Daniel Beck

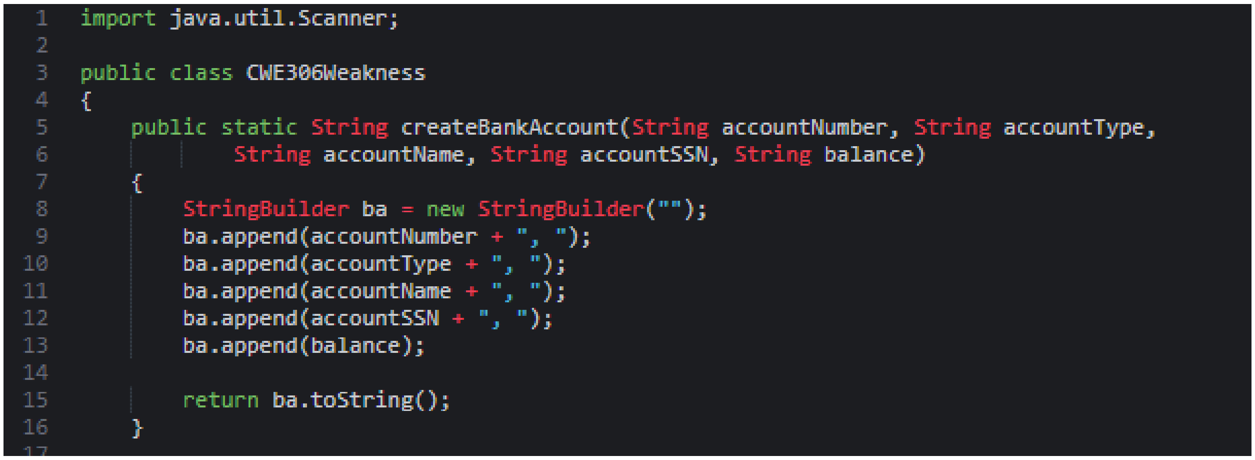
SDEV-325

September 28, 2020

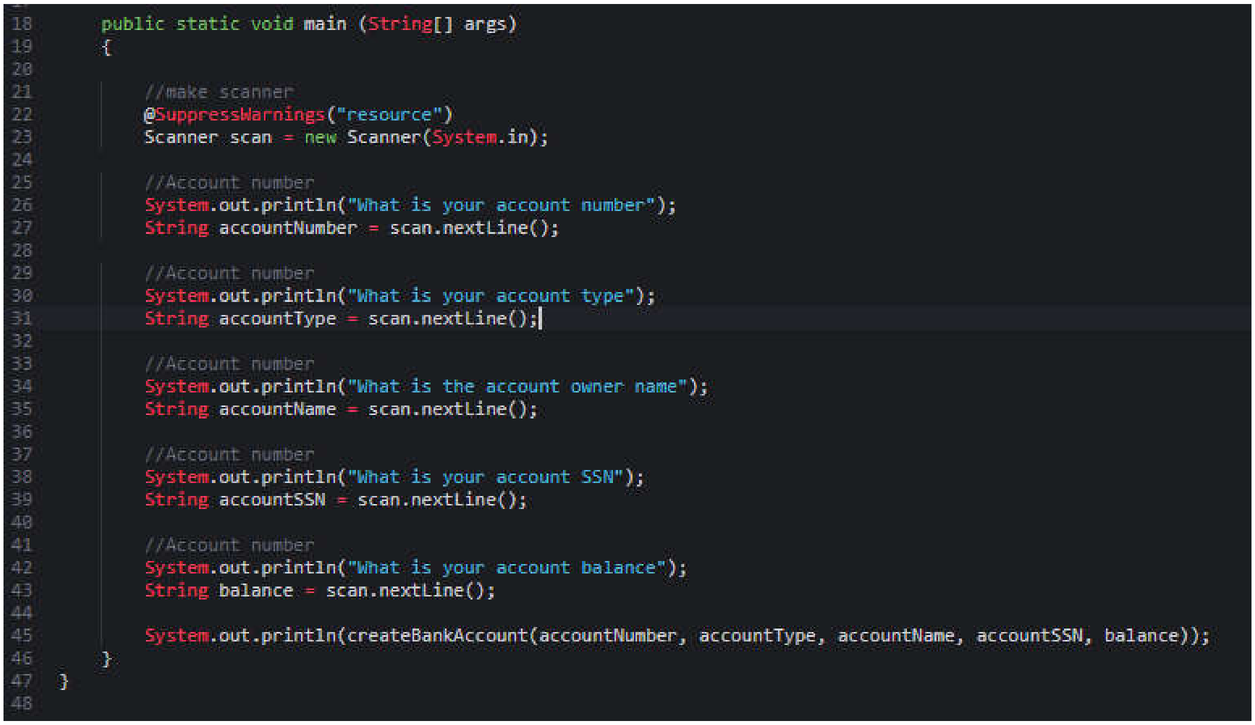
1. The first software vulnerability that was addressed was CWE-306: Missing Authentication for Critical Function. A CWE-306 vulnerability occurs when "the software does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources (*Common Weakness Enumeration,* 2020)." An example of where this vulnerability may be exposed is in software that accepts setting for a bank account. 1a shows a program that accepts information for a bank account. 1b shows the result of the weakness being mitigated by having the program run through a function that prompts the user for a password.

2. The second software vulnerability that was addressed was CWE-311: Missing Encryption of Sensitive Data. A CWE-311 vulnerability occurs when "the software does not encrypt sensitive or critical information before storage or transmission (*Common Weakness Enumeration,* 2020)." If users' passwords are not encrypted, there is a chance that their passwords may be exposed. 2a shows an example of a of what a user's password is stored as. 2b shows the mitigation of the vulnerability by using a python extension, cryptography.fernet, to use functions that encrypt passwords that are passed to them.

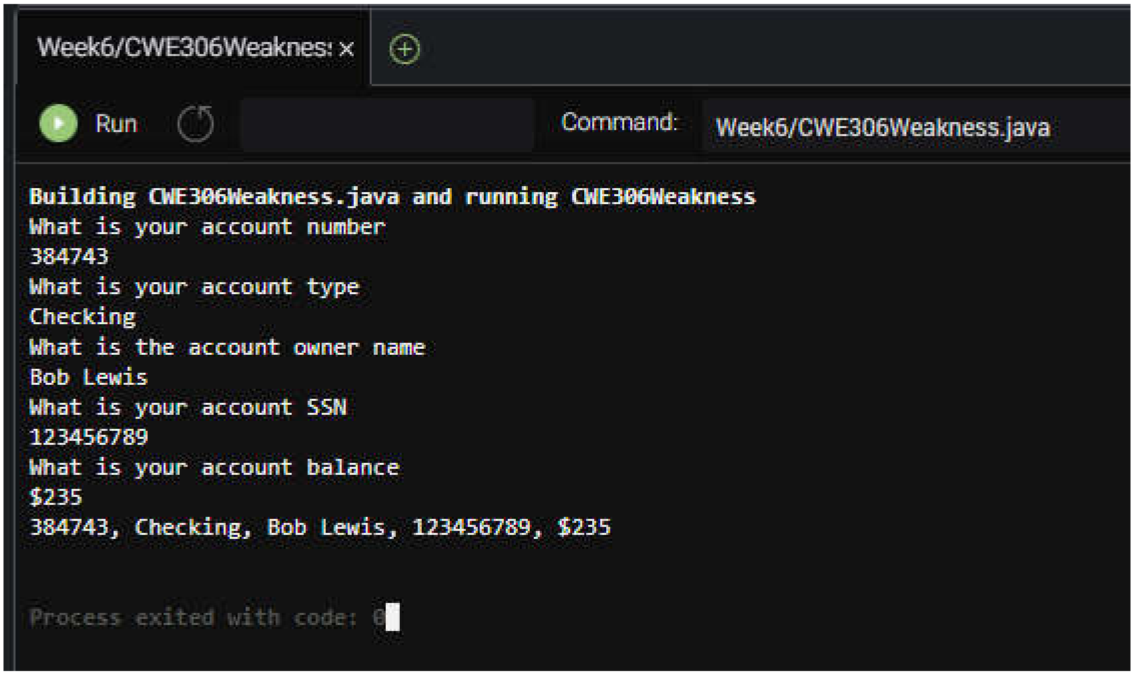
1a.



```java
import java.util.Scanner;

public class CWE306Weakness
{
    public static String createBankAccount(String accountNumber, String accountType,
            String accountName, String accountSSN, String balance)
    {
        StringBuilder ba = new StringBuilder("");
        ba.append(accountNumber + ", ");
        ba.append(accountType + ", ");
        ba.append(accountName + ", ");
        ba.append(accountSSN + ", ");
        ba.append(balance);

        return ba.toString();
    }
```

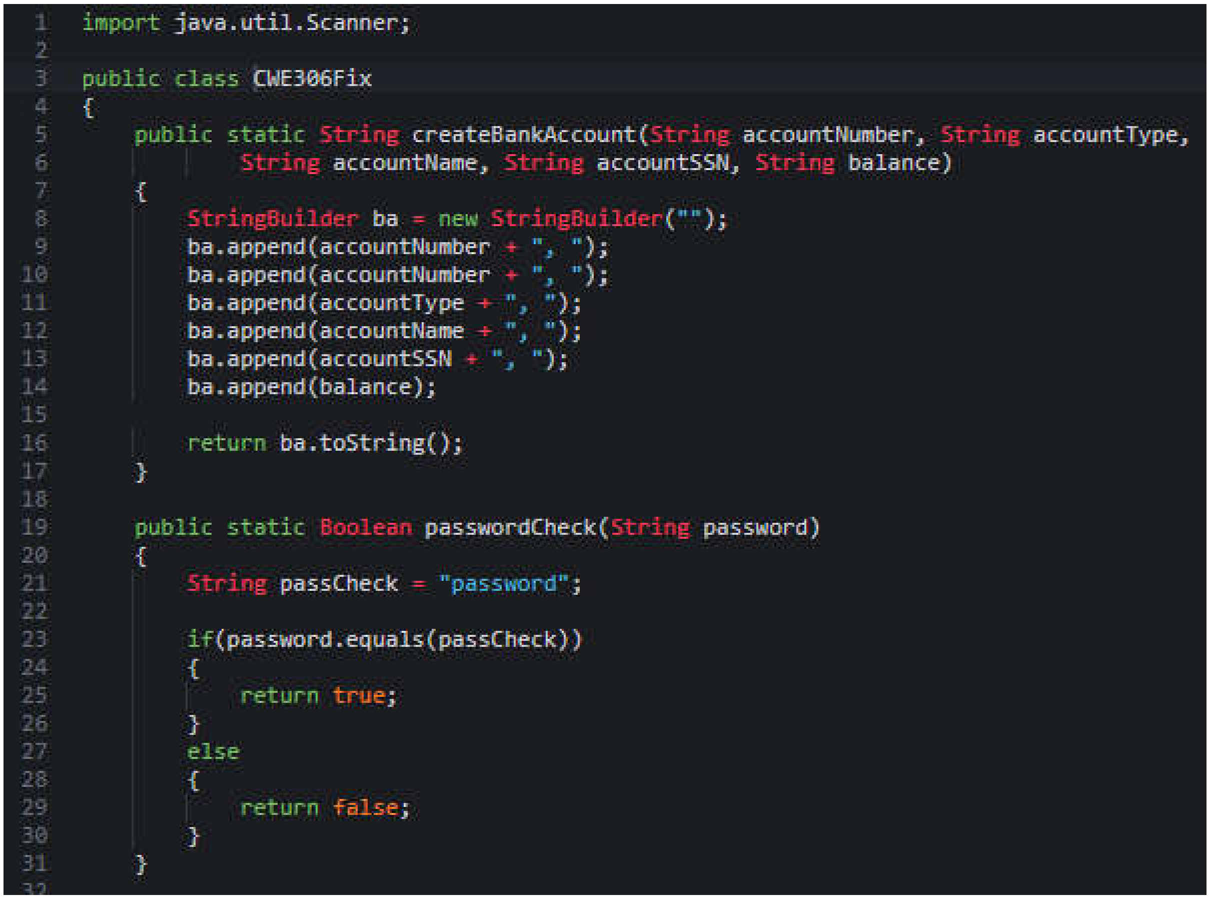


```java
    public static void main (String[] args)
    {

        //make scanner
        @SuppressWarnings("resource")
        Scanner scan = new Scanner(System.in);

        //Account number
        System.out.println("What is your account number");
        String accountNumber = scan.nextLine();

        //Account number
        System.out.println("What is your account type");
        String accountType = scan.nextLine();

        //Account number
        System.out.println("What is the account owner name");
        String accountName = scan.nextLine();

        //Account number
        System.out.println("What is your account SSN");
        String accountSSN = scan.nextLine();

        //Account number
        System.out.println("What is your account balance");
        String balance = scan.nextLine();

        System.out.println(createBankAccount(accountNumber, accountType, accountName, accountSSN, balance));
    }
}
```

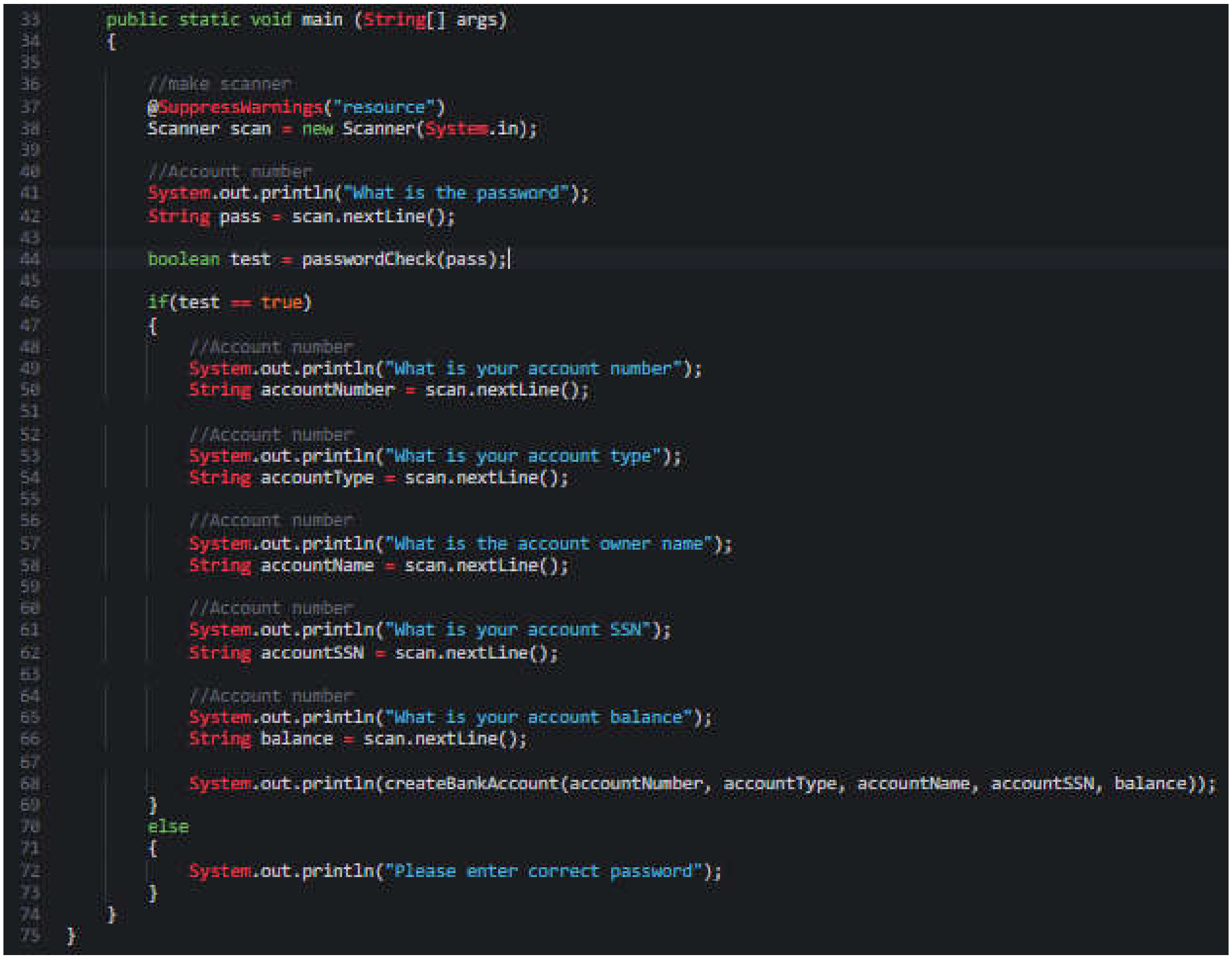


```
Building CWE306Weakness.java and running CWE306Weakness
What is your account number
384743
What is your account type
Checking
What is the account owner name
Bob Lewis
What is your account SSN
123456789
What is your account balance
$235
384743, Checking, Bob Lewis, 123456789, $235


Process exited with code: 0
```

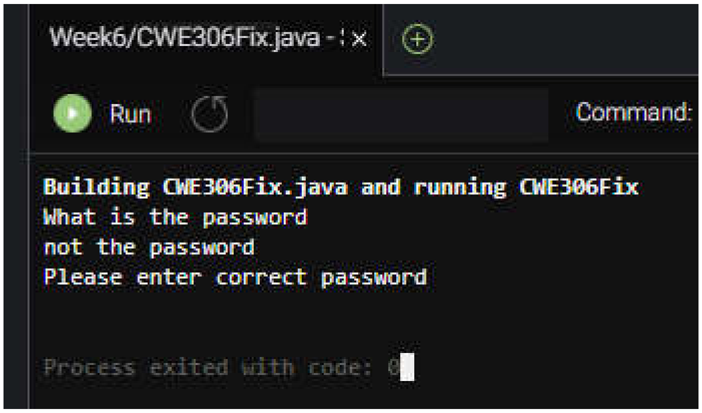
1b.



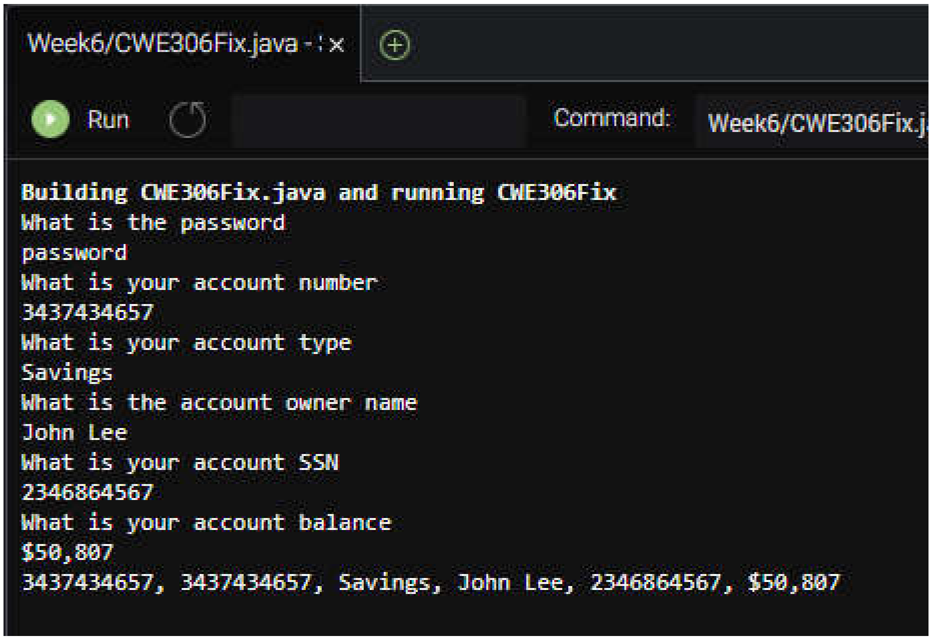
```java
import java.util.Scanner;

public class CWE306Fix
{
    public static String createBankAccount(String accountNumber, String accountType,
            String accountName, String accountSSN, String balance)
    {
        StringBuilder ba = new StringBuilder("");
        ba.append(accountNumber + ", ");
        ba.append(accountNumber + ", ");
        ba.append(accountType + ", ");
        ba.append(accountName + ", ");
        ba.append(accountSSN + ", ");
        ba.append(balance);

        return ba.toString();
    }

    public static Boolean passwordCheck(String password)
    {
        String passCheck = "password";

        if(password.equals(passCheck))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
```



```java
    public static void main (String[] args)
    {

        //make scanner
        @SuppressWarnings("resource")
        Scanner scan = new Scanner(System.in);

        //Account number
        System.out.println("What is the password");
        String pass = scan.nextLine();

        boolean test = passwordCheck(pass);

        if(test == true)
        {
            //Account number
            System.out.println("What is your account number");
            String accountNumber = scan.nextLine();

            //Account number
            System.out.println("What is your account type");
            String accountType = scan.nextLine();

            //Account number
            System.out.println("What is the account owner name");
            String accountName = scan.nextLine();

            //Account number
            System.out.println("What is your account SSN");
            String accountSSN = scan.nextLine();

            //Account number
            System.out.println("What is your account balance");
            String balance = scan.nextLine();

            System.out.println(createBankAccount(accountNumber, accountType, accountName, accountSSN, balance));
        }
        else
        {
            System.out.println("Please enter correct password");
        }
    }
}
```

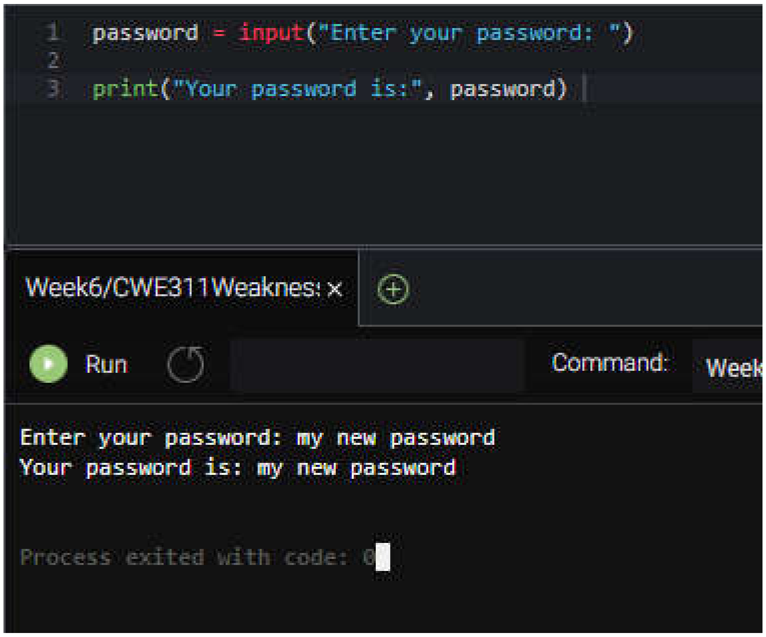


```
Building CWE306Fix.java and running CWE306Fix
What is the password
not the password
Please enter correct password


Process exited with code: 0
```



```
Building CWE306Fix.java and running CWE306Fix
What is the password
password
What is your account number
3437434657
What is your account type
Savings
What is the account owner name
John Lee
What is your account SSN
2346864567
What is your account balance
$50,807
3437434657, 3437434657, Savings, John Lee, 2346864567, $50,807
```

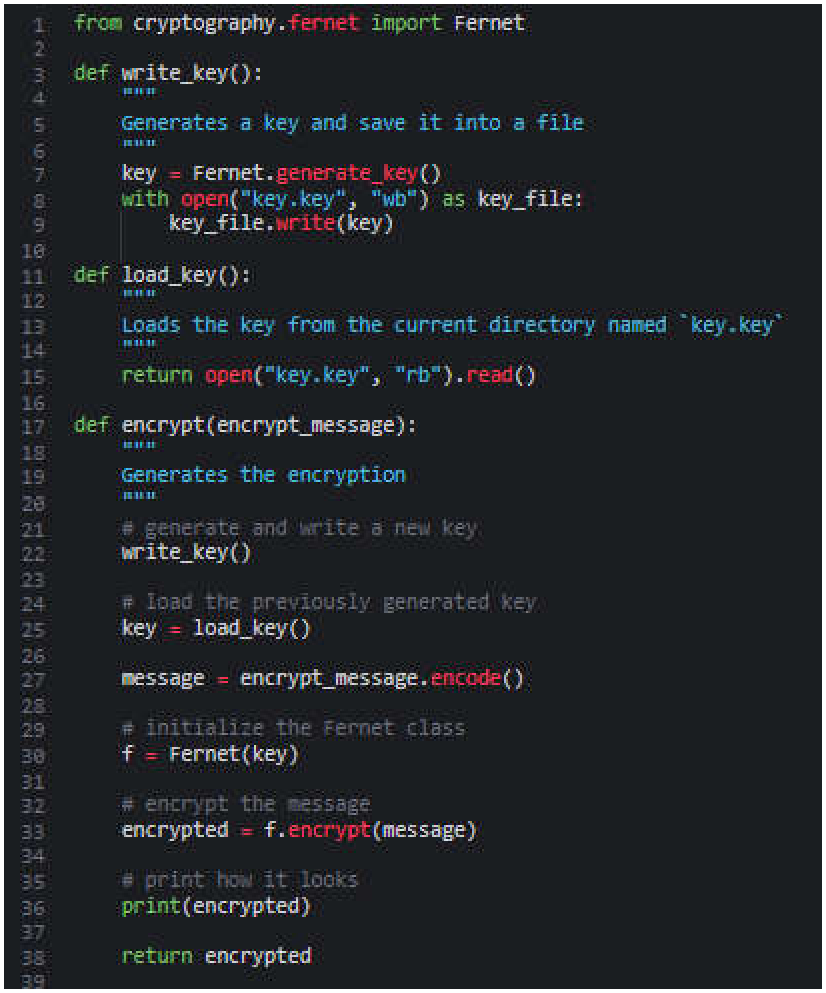
2a.



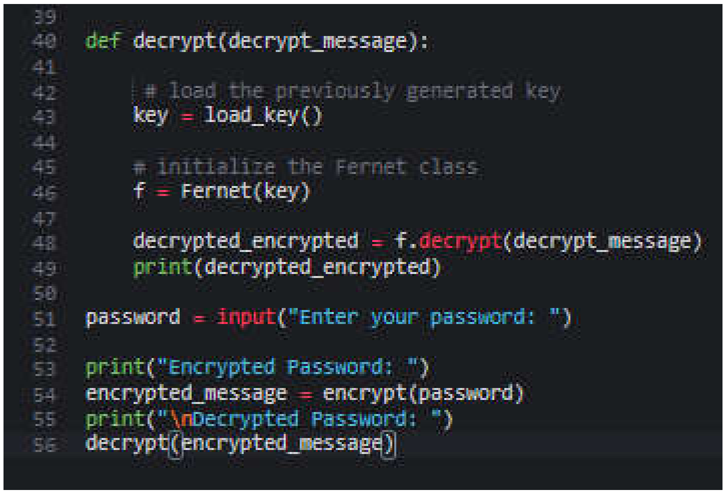
```
password = input("Enter your password: ")

print("Your password is:", password)
```

```
Enter your password: my new password
Your password is: my new password


Process exited with code: 0
```

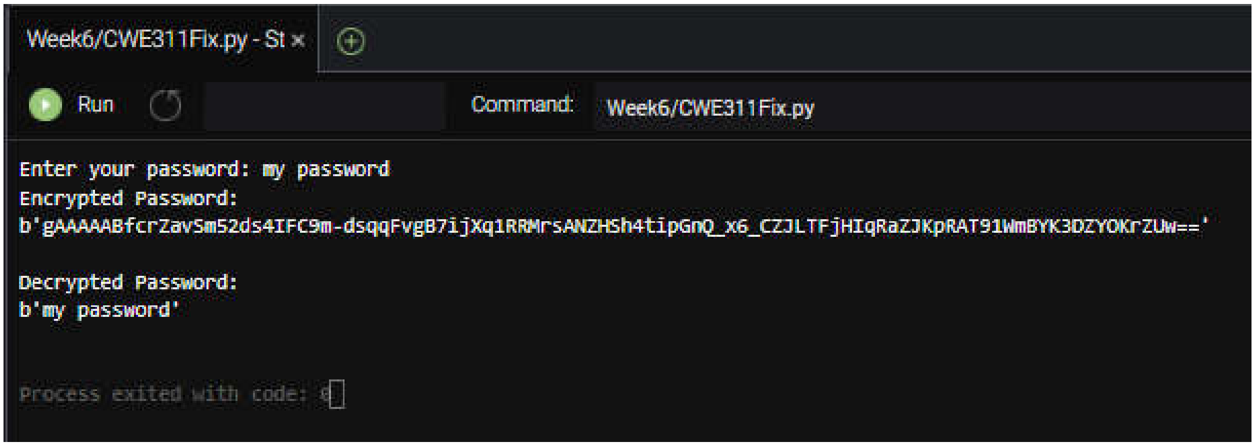
2b.



```python
from cryptography.fernet import Fernet

def write_key():
    """
    Generates a key and save it into a file
    """
    key = Fernet.generate_key()
    with open("key.key", "wb") as key_file:
        key_file.write(key)

def load_key():
    """
    Loads the key from the current directory named `key.key`
    """
    return open("key.key", "rb").read()

def encrypt(encrypt_message):
    """
    Generates the encryption
    """
    # generate and write a new key
    write_key()

    # load the previously generated key
    key = load_key()

    message = encrypt_message.encode()

    # initialize the Fernet class
    f = Fernet(key)

    # encrypt the message
    encrypted = f.encrypt(message)

    # print how it looks
    print(encrypted)

    return encrypted
```



```python
def decrypt(decrypt_message):

    # load the previously generated key
    key = load_key()

    # initialize the Fernet class
    f = Fernet(key)

    decrypted_encrypted = f.decrypt(decrypt_message)
    print(decrypted_encrypted)

password = input("Enter your password: ")

print("Encrypted Password: ")
encrypted_message = encrypt(password)
print("\nDecrypted Password: ")
decrypt(encrypted_message)
```



```
Enter your password: my password
Encrypted Password:
b'gAAAAABfcrZavSm52ds4IFC9m-dsqqFvgB7ijXq1RRMrsANZHSh4tipGnQ_x6_CZJLTFjHIqRaZJKpRAT91WmBYK3DZYOKrZUw=='

Decrypted Password:
b'my password'


Process exited with code: 0
```

References:

Common Weakness Enumeration. (2020, August 20). Retrieved September 29, 2020, from https://cwe.mitre.org/data/definitions/