input values for passwords unique.



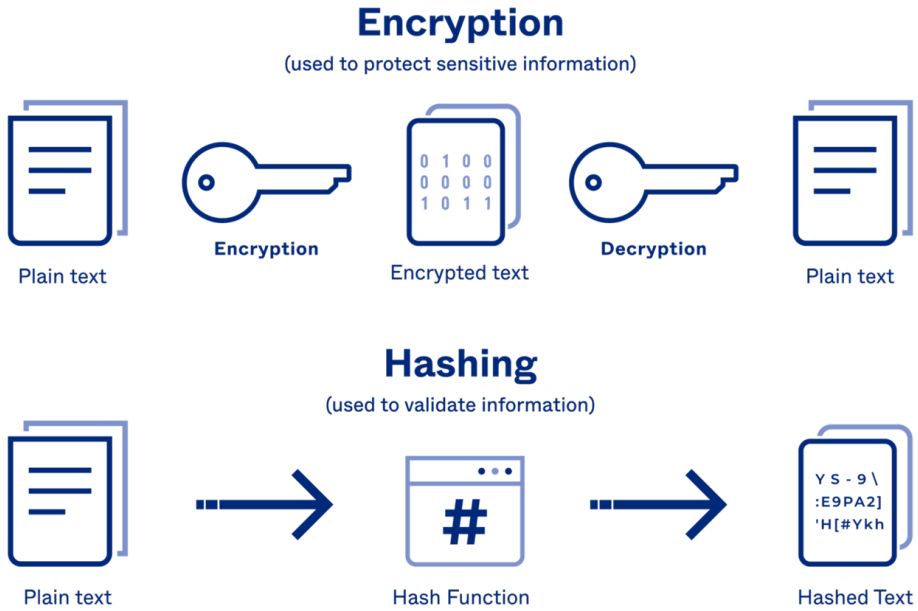
Password Salting



Encryption

Encryption is the most effective way to achieve data security. When data is sent between two parties or stored, it is stored in an encrypted non-human readable format that requires the key to properly decrypt and understand.

[OWASP Guide to Cryptography]



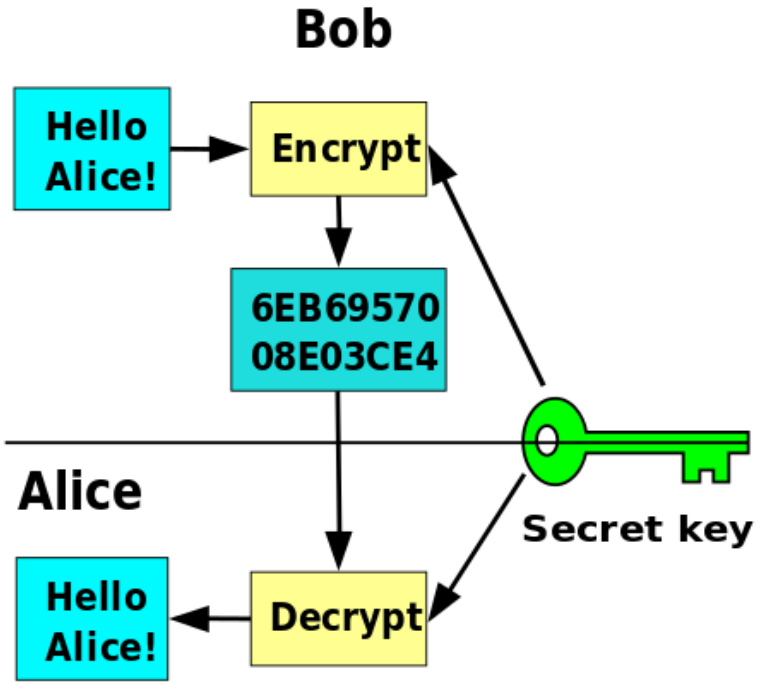
okta

Encrypting Data at Rest

1. Data at rest can use a form of encryption called **symmetric key encryption**
2. Requires both parties to use the key to encrypt and decrypt data.
3. Any party possessing the key can read the data.
4. Has difficulties securing the symmetric key amongst multiple parties.

Symmetric cryptography involves the parties sharing a common secret passphrase or key. Data is encrypted and decrypted using the same key.

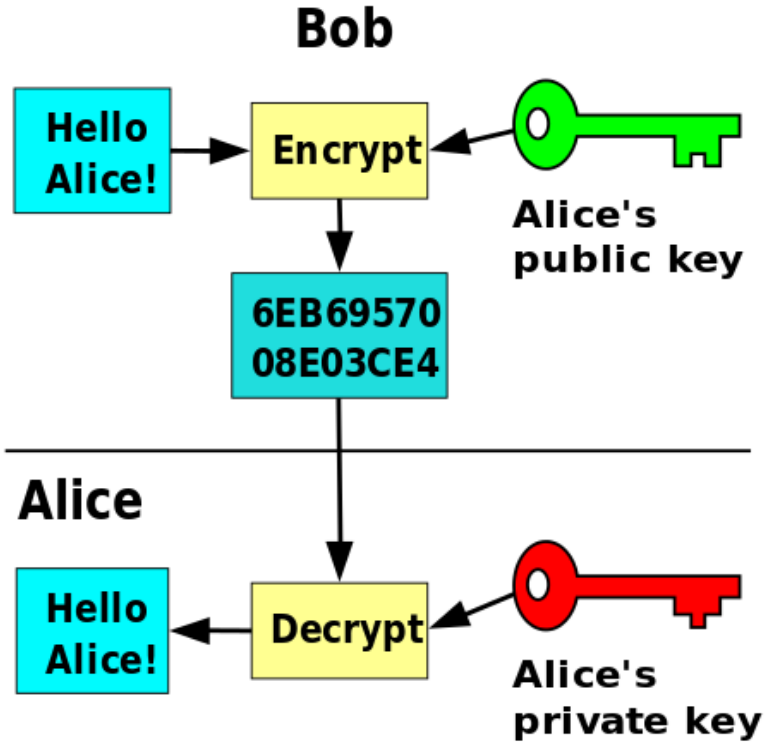
(https://www.techtarget.com/searchstorage/definition/data-at-rest)



Securing Data in Transit

1. Data in transit can use a form of encryption called **asymmetric key encryption**
2. Uses a Private and Public Key
3. Any party can be sent the public key. It can encrypt the data, but not decrypt it.
4. Only the key owner has the private key. It can decrypt data encrypted by the public key.
5. Has difficulties securing the symmetric key amongst multiple parties.

Asymmetric cryptography uses two keys: one to encrypt the data and the other to decrypt.



Asymmetric Encryption Common Usages

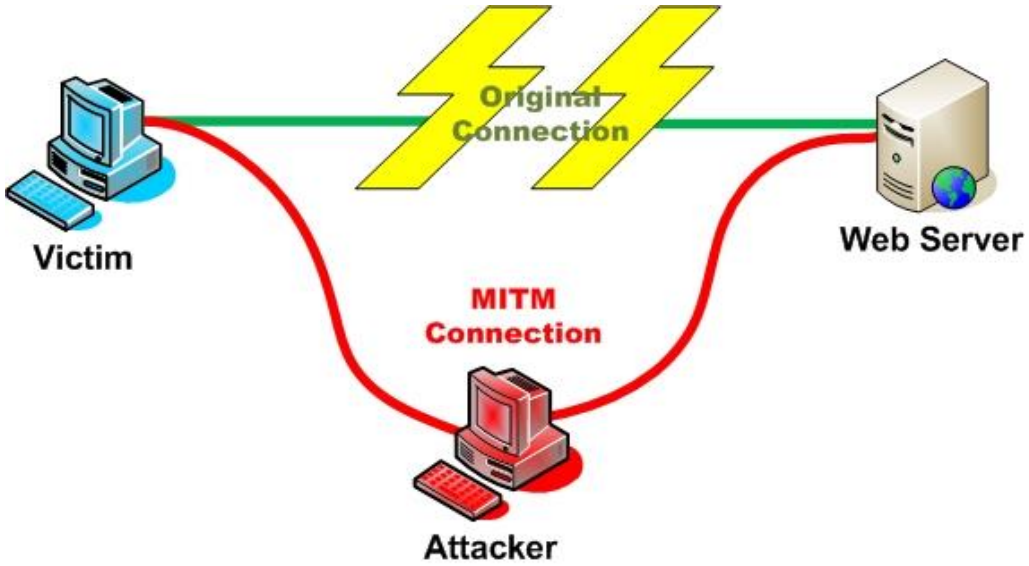
1. **Communication/Network Security:** Digital Certificates
    a. Normally a paid subscription
    b. OpenSSL - Open Source / Free Certificates
    c. **http://** (insecure) vs **https://** (secure)

2. **Web:** HTTPS (SSL (Secure Socket Layer) /TLS (Transport Layer Security))
    a. HTTPS Everywhere Project - movement to make all communication on the internet encrypted using OpenSSL

Man In the Middle Attack

Performed by a local malicious network connection, for example, in a coffee shop or hotel.

1. Attacker provides a fake wifi connection
2. Victim connects and establishes a secure connection with the fake wifi connection.
3. The attacker establishes a secure connection on behalf of the victim to the intended destination.
4. Communication then transmits data from the user to the attackers device and from the attackers device to the destination...



Let’s Code!