# Test

Il y a 53 questions dans ce questionnaire.

## Information personelle

### Combien d'années d'expérience avez-vous en programmation? *

Répondre à cette question seulement si les conditions suivantes sont réunies :
rand(1, 2)

Veuillez sélectionner une réponse ci-dessous
Veuillez sélectionner une seule des propositions suivantes :

- ◯ moins d'un an
- ◯ 1 - 3 ans
- ◯ 3 - 5 ans
- ◯ 5 - 7 ans
- ◯ 7 ans ou plus

### Combien d'années d'expérience avez-vous en cyptographie? *

Veuillez sélectionner une réponse ci-dessous
Veuillez sélectionner une seule des propositions suivantes :

- ◯ moins d'un an
- ◯ 1 - 3 ans
- ◯ 3 - 5 ans
- ◯ 5 - 7 ans
- ◯ 7 ans ou plus

## Instructions

Chaque page contient le code qui sera à analysé.

Vous pouvez cliquez sur l'image pour accéder au code sous format texte.

Après votre analyse du code vous n'avez qu'à indiquer si vous avez trouvé quelque chose.

Si la réponse est oui, vous allez pouvoir brièvement expliquer votre découverte.

**Aucune des erreurs n'a rapport avec une erreur de syntaxe**

# Il y a-t-il une erreur dans ce code? (exStringUtils.java)

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal4.NAOK),rand(1,2),randVal4.NAOK)}**

```java
import java.lang.*;
import java.util.*;
import java.security.SecureRandom;
import org.apache.commons.lang3.RandomStringUtils;

public class PasswordGenerator {
    public String generateCommonLangPassword() {
        String upperCaseLetters = RandomStringUtils.random(2, 65, 90, true, true, new SecureRandom());
        String lowerCaseLetters = RandomStringUtils.random(2, 97, 122, true, true, new SecureRandom());
        String numbers = RandomStringUtils.randomNumeric(2);
        String specialChar = RandomStringUtils.random(2, 33, 47, false, false, new SecureRandom());
        String totalChars = RandomStringUtils.randomAlphanumeric(2);
        String combinedChars = upperCaseLetters.concat(lowerCaseLetters)
        .concat(numbers)
        .concat(specialChar)
        .concat(totalChars);
        List<Character> pwdChars = combinedChars.chars()
        .mapToObj(c -> (char) c)
        .collect(Collectors.toList());
        Collections.shuffle(pwdChars);
        String password = pwdChars.stream()
        .collect(StringBuilder::new, StringBuilder::append, StringBuilder::append)
        .toString();
        return password;
    }
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/exStringUtils.java)

STANDARD: Potentially Unsafe Code - RandomStringUtils

Line: 4 - D:\Question_code\exStringUtils.java

This class uses java.util.Random under the hood which can be bruteforced, use java.security.SecureRandom.

import org.apache.commons.lang3.RandomStringUtils;

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 6 - D:\Question_code\exStringUtils.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and

practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class PasswordCenerator {

STANDARD: Potentially Unsafe Code - RandomStringUtils

Line: 8 - D:\Question_code\exStringUtils.java

This class uses java.util.Random under the hood which can be bruteforced, use java.security.SecureRandom.

String upperCaseLetters = RandomStringUtils.random(2, 65, 90, true, true, new SecureRandom());

STANDARD: Potentially Unsafe Code - RandomStringUtils

Line: 9 - D:\Question_code\exStringUtils.java

This class uses java.util.Random under the hood which can be bruteforced, use java.security.SecureRandom.

String lowerCaseLetters = RandomStringUtils.random(2, 97, 122, true, true, new SecureRandom());

STANDARD: Potentially Unsafe Code - RandomStringUtils

Line: 10 - D:\Question_code\exStringUtils.java

This class uses java.util.Random under the hood which can be bruteforced, use java.security.SecureRandom.

String numbers = RandomStringUtils.randomNumeric(2);

STANDARD: Potentially Unsafe Code - RandomStringUtils

Line: 11 - D:\Question_code\exStringUtils.java

This class uses java.util.Random under the hood which can be bruteforced, use java.security.SecureRandom.

String specialChar = RandomStringUtils.random(2, 33, 47, false, false, new SecureRandom());

STANDARD: Potentially Unsafe Code - RandomStringUtils

Line: 12 - D:\Question_code\exStringUtils.java

This class uses java.util.Random under the hood which can be bruteforced, use java.security.SecureRandom.

String totalChars = RandomStringUtils.randomAlphanumeric(2);

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal4.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/507/qid/16193) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.

```java
import java.lang.*;
import java.util.*;
import java.security.SecureRandom;
import org.apache.commons.lang3.RandomStringUtils;

public class PasswordGenerator {
    public String generateCommonLangPassword() {
        String upperCaseLetters = RandomStringUtils.random(2, 65, 90, true, true, new SecureRandom());
        String lowerCaseLetters = RandomStringUtils.random(2, 97, 122, true, true, new SecureRandom());
        String numbers = RandomStringUtils.randomNumeric(2);
        String specialChar = RandomStringUtils.random(2, 33, 47, false, false, new SecureRandom());
        String totalChars = RandomStringUtils.randomAlphanumeric(2);
        String combinedChars = upperCaseLetters.concat(lowerCaseLetters)
        .concat(numbers)
        .concat(specialChar)
        .concat(totalChars);
        List<Character> pwdChars = combinedChars.chars()
        .mapToObj(c -> (char) c)
        .collect(Collectors.toList());
        Collections.shuffle(pwdChars);
        String password = pwdChars.stream()
        .collect(StringBuilder::new, StringBuilder::append, StringBuilder::append)
        .toString();
        return password;
    }
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/exStringUtils.java)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal4.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/507/qid/16193) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

- ◯ oui
- ◯ non

## Donner une courte description de l'erreur trouvé. *

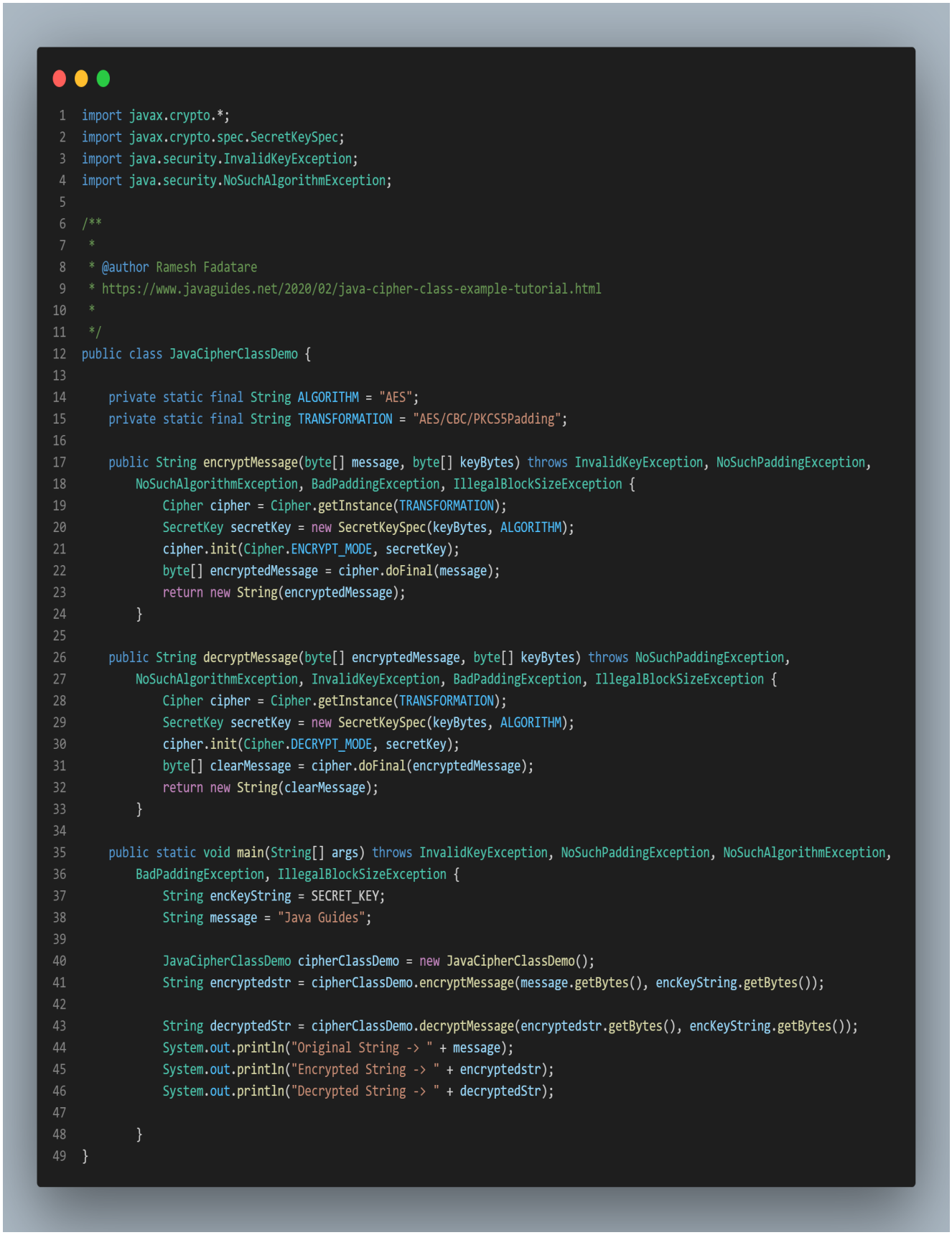
Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r159q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

# Il y a-t-il une erreur dans ce code? (exCBC.java)

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal8.NAOK),rand(1,2),randVal8.NAOK)}**



```java
import javax.crypto.*;
import javax.crypto.spec.SecretKeySpec;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;

/**
 *
 * @author Ramesh Fadatare
 * https://www.javaguides.net/2020/02/java-cipher-class-example-tutorial.html
 *
 */
public class JavaCipherClassDemo {

    private static final String ALGORITHM = "AES";
    private static final String TRANSFORMATION = "AES/CBC/PKCS5Padding";

    public String encryptMessage(byte[] message, byte[] keyBytes) throws InvalidKeyException, NoSuchPaddingException,
        NoSuchAlgorithmException, BadPaddingException, IllegalBlockSizeException {
            Cipher cipher = Cipher.getInstance(TRANSFORMATION);
            SecretKey secretKey = new SecretKeySpec(keyBytes, ALGORITHM);
            cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            byte[] encryptedMessage = cipher.doFinal(message);
            return new String(encryptedMessage);
        }

    public String decryptMessage(byte[] encryptedMessage, byte[] keyBytes) throws NoSuchPaddingException,
        NoSuchAlgorithmException, InvalidKeyException, BadPaddingException, IllegalBlockSizeException {
            Cipher cipher = Cipher.getInstance(TRANSFORMATION);
            SecretKey secretKey = new SecretKeySpec(keyBytes, ALGORITHM);
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            byte[] clearMessage = cipher.doFinal(encryptedMessage);
            return new String(clearMessage);
        }

    public static void main(String[] args) throws InvalidKeyException, NoSuchPaddingException, NoSuchAlgorithmException,
        BadPaddingException, IllegalBlockSizeException {
            String encKeyString = SECRET_KEY;
            String message = "Java Guides";

            JavaCipherClassDemo cipherClassDemo = new JavaCipherClassDemo();
            String encryptedstr = cipherClassDemo.encryptMessage(message.getBytes(), encKeyString.getBytes());

            String decryptedStr = cipherClassDemo.decryptMessage(encryptedstr.getBytes(), encKeyString.getBytes());
            System.out.println("Original String -> " + message);
            System.out.println("Encrypted String -> " + encryptedstr);
            System.out.println("Decrypted String -> " + decryptedStr);

        }
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/exCBC.java)

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 12 - D:\Question_code\exCBC.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

public class JavaCipherClassDemo {

MEDIUM: Potentially Unsafe Code - .getInstance(

Line: 19 - D:\Question_code\exCBC.java

This class defaults to ECB mode, which is unsafe.

Cipher cipher = Cipher.getInstance(TRANSFORMATION);

MEDIUM: Potentially Unsafe Code - .getInstance(

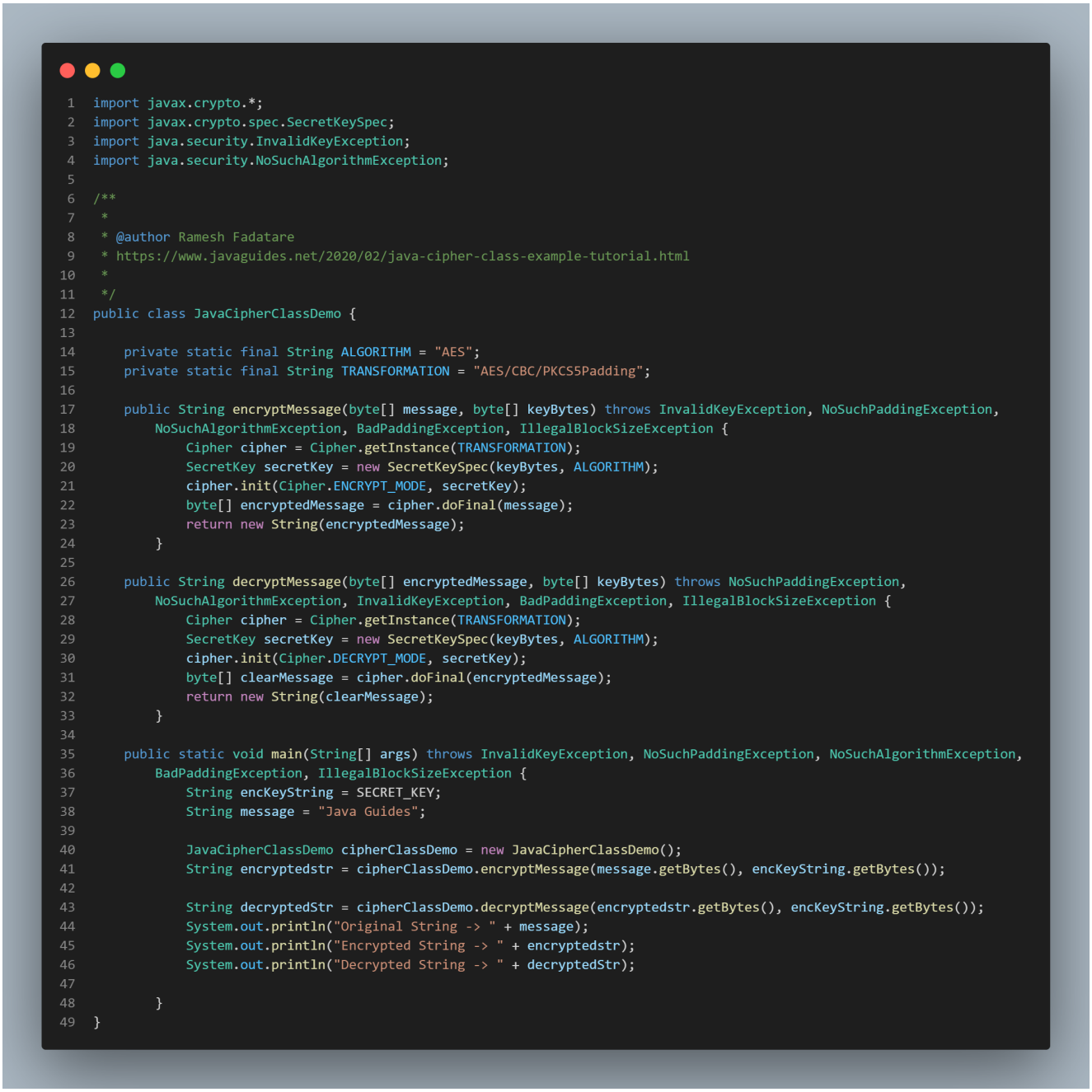
Line: 28 - D:\Question_code\exCBC.java

This class defaults to ECB mode, which is unsafe.

Cipher cipher = Cipher.getInstance(TRANSFORMATION);

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal8.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/511/qid/16213) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.



```java
import javax.crypto.*;
import javax.crypto.spec.SecretKeySpec;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;

/**
 *
 * @author Ramesh Fadatare
 * https://www.javaguides.net/2020/02/java-cipher-class-example-tutorial.html
 *
 */
public class JavaCipherClassDemo {

    private static final String ALGORITHM = "AES";
    private static final String TRANSFORMATION = "AES/CBC/PKCS5Padding";

    public String encryptMessage(byte[] message, byte[] keyBytes) throws InvalidKeyException, NoSuchPaddingException,
        NoSuchAlgorithmException, BadPaddingException, IllegalBlockSizeException {
            Cipher cipher = Cipher.getInstance(TRANSFORMATION);
            SecretKey secretKey = new SecretKeySpec(keyBytes, ALGORITHM);
            cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            byte[] encryptedMessage = cipher.doFinal(message);
            return new String(encryptedMessage);
        }

    public String decryptMessage(byte[] encryptedMessage, byte[] keyBytes) throws NoSuchPaddingException,
        NoSuchAlgorithmException, InvalidKeyException, BadPaddingException, IllegalBlockSizeException {
            Cipher cipher = Cipher.getInstance(TRANSFORMATION);
            SecretKey secretKey = new SecretKeySpec(keyBytes, ALGORITHM);
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            byte[] clearMessage = cipher.doFinal(encryptedMessage);
            return new String(clearMessage);
        }

    public static void main(String[] args) throws InvalidKeyException, NoSuchPaddingException, NoSuchAlgorithmException,
        BadPaddingException, IllegalBlockSizeException {
            String encKeyString = SECRET_KEY;
            String message = "Java Guides";

            JavaCipherClassDemo cipherClassDemo = new JavaCipherClassDemo();
            String encryptedstr = cipherClassDemo.encryptMessage(message.getBytes(), encKeyString.getBytes());

            String decryptedStr = cipherClassDemo.decryptMessage(encryptedstr.getBytes(), encKeyString.getBytes());
            System.out.println("Original String -> " + message);
            System.out.println("Encrypted String -> " + encryptedstr);
            System.out.println("Decrypted String -> " + decryptedStr);

        }
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/exCBC.java)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal8.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/511/qid/16213) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

- ◯ oui
- ◯ non

## Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r739q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

# Il y a-t-il une erreur dans ce code?

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal1.NAOK),rand(1,2),randVal1.NAOK)}**

```cpp
# include <iostream>
using namespace std;
# include "Hashtable.h"
# include <vector>

Hashtable:: Hashtable()
{
    start = nullptr;
}
void Hashtable:: starthash()
{
    for (int i = 0; i < 12; i++)
    {
        Node * temp1 = new Node(i);
        if (start == nullptr)
        {
            start = temp1;
        }
        else
        {
            Node * current = start;
            while (current ->next != nullptr)
            {
                current = current->next;
            }
            current->next = temp1;
        }
    }
    loadhashtable();
}
void Hashtable::add(int a, int p)
{
    static int i = 0;
    ofstream write;
    write.open("hashtable.txt",ios::app);
    if (i != 0)
    {
        write << endl;
        write << a << endl << p;
    }
    else
    {
        i++;
        write << a << endl << p;
    }
    write.close();

    starthash();
}
bool Hashtable::match(int a, int p)
{
    bool flag = false;
    int r = a % 10;
    Node * c = start;
    while (c->data != r)
    {
        c = c->next;
    }
    Node_1 *c1 = c->pre;
    while (c1 != nullptr)
    {
        if (c1->accountNumber == a && c1->password == p)
        {
            flag = true;
            break;
        }
        c1 = c1->next;
    }
    return flag;
}
void Hashtable:: display()
```

```
void Hashtable:: display()
{
    Node * current = start;
    while (current != nullptr)
    {
        cout<<current->data<<endl;
        current = current->next;
    }
}
void  Hashtable::loadhashtable()
{
    int acc = 0, r, pass;

    ifstream read;
    read.open("hashtable.txt");
    while (!read.eof())
    {

        read >> acc;
        read >> pass;
        if (match(acc, pass))
        {
            continue;
        }
        if (acc!= -858993460 && pass!= -858993460)
        {
            r = acc % 10;

            Node * c = start;
            while (c->data != r)
            {
                c = c->next;
            }
            Node_1 *md5 = new Node_1(acc, pass);
            if (c->pre == nullptr)
            {
                c->pre = md5;
            }
            else
            {
                Node_1 *root;
                root = c->pre;
                while (root->next != nullptr)
                {
                    root = root->next;
                }
                root->next = md5;
            }
        }
        else
        {
            cout << "NO password present" << endl;
        }
    }
    read.close();
}
void  Hashtable::displayPasswords()
{
    starthash();
    Node *c = start;
    while (c != nullptr)
    {
        Node_1 *c1 = c->pre;
        while (c1 != nullptr)
        {
            cout<<c1->accountNumber<<endl;
            cout<<c1->password<<endl<<endl;
            c1 = c1->next;
        }
        c = c->next;
    }
}
void  Hashtable:: delete_password(int accountno)
{
    ifstream read;
    read.open("hashtable.txt");
    vector <int> v;
    int acc=0,pass=0;
```

```
    int i = 0;
    while (!read.eof())
    {
        i++;
        read >> acc;
        read >> pass;
        if (acc == accountno)
        {                                          // read both account number and password to skip them
            continue;
        }
        v.push_back(acc);
        v.push_back(pass);
    }
    read.close();
    ofstream write;
    write.open("temp.txt", ios::app);

        for (int i = 0; i < v.size(); i++)
        {
            if (v[i] != 0)
            {
                write << v[i] << endl;
            }
        }


    write.close();
    remove("hashtable.txt");
    rename("temp.txt", "hashtable.txt");
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/Example.cpp)

MEDIUM: Potentially Unsafe Code - md5
Line: 104 - D:\Question_code\Example.cpp
MD5 is considered cryptographically insecure
Node_1 *md5 = new Node_1(acc, pass);

MEDIUM: Potentially Unsafe Code - md5
Line: 107 - D:\Question_code\Example.cpp
MD5 is considered cryptographically insecure
c->pre = md5;

MEDIUM: Potentially Unsafe Code - md5
Line: 117 - D:\Question_code\Example.cpp
MD5 is considered cryptographically insecure
root->next = md5;

STANDARD: Potential Memory Mis-management. Variable Name: temp1

new without delete.
Line: 14 FileName: D:\Question_code\Example.cpp

STANDARD: Potential Memory Mis-management. Variable Name: md5
new without delete.
Line: 14 FileName: D:\Question_code\Example.cpp

new without delete.
Line: 104 FileName: D:\Question_code\Example.cpp

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal1.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/504/qid/16178) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.

```cpp
# include <iostream>
using namespace std;
# include "Hashtable.h"
# include <vector>

Hashtable:: Hashtable()
{
    start = nullptr;
}
void Hashtable:: starthash()
{
    for (int i = 0; i < 12; i++)
    {
        Node * temp1 = new Node(i);
        if (start == nullptr)
        {
            start = temp1;
        }
        else
        {
            Node * current = start;
            while (current ->next != nullptr)
            {
                current = current->next;
            }
            current->next = temp1;
        }
    }
    loadhashtable();
}
void Hashtable::add(int a, int p)
{
    static int i = 0;
    ofstream write;
    write.open("hashtable.txt",ios::app);
    if (i != 0)
    {
        write << endl;
        write << a << endl << p;
    }
    else
    {
        i++;
        write << a << endl << p;
    }
    write.close();

    starthash();
}
bool Hashtable::match(int a, int p)
{
    bool flag = false;
    int r = a % 10;
    Node * c = start;
    while (c->data != r)
    {
        c = c->next;
    }
    Node_1 *c1 = c->pre;
    while (c1 != nullptr)
    {
        if (c1->accountNumber == a && c1->password == p)
        {
            flag = true;
            break;
        }
        c1 = c1->next;
    }
    return flag;
}
void Hashtable:: display()
```

```
void Hashtable:: display()
{
    Node * current = start;
    while (current != nullptr)
    {
        cout<<current->data<<endl;
        current = current->next;
    }
}
void  Hashtable::loadhashtable()
{
    int acc = 0, r, pass;

    ifstream read;
    read.open("hashtable.txt");
    while (!read.eof())
    {

        read >> acc;
        read >> pass;
        if (match(acc, pass))
        {
            continue;
        }
        if (acc!= -858993460 && pass!= -858993460)
        {
            r = acc % 10;

            Node * c = start;
            while (c->data != r)
            {
                c = c->next;
            }
            Node_1 *md5 = new Node_1(acc, pass);
            if (c->pre == nullptr)
            {
                c->pre = md5;
            }
            else
            {
                Node_1 *root;
                root = c->pre;
                while (root->next != nullptr)
                {
                    root = root->next;
                }
                root->next = md5;
            }
        }
        else
        {
            cout << "NO password present" << endl;
        }
    }
    read.close();
}
void  Hashtable::displayPasswords()
{
    starthash();
    Node *c = start;
    while (c != nullptr)
    {
        Node_1 *c1 = c->pre;
        while (c1 != nullptr)
        {
            cout<<c1->accountNumber<<endl;
            cout<<c1->password<<endl<<endl;
            c1 = c1->next;
        }
        c = c->next;
    }
}
void  Hashtable:: delete_password(int accountno)
{
    ifstream read;
    read.open("hashtable.txt");
    vector <int> v;
    int acc=0,pass=0;
```

```
    int i = 0;
    while (!read.eof())
    {
        i++;
        read >> acc;
        read >> pass;
        if (acc == accountno)
        {                                          // read both account number and password to skip them
            continue;
        }
        v.push_back(acc);
        v.push_back(pass);
    }
    read.close();
    ofstream write;
    write.open("temp.txt", ios::app);

        for (int i = 0; i < v.size(); i++)
        {
            if (v[i] != 0)
            {
                write << v[i] << endl;
            }
        }


    write.close();
    remove("hashtable.txt");
    rename("temp.txt", "hashtable.txt");
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/Example.cpp)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal1.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/504/qid/16178) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

- ◯ oui
- ◯ non

## Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r821q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

# Il y a-t-il une erreur dans ce code? (enc.php)

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal2.NAOK),rand(1,2),randVal2.NAOK)}**

```php
<?php
// https://github.com/defuse/php-encryption/blob/master/docs/Tutorial.md
use Defuse\Crypto\Crypto;
use Defuse\Crypto\KeyProtectedByPassword;
use Defuse\Crypto\Key;

$user_key_encoded = // ... get it out of the cookie ...
$user_locked_key = KeyProtectedByPassword::loadFromAsciiSafeString($user_key_encoded)
$user_key = $user_locked_key->unlockKey($user_key_encoded);

// ...

$credit_card_number = // ... get credit card number from the user
$encrypted_card_number = Crypto::encrypt($credit_card_number, $user_key);
?>
```

(https://github.com/2longAGO/Projet_Synth/blob/main/enc.php)

MEDIUM: Potentially Unsafe Code - KeyProtectedByPassword

Line: 4 - D:\Question_code\enc.php

This class is vulnerable because it save and check the password in the format sha256($password): https://github.com/defuse/php-encryption/issues/392

use Defuse\Crypto\KeyProtectedByPassword;

MEDIUM: Potentially Unsafe Code - KeyProtectedByPassword

Line: 8 - D:\Question_code\enc.php

This class is vulnerable because it save and check the password in the format sha256($password): https://github.com/defuse/php-encryption/issues/392

$user_locked_key = KeyProtectedByPassword::loadFromAsciiSafeString($user_key_encoded)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal2.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/513/qid/16223) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.

```php
<?php
// https://github.com/defuse/php-encryption/blob/master/docs/Tutorial.md
use Defuse\Crypto\Crypto;
use Defuse\Crypto\KeyProtectedByPassword;
use Defuse\Crypto\Key;

$user_key_encoded = // ... get it out of the cookie ...
$user_locked_key = KeyProtectedByPassword::loadFromAsciiSafeString($user_key_encoded)
$user_key = $user_locked_key->unlockKey($user_key_encoded);

// ...

$credit_card_number = // ... get credit card number from the user
$encrypted_card_number = Crypto::encrypt($credit_card_number, $user_key);
?>
```

(https://github.com/2longAGO/Projet_Synth/blob/main/enc.php)

Répondre à cette question seulement si les conditions suivantes sont réunies : ((randVal2.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/513/qid/16223) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

- ◯ oui
- ◯ non

## Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r189q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

# Il y a-t-il une erreur dans ce code? (crypto_user.c)

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal3.NAOK),rand(1,2),randVal3.NAOK)}**

```c
#include <linux/module.h>
#include <linux/crypto.h>
#include <linux/cryptouser.h>
#include <linux/sched.h>
#include <net/netlink.h>
#include <linux/security.h>
#include <net/net_namespace.h>
#include <crypto/internal/aead.h>
#include <crypto/internal/skcipher.h>

#include "internal.h"

static DEFINE_MUTEX(crypto_cfg_mutex);

/* The crypto netlink socket */
static struct sock *crypto_nlsk;

struct crypto_dump_info {
    struct sk_buff *in_skb;
    struct sk_buff *out_skb;
    u32 nlmsg_seq;
    u16 nlmsg_flags;
};

static struct crypto_alg *crypto_alg_match(struct crypto_user_alg *p, int exact)
{
    struct crypto_alg *q, *alg = NULL;

    down_read(&crypto_alg_sem);

    list_for_each_entry(q, &crypto_alg_list, cra_list) {
        int match = 0;

        if ((q->cra_flags ^ p->cru_type) & p->cru_mask)
            continue;

        if (strlen(p->cru_driver_name))
            match = !strcmp(q->cra_driver_name,
                    p->cru_driver_name);
        else if (!exact)
            match = !strcmp(q->cra_name, p->cru_name);

        if (match) {
            alg = q;
            break;
        }
    }

    up_read(&crypto_alg_sem);

    return alg;
}

static int crypto_report_cipher(struct sk_buff *skb, struct crypto_alg *alg)
```

```
{
    struct crypto_report_cipher rcipher;

    snprintf(rcipher.type, CRYPTO_MAX_ALG_NAME, "%s", "cipher");

    rcipher.blocksize = alg->cra_blocksize;
    rcipher.min_keysize = alg->cra_cipher.cia_min_keysize;
    rcipher.max_keysize = alg->cra_cipher.cia_max_keysize;

    if (nla_put(skb, CRYPTOCFGA_REPORT_CIPHER,
            sizeof(struct crypto_report_cipher), &rcipher))
        goto nla_put_failure;
    return 0;

nla_put_failure:
    return -EMSGSIZE;
}

static int crypto_report_comp(struct sk_buff *skb, struct crypto_alg *alg)
{
    struct crypto_report_comp rcomp;

    snprintf(rcomp.type, CRYPTO_MAX_ALG_NAME, "%s", "compression");

    if (nla_put(skb, CRYPTOCFGA_REPORT_COMPRESS,
            sizeof(struct crypto_report_comp), &rcomp))
        goto nla_put_failure;
    return 0;

nla_put_failure:
    return -EMSGSIZE;
}

static int crypto_report_one(struct crypto_alg *alg,
                 struct crypto_user_alg *ualg, struct sk_buff *skb)
{
    memcpy(&ualg->cru_name, &alg->cra_name, sizeof(ualg->cru_name));
    memcpy(&ualg->cru_driver_name, &alg->cra_driver_name,
           sizeof(ualg->cru_driver_name));
    memcpy(&ualg->cru_module_name, module_name(alg->cra_module),
           CRYPTO_MAX_ALG_NAME);

    ualg->cru_flags = alg->cra_flags;
    ualg->cru_refcnt = atomic_read(&alg->cra_refcnt);

    if (nla_put_u32(skb, CRYPTOCFGA_PRIORITY_VAL, alg->cra_priority))
        goto nla_put_failure;
    if (alg->cra_flags & CRYPTO_ALG_LARVAL) {
        struct crypto_report_larval rl;

        snprintf(rl.type, CRYPTO_MAX_ALG_NAME, "%s", "larval");

        if (nla_put(skb, CRYPTOCFGA_REPORT_LARVAL,
                sizeof(struct crypto_report_larval), &rl))
            goto nla_put_failure;
        goto out;
    }

    if (alg->cra_type && alg->cra_type->report) {
        if (alg->cra_type->report(skb, alg))
            goto nla_put_failure;
```

```
116
117         goto out;
118     }
119
120     switch (alg->cra_flags & (CRYPTO_ALG_TYPE_MASK | CRYPTO_ALG_LARVAL)) {
121     case CRYPTO_ALG_TYPE_CIPHER:
122         if (crypto_report_cipher(skb, alg))
123             goto nla_put_failure;
124
125         break;
126     case CRYPTO_ALG_TYPE_COMPRESS:
127         if (crypto_report_comp(skb, alg))
128             goto nla_put_failure;
129
130         break;
131     }
132
133 out:
134     return 0;
135
136 nla_put_failure:
137     return -EMSGSIZE;
138 }
139
140 static int crypto_report_alg(struct crypto_alg *alg,
141                  struct crypto_dump_info *info)
142 {
143     //.......
144 }
145
146 static int crypto_report(struct sk_buff *in_skb, struct nlmsghdr *in_nlh,
147              struct nlattr **attrs)
148 {
149     //.......
150 }
151
152 static int crypto_dump_report(struct sk_buff *skb, struct netlink_callback *cb)
153 {
154     //.......
155 }
156
157 static int crypto_dump_report_done(struct netlink_callback *cb)
158 {
159     return 0;
160 }
161
162 static int crypto_update_alg(struct sk_buff *skb, struct nlmsghdr *nlh,
163                  struct nlattr **attrs)
164 {
165     struct crypto_alg *alg;
166     struct crypto_user_alg *p = nlmsg_data(nlh);
167     struct nlattr *priority = attrs[CRYPTOCFGA_PRIORITY_VAL];
168     LIST_HEAD(list);
169
170     if (priority && !strlen(p->cru_driver_name))
171         return -EINVAL;
172
173     alg = crypto_alg_match(p, 1);
174     if (!alg)
175         return -ENOENT;
176
```

```
177     down_write(&crypto_alg_sem);
178
179     crypto_remove_spawns(alg, &list, NULL);
180
181     if (priority)
182         alg->cra_priority = nla_get_u32(priority);
183
184     up_write(&crypto_alg_sem);
185
186     crypto_remove_final(&list);
187
188     return 0;
189 }
190
191 static int crypto_del_alg(struct sk_buff *skb, struct nlmsghdr *nlh,
192               struct nlattr **attrs)
193 {
194     //.......
195 }
196
197 static struct crypto_alg *crypto_user_skcipher_alg(const char *name, u32 type,
198                            u32 mask)
199 {
200     //.......
201 }
202
203 static struct crypto_alg *crypto_user_aead_alg(const char *name, u32 type,
204                            u32 mask)
205 {
206     //.......
207 }
208
209 static int crypto_add_alg(struct sk_buff *skb, struct nlmsghdr *nlh,
210               struct nlattr **attrs)
211 {
212     int exact = 0;
213     const char *name;
214     struct crypto_alg *alg;
215     struct crypto_user_alg *p = nlmsg_data(nlh);
216     struct nlattr *priority = attrs[CRYPTOCFGA_PRIORITY_VAL];
217
218     if (strlen(p->cru_driver_name))
219         exact = 1;
220
221     if (priority && !exact)
222         return -EINVAL;
223
224     alg = crypto_alg_match(p, exact);
225     if (alg)
226         return -EEXIST;
227
228     if (strlen(p->cru_driver_name))
229         name = p->cru_driver_name;
230     else
231         name = p->cru_name;
232
233     switch (p->cru_type & p->cru_mask & CRYPTO_ALG_TYPE_MASK) {
234     case CRYPTO_ALG_TYPE_AEAD:
235         alg = crypto_user_aead_alg(name, p->cru_type, p->cru_mask);
236         break;
237     case CRYPTO_ALG_TYPE_GIVCIPHER:
```

```
238     case CRYPTO_ALG_TYPE_BLKCIPHER:
239     case CRYPTO_ALG_TYPE_ABLKCIPHER:
240         alg = crypto_user_skcipher_alg(name, p->cru_type, p->cru_mask);
241         break;
242     default:
243         alg = crypto_alg_mod_lookup(name, p->cru_type, p->cru_mask);
244     }
245
246     if (IS_ERR(alg))
247         return PTR_ERR(alg);
248
249     down_write(&crypto_alg_sem);
250
251     if (priority)
252         alg->cra_priority = nla_get_u32(priority);
253
254     up_write(&crypto_alg_sem);
255
256     crypto_mod_put(alg);
257
258     return 0;
259 }
```

STANDARD: Potentially Unsafe Code - strlen

Line: 57 - D:\Question_code\crypto_user.c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

if (strlen(p-gt;cru_driver_name))

MEDIUM: Potentially Unsafe Code - snprintf(

Line: 78 - D:\Question_code\crypto_user.c

snprintf() does not fill the remainder of the buffer with null byte

snprintf(rcipher.type, CRYPTO_MAX_ALG_NAME, "%s", "cipher");

MEDIUM: Potentially Unsafe Code - goto

Line: 86 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto nla_put_failure;

MEDIUM: Potentially Unsafe Code - snprintf(

Line: 97 - D:\Question_code\crypto_user.c

snprintf() does not fill the remainder of the buffer with null byte

snprintf(rcomp.type, CRYPTO_MAX_ALG_NAME, "%s", "compression");

MEDIUM: Potentially Unsafe Code - goto

Line: 101 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto nla_put_failure;

MEDIUM: Potentially Unsafe Code - memcpy

Line: 111 - D:\Question_code\crypto_user.c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(ualg-gt;cru_name, alg-gt;cra_name, sizeof(ualg-gt;cru_name));

MEDIUM: Potentially Unsafe Code - memcpy

Line: 112 - D:\Question_code\crypto_user.c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(ualg-gt;cru_driver_name, alg-gt;cra_driver_name,

MEDIUM: Potentially Unsafe Code - memcpy

Line: 114 - D:\Question_code\crypto_user.c

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(ualg-gt;cru_module_name, module_name(alg-gt;cra_module),

MEDIUM: Potentially Unsafe Code - goto

Line: 121 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto nla_put_failure;

MEDIUM: Potentially Unsafe Code - snprintf(

Line: 125 - D:\Question_code\crypto_user.c

snprintf() does not fill the remainder of the buffer with null byte

snprintf(rl.type, CRYPTO_MAX_ALG_NAME, "%s", "larval");

MEDIUM: Potentially Unsafe Code - goto

Line: 129 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto nla_put_failure;

MEDIUM: Potentially Unsafe Code - goto

Line: 130 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto out;

MEDIUM: Potentially Unsafe Code - goto

Line: 135 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto nla_put_failure;

MEDIUM: Potentially Unsafe Code - goto

Line: 137 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto out;

MEDIUM: Potentially Unsafe Code - goto

Line: 143 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto nla_put_failure;

MEDIUM: Potentially Unsafe Code - goto

Line: 148 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto nla_put_failure;

MEDIUM: Potentially Unsafe Code - goto

Line: 173 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto out;

MEDIUM: Potentially Unsafe Code - goto

Line: 181 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto out;

MEDIUM: Potentially Unsafe Code - goto

Line: 229 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto out;

MEDIUM: Potentially Unsafe Code - goto

Line: 241 - D:\Question_code\crypto_user.c

Use of 'goto' function. The goto function can result in unstructured code which is difficult to maintain and can result in failures to initialise or de-allocate memory.

goto out_err;

STANDARD: Potentially Unsafe Code - strlen

Line: 263 - D:\Question_code\crypto_user.c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

if (priority !strlen(p-gt;cru_driver_name))

STANDARD: Potentially Unsafe Code - strlen

Line: 371 - D:\Question_code\crypto_user.c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

if (strlen(p-gt;cru_driver_name))

STANDARD: Potentially Unsafe Code - strlen

Line: 381 - D:\Question_code\crypto_user.c

Function appears in Microsoft's banned function list. For critical applications, particularly applications accepting anonymous Internet connections or unverified input data, strlen and similar functions can become victims of integer overflow or 'wraparound' errors.

if (strlen(p-gt;cru_driver_name))

STANDARD: Potential Memory Mis-management. Variable Name: skb new without delete.

Line: 206 FileName: D:\Question_code\crypto_user.c

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal3.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/514/qid/16228) == "1"))

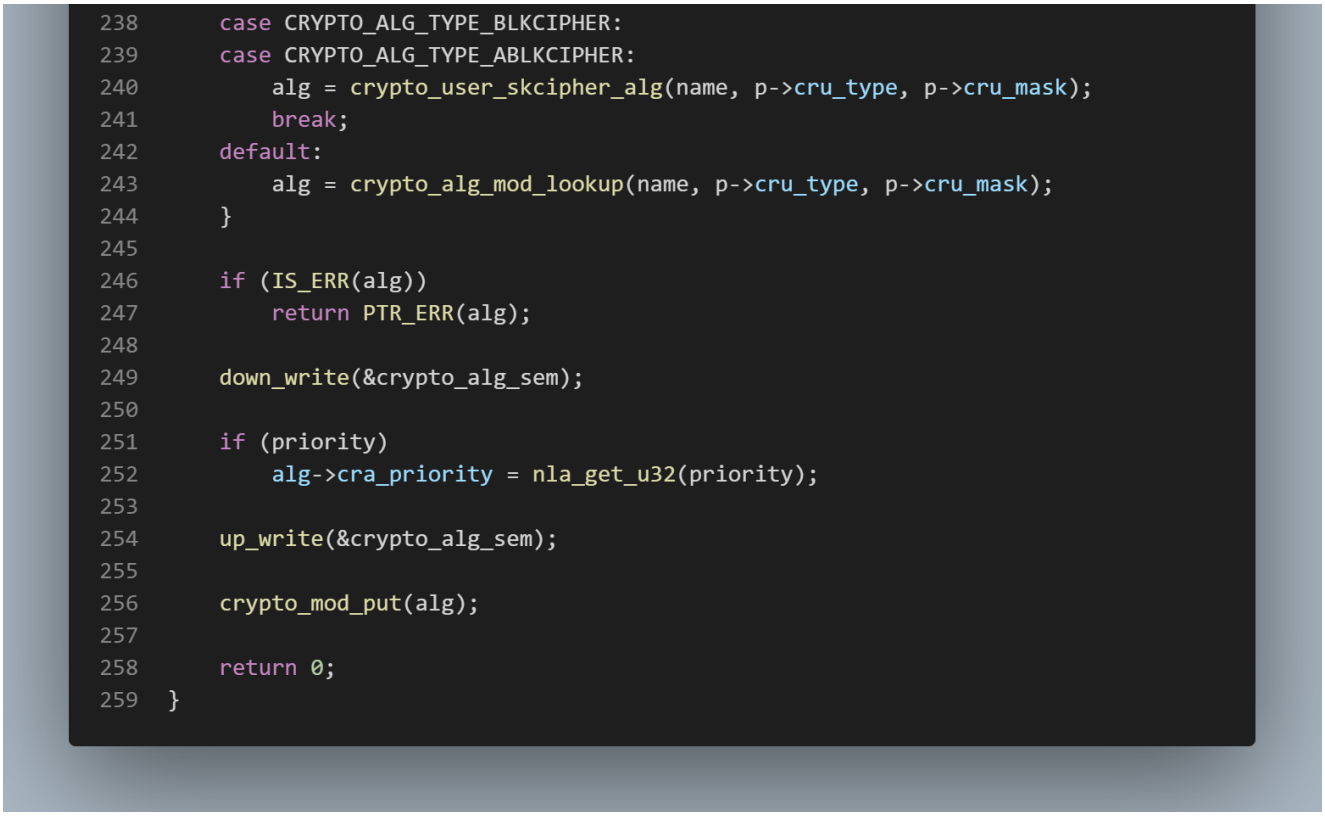
Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du fichier complet.

```c
#include <linux/module.h>
#include <linux/crypto.h>
#include <linux/cryptouser.h>
#include <linux/sched.h>
#include <net/netlink.h>
#include <linux/security.h>
#include <net/net_namespace.h>
#include <crypto/internal/aead.h>
#include <crypto/internal/skcipher.h>

#include "internal.h"

static DEFINE_MUTEX(crypto_cfg_mutex);

/* The crypto netlink socket */
static struct sock *crypto_nlsk;

struct crypto_dump_info {
    struct sk_buff *in_skb;
    struct sk_buff *out_skb;
    u32 nlmsg_seq;
    u16 nlmsg_flags;
};

static struct crypto_alg *crypto_alg_match(struct crypto_user_alg *p, int exact)
{
    struct crypto_alg *q, *alg = NULL;

    down_read(&crypto_alg_sem);

    list_for_each_entry(q, &crypto_alg_list, cra_list) {
        int match = 0;

        if ((q->cra_flags ^ p->cru_type) & p->cru_mask)
            continue;

        if (strlen(p->cru_driver_name))
            match = !strcmp(q->cra_driver_name,
                    p->cru_driver_name);
        else if (!exact)
            match = !strcmp(q->cra_name, p->cru_name);

        if (match) {
            alg = q;
            break;
        }
    }

    up_read(&crypto_alg_sem);

    return alg;
}

static int crypto_report_cipher(struct sk_buff *skb, struct crypto_alg *alg)
```

```
{
    struct crypto_report_cipher rcipher;

    snprintf(rcipher.type, CRYPTO_MAX_ALG_NAME, "%s", "cipher");

    rcipher.blocksize = alg->cra_blocksize;
    rcipher.min_keysize = alg->cra_cipher.cia_min_keysize;
    rcipher.max_keysize = alg->cra_cipher.cia_max_keysize;

    if (nla_put(skb, CRYPTOCFGA_REPORT_CIPHER,
            sizeof(struct crypto_report_cipher), &rcipher))
        goto nla_put_failure;
    return 0;

nla_put_failure:
    return -EMSGSIZE;
}

static int crypto_report_comp(struct sk_buff *skb, struct crypto_alg *alg)
{
    struct crypto_report_comp rcomp;

    snprintf(rcomp.type, CRYPTO_MAX_ALG_NAME, "%s", "compression");

    if (nla_put(skb, CRYPTOCFGA_REPORT_COMPRESS,
            sizeof(struct crypto_report_comp), &rcomp))
        goto nla_put_failure;
    return 0;

nla_put_failure:
    return -EMSGSIZE;
}

static int crypto_report_one(struct crypto_alg *alg,
                 struct crypto_user_alg *ualg, struct sk_buff *skb)
{
    memcpy(&ualg->cru_name, &alg->cra_name, sizeof(ualg->cru_name));
    memcpy(&ualg->cru_driver_name, &alg->cra_driver_name,
           sizeof(ualg->cru_driver_name));
    memcpy(&ualg->cru_module_name, module_name(alg->cra_module),
           CRYPTO_MAX_ALG_NAME);

    ualg->cru_flags = alg->cra_flags;
    ualg->cru_refcnt = atomic_read(&alg->cra_refcnt);

    if (nla_put_u32(skb, CRYPTOCFGA_PRIORITY_VAL, alg->cra_priority))
        goto nla_put_failure;
    if (alg->cra_flags & CRYPTO_ALG_LARVAL) {
        struct crypto_report_larval rl;

        snprintf(rl.type, CRYPTO_MAX_ALG_NAME, "%s", "larval");

        if (nla_put(skb, CRYPTOCFGA_REPORT_LARVAL,
                sizeof(struct crypto_report_larval), &rl))
            goto nla_put_failure;
        goto out;
    }

    if (alg->cra_type && alg->cra_type->report) {
        if (alg->cra_type->report(skb, alg))
            goto nla_put_failure;
```

```
116
117         goto out;
118     }
119
120     switch (alg->cra_flags & (CRYPTO_ALG_TYPE_MASK | CRYPTO_ALG_LARVAL)) {
121     case CRYPTO_ALG_TYPE_CIPHER:
122         if (crypto_report_cipher(skb, alg))
123             goto nla_put_failure;
124
125         break;
126     case CRYPTO_ALG_TYPE_COMPRESS:
127         if (crypto_report_comp(skb, alg))
128             goto nla_put_failure;
129
130         break;
131     }
132
133 out:
134     return 0;
135
136 nla_put_failure:
137     return -EMSGSIZE;
138 }
139
140 static int crypto_report_alg(struct crypto_alg *alg,
141                  struct crypto_dump_info *info)
142 {
143     //.......
144 }
145
146 static int crypto_report(struct sk_buff *in_skb, struct nlmsghdr *in_nlh,
147              struct nlattr **attrs)
148 {
149     //.......
150 }
151
152 static int crypto_dump_report(struct sk_buff *skb, struct netlink_callback *cb)
153 {
154     //.......
155 }
156
157 static int crypto_dump_report_done(struct netlink_callback *cb)
158 {
159     return 0;
160 }
161
162 static int crypto_update_alg(struct sk_buff *skb, struct nlmsghdr *nlh,
163                  struct nlattr **attrs)
164 {
165     struct crypto_alg *alg;
166     struct crypto_user_alg *p = nlmsg_data(nlh);
167     struct nlattr *priority = attrs[CRYPTOCFGA_PRIORITY_VAL];
168     LIST_HEAD(list);
169
170     if (priority && !strlen(p->cru_driver_name))
171         return -EINVAL;
172
173     alg = crypto_alg_match(p, 1);
174     if (!alg)
175         return -ENOENT;
176
```

```
177     down_write(&crypto_alg_sem);
178
179     crypto_remove_spawns(alg, &list, NULL);
180
181     if (priority)
182         alg->cra_priority = nla_get_u32(priority);
183
184     up_write(&crypto_alg_sem);
185
186     crypto_remove_final(&list);
187
188     return 0;
189 }
190
191 static int crypto_del_alg(struct sk_buff *skb, struct nlmsghdr *nlh,
192               struct nlattr **attrs)
193 {
194     //.......
195 }
196
197 static struct crypto_alg *crypto_user_skcipher_alg(const char *name, u32 type,
198                            u32 mask)
199 {
200     //.......
201 }
202
203 static struct crypto_alg *crypto_user_aead_alg(const char *name, u32 type,
204                            u32 mask)
205 {
206     //.......
207 }
208
209 static int crypto_add_alg(struct sk_buff *skb, struct nlmsghdr *nlh,
210               struct nlattr **attrs)
211 {
212     int exact = 0;
213     const char *name;
214     struct crypto_alg *alg;
215     struct crypto_user_alg *p = nlmsg_data(nlh);
216     struct nlattr *priority = attrs[CRYPTOCFGA_PRIORITY_VAL];
217
218     if (strlen(p->cru_driver_name))
219         exact = 1;
220
221     if (priority && !exact)
222         return -EINVAL;
223
224     alg = crypto_alg_match(p, exact);
225     if (alg)
226         return -EEXIST;
227
228     if (strlen(p->cru_driver_name))
229         name = p->cru_driver_name;
230     else
231         name = p->cru_name;
232
233     switch (p->cru_type & p->cru_mask & CRYPTO_ALG_TYPE_MASK) {
234     case CRYPTO_ALG_TYPE_AEAD:
235         alg = crypto_user_aead_alg(name, p->cru_type, p->cru_mask);
236         break;
237     case CRYPTO_ALG_TYPE_GIVCIPHER:
```



```
238     case CRYPTO_ALG_TYPE_BLKCIPHER:
239     case CRYPTO_ALG_TYPE_ABLKCIPHER:
240         alg = crypto_user_skcipher_alg(name, p->cru_type, p->cru_mask);
241         break;
242     default:
243         alg = crypto_alg_mod_lookup(name, p->cru_type, p->cru_mask);
244     }
245
246     if (IS_ERR(alg))
247         return PTR_ERR(alg);
248
249     down_write(&crypto_alg_sem);
250
251     if (priority)
252         alg->cra_priority = nla_get_u32(priority);
253
254     up_write(&crypto_alg_sem);
255
256     crypto_mod_put(alg);
257
258     return 0;
259 }
```

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal3.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/514/qid/16228) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

◯ oui
◯ non

## Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r119q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

# Il y a-t-il une erreur dans ce code? (Program.cs)

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal9.NAOK),rand(1,2),randVal9.NAOK)}**

```
using HashLib;
using System;

// https://github.com/yh200212121212/HashLib2/blob/master/Demo/Program.cs

namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            string str = "Test";
            int[] input = {
                1,3,2,3,4,5,1,2,3,4,99,109,33,4,2,32,32,-1,-33,-1078,-3459,0xcf,0x9c,0xff,0x777ffdc,-0x934dc,3,40560,20,930,1809,2,3,4,5,66,54,190,304,22,34,56,1,2,1,0,-9,-99,-190,-1238&0x9c,~010010,-2,-193456,int.MaxValue,int.MinValue
            };
            IHashLib hashLib = new HashLib.HashLib();
            System.Diagnostics.Stopwatch stopwatch = new System.Diagnostics.Stopwatch();
            stopwatch.Start();
            long[] result = hashLib.hash2(input), result2 = hashLib.hash_salt2(input), result3 = hashLib.hash2(str), result4 = hashLib.hash_salt2(str);
            stopwatch.Stop();
            Console.WriteLine("The first of the output array:{0},{1},{2},{3}", result[0], result2[0], result3[0], result4[0]);
            Console.WriteLine(TimeSpan.FromMilliseconds(stopwatch.ElapsedMilliseconds).TotalMinutes.ToString());
            Console.ReadKey();
        }
    }
}

```

(https://github.com/2longAGO/Projet_Synth/blob/main/Program.cs)

HIGH: Potentially Unsafe Code - HashLib

Line: 1 - D:\Question_code\Program.cs

The cryptography library Hashlib is no longer supported

using HashLib;

HIGH: Potentially Unsafe Code - HashLib

Line: 16 - D:\Question_code\Program.cs

The cryptography library Hashlib is no longer supported

IHashLib hashLib = new HashLib.HashLib();

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal9.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/512/qid/16218) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.

```
using HashLib;
using System;

// https://github.com/yh200212121212/HashLib2/blob/master/Demo/Program.cs

namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            string str = "Test";
            int[] input = {
                1,3,2,3,4,5,1,2,3,4,99,109,33,4,2,32,32,-1,-33,-1078,-3459,0xcf,0x9c,0xff,0x777ffdc,-0x934dc,3,40560,20,930,1809,2,3,4,5,66,54,190,304,22,34,56,1,2,1,0,-9,-99,-190,-1238x9c,~010010,-2,-193456,int.MaxValue,int.MinValue
            };
            IHashLib hashLib = new HashLib.HashLib();
            System.Diagnostics.Stopwatch stopwatch = new System.Diagnostics.Stopwatch();
            stopwatch.Start();
            long[] result = hashLib.hash2(input), result2 = hashLib.hash_salt2(input), result3 = hashLib.hash2(str), result4 = hashLib.hash_salt2(str);
            stopwatch.Stop();
            Console.WriteLine("The first of the output array:{0},{1},{2},{3}", result[0], result2[0], result3[0], result4[0]);
            Console.WriteLine(TimeSpan.FromMilliseconds(stopwatch.ElapsedMilliseconds).TotalMinutes.ToString());
            Console.ReadKey();
        }
    }
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/Program.cs)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal9.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/512/qid/16218) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

◯ oui
◯ non

## Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r840q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

# Il y a-t-il une erreur dans ce code?

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne

accès au fichier du code.

**{if(is_empty(randVal6.NAOK),rand(1,2),randVal6.NAOK)}**

```php
<?php
/**
 * /reset.php
 *
 * This file is part of DomainMOD, an open source domain and internet asset manager.
 * Copyright (c) 2010-2019 Greg Chetcuti <greg@chetcuti.com>
 *
 * Project: http://domainmod.org   Author: http://chetcuti.com
 *
 * DomainMOD is free software: you can redistribute it and/or modify it under the terms of the GNU General Public
 * License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later
 * version.
 *
 * DomainMOD is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied
 * warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License along with DomainMOD. If not, see
 * http://www.gnu.org/licenses/.
 *
 */
?>
<?php
require_once __DIR__ . '/_includes/start-session.inc.php';
require_once __DIR__ . '/_includes/init.inc.php';
require_once DIR_INC . '/config.inc.php';
require_once DIR_INC . '/software.inc.php';
require_once DIR_ROOT . '/vendor/autoload.php';

$deeb = DomainMOD\Database::getInstance();
$system = new DomainMOD\System();
$maint = new DomainMOD\Maintenance();
$layout = new DomainMOD\Layout();
$time = new DomainMOD\Time();
$form = new DomainMOD\Form();

require_once DIR_INC . '/head.inc.php';
require_once DIR_INC . '/debug.inc.php';

$system->loginCheck();
$pdo = $deeb->cnxx;

$page_title = "Reset Password";
$software_section = "resetpassword";

$user_identifier = $_REQUEST['user_identifier'];

if ($user_identifier != '') {

    $stmt = $pdo->prepare("
        SELECT first_name, last_name, username, email_address
        FROM users
        WHERE (username = :username OR email_address = :email_address)
          AND active = '1'");
    $stmt->bindValue('username', $user_identifier, PDO::PARAM_STR);
    $stmt->bindValue('email_address', $user_identifier, PDO::PARAM_STR);
    $stmt->execute();
    $result = $stmt->fetch();
    $stmt->closeCursor();

    if (!$result) {

        $_SESSION['s_message_success'] .= "If there is a matching username or email address in the system your new password will been emailed to you.<BR>";

        header('Location: ' . $web_root . "/");
        exit;

    } else {

        $new_password = substr(md5(time()), 0, 8);

        $stmt = $pdo->prepare("
            UPDATE users
            SET `password` = password(:new_password),
                new_password = '1',
                update_time = :timestamp
            WHERE username = :username
              AND email_address = :email_address");
        $stmt->bindValue('new_password', $new_password, PDO::PARAM_STR);
        $bind_timestamp = $time->stamp();
        $stmt->bindValue('timestamp', $bind_timestamp, PDO::PARAM_STR);
        $stmt->bindValue('username', $result->username, PDO::PARAM_STR);
        $stmt->bindValue('email_address', $result->email_address, PDO::PARAM_STR);
        $stmt->execute();

        $first_name = $result->first_name;
        $last_name = $result->last_name;
        $username = $result->username;
        $email_address = $result->email_address;
        require_once DIR_INC . '/email/send-new-password.inc.php';

        $_SESSION['s_message_success'] .= "If there is a matching username or email address in the system your new password will been emailed to you.<BR>";

        header('Location: ' . $web_root . "/");
        exit;

    }

} else {
```

```
99
100     if ($_SERVER['REQUEST_METHOD'] == 'POST') {
101
102         if ($user_identifier == "") {
103             $_SESSION['s_message_danger'] .= "Enter your username or email address<BR>";
104         }
105
106     }
107
108 }
109 ?>
110 <?php require_once DIR_INC . '/doctype.inc.php'; ?>
111 <html>
112 <head>
113     <title><?php echo $layout->pageTitle($page_title); ?></title>
114     <?php require_once DIR_INC . '/layout/head-tags.inc.php'; ?>
115 </head>
116 <body class="hold-transition skin-red" onLoad="document.forms[0].elements[0].focus()">
117 <?php require_once DIR_INC . '/layout/header-login.inc.php'; ?>
118 <?php
119     echo $form->showFormTop('');
120     echo $form->showInputText('user_identifier', 'Username or Email Address', '', $user_identifier, '100', '', '', '', '');
121     echo $form->showSubmitButton('Reset Password', '', '');
122     echo $form->showFormBottom('');
123 ?>
124 <BR><a href="<?php echo $web_root; ?>/">Cancel Password Reset</a>
125 <?php require_once DIR_INC . '/layout/footer-login.inc.php'; ?>
126 </body>
127 </html>
128
```

(https://github.com/2longAGO/Projet_Synth/blob/main/reset.php)

MEDIUM: Potentially Unsafe Code - AESEngine

Line: 27 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

This class defaults to ECB mode, which is unsafe.

```
import org.bouncycastle.crypto.engines.AESEngine;
```

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 52 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

```
public class IESCipher
```

LOW: Potentially Unsafe Code - Operation on Primitive Data Type

Line: 58 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

```
private int state = -1;
```

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 427 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

```
static public class IES
```

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 438 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

```
static public class IESwithDESede
```

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 450 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class IESwithAES

MEDIUM: Potentially Unsafe Code - AESEngine

Line: 458 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

This class defaults to ECB mode, which is unsafe.

new PaddedBufferedBlockCipher(new AESEngine()));

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 465 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class OldIESwithCipher

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 477 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually

inspect the code to determine whether or not it is practical to make this class final.

static public class OldIES

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 488 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class OldIESwithDESede

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 497 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class OldIESwithAES

MEDIUM: Potentially Unsafe Code - AESEngine

Line: 502 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

This class defaults to ECB mode, which is unsafe.

super(new AESEngine());

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal6.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/509/qid/16203) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.

```php
<?php
/**
 * /reset.php
 *
 * This file is part of DomainMOD, an open source domain and internet asset manager.
 * Copyright (c) 2010-2019 Greg Chetcuti <greg@chetcuti.com>
 *
 * Project: http://domainmod.org   Author: http://chetcuti.com
 *
 * DomainMOD is free software: you can redistribute it and/or modify it under the terms of the GNU General Public
 * License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later
 * version.
 *
 * DomainMOD is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied
 * warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License along with DomainMOD. If not, see
 * http://www.gnu.org/licenses/.
 *
 */
?>
<?php
require_once __DIR__ . '/_includes/start-session.inc.php';
require_once __DIR__ . '/_includes/init.inc.php';
require_once DIR_INC . '/config.inc.php';
require_once DIR_INC . '/software.inc.php';
require_once DIR_ROOT . '/vendor/autoload.php';

$deeb = DomainMOD\Database::getInstance();
$system = new DomainMOD\System();
$maint = new DomainMOD\Maintenance();
$layout = new DomainMOD\Layout();
$time = new DomainMOD\Time();
$form = new DomainMOD\Form();

require_once DIR_INC . '/head.inc.php';
require_once DIR_INC . '/debug.inc.php';

$system->loginCheck();
$pdo = $deeb->cnxx;

$page_title = "Reset Password";
$software_section = "resetpassword";

$user_identifier = $_REQUEST['user_identifier'];

if ($user_identifier != '') {

    $stmt = $pdo->prepare("
        SELECT first_name, last_name, username, email_address
        FROM users
        WHERE (username = :username OR email_address = :email_address)
          AND active = '1'");
    $stmt->bindValue('username', $user_identifier, PDO::PARAM_STR);
    $stmt->bindValue('email_address', $user_identifier, PDO::PARAM_STR);
    $stmt->execute();
    $result = $stmt->fetch();
    $stmt->closeCursor();

    if (!$result) {

        $_SESSION['s_message_success'] .= "If there is a matching username or email address in the system your new password will been emailed to you.<BR>";

        header('Location: ' . $web_root . "/");
        exit;

    } else {

        $new_password = substr(md5(time()), 0, 8);

        $stmt = $pdo->prepare("
            UPDATE users
            SET `password` = password(:new_password),
                new_password = '1',
                update_time = :timestamp
            WHERE username = :username
              AND email_address = :email_address");
        $stmt->bindValue('new_password', $new_password, PDO::PARAM_STR);
        $bind_timestamp = $time->stamp();
        $stmt->bindValue('timestamp', $bind_timestamp, PDO::PARAM_STR);
        $stmt->bindValue('username', $result->username, PDO::PARAM_STR);
        $stmt->bindValue('email_address', $result->email_address, PDO::PARAM_STR);
        $stmt->execute();

        $first_name = $result->first_name;
        $last_name = $result->last_name;
        $username = $result->username;
        $email_address = $result->email_address;
        require_once DIR_INC . '/email/send-new-password.inc.php';

        $_SESSION['s_message_success'] .= "If there is a matching username or email address in the system your new password will been emailed to you.<BR>";

        header('Location: ' . $web_root . "/");
        exit;

    }

} else {
```

```php

    if ($_SERVER['REQUEST_METHOD'] == 'POST') {

        if ($user_identifier == "") {
            $_SESSION['s_message_danger'] .= "Enter your username or email address<BR>";
        }

    }

}
?>
<?php require_once DIR_INC . '/doctype.inc.php'; ?>
<html>
<head>
    <title><?php echo $layout->pageTitle($page_title); ?></title>
    <?php require_once DIR_INC . '/layout/head-tags.inc.php'; ?>
</head>
<body class="hold-transition skin-red" onLoad="document.forms[0].elements[0].focus()">
<?php require_once DIR_INC . '/layout/header-login.inc.php'; ?>
<?php
    echo $form->showFormTop('');
    echo $form->showInputText('user_identifier', 'Username or Email Address', '', $user_identifier, '100', '', '', '', '');
    echo $form->showSubmitButton('Reset Password', '', '');
    echo $form->showFormBottom('');
?>
<BR><a href="<?php echo $web_root; ?>/">Cancel Password Reset</a>
<?php require_once DIR_INC . '/layout/footer-login.inc.php'; ?>
</body>
</html>

```

(https://github.com/2longAGO/Projet_Synth/blob/main/reset.php)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal6.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/509/qid/16203) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

- ◯ oui
- ◯ non

## Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r974q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

# Il y a-t-il une erreur dans ce code?

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal7.NAOK),rand(1,2),randVal7.NAOK)}**

```php
<?php
// https://github.com/defuse/php-encryption/blob/master/docs/Tutorial.md
use Defuse\Crypto\Crypto;
use Defuse\Crypto\KeyProtectedByPassword;
use Defuse\Crypto\Key;

$user_key_encoded = // ... get it out of the cookie ...
$user_locked_key = KeyProtectedByPassword::loadFromAsciiSafeString($user_key_encoded)
$user_key = $user_locked_key->unlockKey($user_key_encoded);

// ...

$credit_card_number = // ... get credit card number from the user
$encrypted_card_number = Crypto::encrypt($credit_card_number, $user_key);
?>
```

(https://github.com/2longAGO/Projet_Synth/blob/main/enc_.php)

MEDIUM: Potentially Unsafe Code - KeyProtectedByPassword

Line: 4 - D:\Question_code\enc.php

This class is vulnerable because it save and check the password in the format sha256($password): https://github.com/defuse/php-encryption/issues/392

use Defuse\Crypto\KeyProtectedByPassword;

MEDIUM: Potentially Unsafe Code - KeyProtectedByPassword

Line: 8 - D:\Question_code\enc.php

This class is vulnerable because it save and check the password in the format sha256($password): https://github.com/defuse/php-encryption/issues/392

$user_locked_key = KeyProtectedByPassword::loadFromAsciiSafeString($user_key_encoded)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal7.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/510/qid/16208) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.

```php
<?php
// https://github.com/defuse/php-encryption/blob/master/docs/Tutorial.md
use Defuse\Crypto\Crypto;
use Defuse\Crypto\KeyProtectedByPassword;
use Defuse\Crypto\Key;

$user_key_encoded = // ... get it out of the cookie ...
$user_locked_key = KeyProtectedByPassword::loadFromAsciiSafeString($user_key_encoded)
$user_key = $user_locked_key->unlockKey($user_key_encoded);

// ...

$credit_card_number = // ... get credit card number from the user
$encrypted_card_number = Crypto::encrypt($credit_card_number, $user_key);
?>
```

(https://github.com/2longAGO/Projet_Synth/blob/main/enc_.php)

Répondre à cette question seulement si les conditions suivantes sont réunies : ((randVal7.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/510/qid/16208) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

- ◯ oui
- ◯ non

## Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r438q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

# Il y a-t-il une erreur dans ce code? (ext_mcrypt.cpp)

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal5.NAOK),rand(1,2),randVal5.NAOK)}**

```
#include "hphp/runtime/base/base-includes.h"
#include "hphp/runtime/base/runtime-error.h"

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define NON_FREE
#define MCRYPT2
#include <mcrypt.h>

namespace HPHP {

////////////////////////////////////////////////////////////////////////////////

class MCrypt : public SweepableResourceData {
public:
  explicit MCrypt(MCRYPT td) : m_td(td), m_init(false) {
  }

  ~MCrypt() {
    MCrypt::close();
  }

  void sweep() FOLLY_OVERRIDE {
    close();
  }

  void close() {
    if (m_td != MCRYPT_FAILED) {
      mcrypt_generic_deinit(m_td);
      mcrypt_module_close(m_td);
      m_td = MCRYPT_FAILED;
    }
  }

  CLASSNAME_IS("mcrypt");
  // overriding ResourceData
  virtual const String& o_getClassNameHook() const { return classnameof(); }

  MCRYPT m_td;
  bool m_init;
};

typedef enum {
  RANDOM = 0,
  URANDOM,
  RAND
} iv_source;

class mcrypt_data {
public:
  std::string algorithms_dir;
  std::string modes_dir;
```

```
};
static mcrypt_data s_globals;
#define MCG(n) (s_globals.n)
#ifndef MIN
#define MIN(a, b) ((a) < (b) ? (a) : (b))
#endif
#define MCRYPT_OPEN_MODULE_FAILED(str) \
 raise_warning("%s(): Module initialization failed", str);

static Variant php_mcrypt_do_crypt(const String& cipher, const String& key,
                                   const String& data, const String& mode,
                                   const String& iv, bool dencrypt,
                                   char *name) {
  // .....
}

static Variant mcrypt_generic(const Resource& td, const String& data,
                              bool dencrypt) {
  MCrypt *pm = td.getTyped<MCrypt>();
  if (!pm->m_init) {
    raise_warning("Operation disallowed prior to mcrypt_generic_init().");
    return false;
  }

  if (data.empty()) {
    raise_warning("An empty string was passed");
    return false;
  }

  String s;
  unsigned char* data_s;
  int block_size, data_size;
  /* Check blocksize */
  if (mcrypt_enc_is_block_mode(pm->m_td) == 1) { /* It's a block algorithm */
    block_size = mcrypt_enc_get_block_size(pm->m_td);
    data_size = (((data.size() - 1) / block_size) + 1) * block_size;
    s = String(data_size, ReserveString);
    data_s = (unsigned char *)s.bufferSlice().ptr;
    memset(data_s, 0, data_size);
    memcpy(data_s, data.data(), data.size());
  } else { /* It's not a block algorithm */
    data_size = data.size();
    s = String(data_size, ReserveString);
    data_s = (unsigned char *)s.bufferSlice().ptr;
    memcpy(data_s, data.data(), data.size());
  }

  if (dencrypt) {
    mdecrypt_generic(pm->m_td, data_s, data_size);
  } else {
    mcrypt_generic(pm->m_td, data_s, data_size);
  }
  s.setSize(data_size);
  return s;
}

////////////////////////////////////////////////////////////////////////////////

Variant HHVM_FUNCTION(mcrypt_module_open, const String& algorithm,
                                          const String& algorithm_directory,
                              const String& mode, const String& mode_directory) {
```

```
116   // .....
117 }
118
119 bool HHVM_FUNCTION(mcrypt_module_close, const Resource& td) {
120   td.getTyped<MCrypt>()->close();
121   return true;
122 }
123
124 Array HHVM_FUNCTION(mcrypt_list_algorithms,
125                     const String& lib_dir /* = null_string */) {
126   // .....
127 }
128
129 Array HHVM_FUNCTION(mcrypt_list_modes,
130                     const String& lib_dir /* = null_string */) {
131   // .....
132 }
133
134 int64_t HHVM_FUNCTION(mcrypt_module_get_algo_block_size,
135                                  const String& algorithm,
136                                  const String& lib_dir /* = null_string */) {
137   String dir = lib_dir.empty() ? String(MCG(algorithms_dir)) : lib_dir;
138   return mcrypt_module_get_algo_block_size((char*)algorithm.data(),
139                                            (char*)dir.data());
140 }
141
142 int64_t HHVM_FUNCTION(mcrypt_module_get_algo_key_size, const String& algorithm,
143                                  const String& lib_dir /* = null_string */) {
144   String dir = lib_dir.empty() ? String(MCG(algorithms_dir)) : lib_dir;
145   return mcrypt_module_get_algo_key_size((char*)algorithm.data(),
146                                          (char*)dir.data());
147 }
148
149 Array HHVM_FUNCTION(mcrypt_module_get_supported_key_sizes,
150                     const String& algorithm,
151                     const String& lib_dir /* = null_string */) {
152   // .....
153 }
154
155 bool HHVM_FUNCTION(mcrypt_module_is_block_algorithm_mode, const String& mode,
156                                 const String& lib_dir /* = null_string */) {
157   String dir = lib_dir.empty() ? String(MCG(modes_dir)) : lib_dir;
158   return mcrypt_module_is_block_algorithm_mode((char*)mode.data(),
159                                                (char*)dir.data()) == 1;
160 }
161
162 bool HHVM_FUNCTION(mcrypt_module_is_block_algorithm, const String& algorithm,
163                                 const String& lib_dir /* = null_string */) {
164   String dir = lib_dir.empty() ? String(MCG(algorithms_dir)) : lib_dir;
165   return mcrypt_module_is_block_algorithm((char*)algorithm.data(),
166                                           (char*)dir.data()) == 1;
167 }
168
169 bool HHVM_FUNCTION(mcrypt_module_is_block_mode, const String& mode,
170                                  const String& lib_dir /* = null_string */) {
171   String dir = lib_dir.empty() ? String(MCG(modes_dir)) : lib_dir;
172   return mcrypt_module_is_block_mode((char*)mode.data(),
173                                      (char*)dir.data()) == 1;
174 }
175
176 bool HHVM_FUNCTION(mcrypt_module_self_test, const String& algorithm,
```

```
                                const String& lib_dir /* = null_string */) {
  String dir = lib_dir.empty() ? String(MCG(algorithms_dir)) : lib_dir;
  return mcrypt_module_self_test((char*)algorithm.data(),
                                 (char*)dir.data()) == 0;
}

Variant HHVM_FUNCTION(mcrypt_create_iv, int size, int source /* = 0 */) {
  if (size <= 0 || size >= INT_MAX) {
    raise_warning("Can not create an IV with a size of less than 1 or "
                  "greater than %d", INT_MAX);
    return false;
  }

  int n = 0;
  char *iv = (char*)calloc(size + 1, 1);
  if (source == RANDOM || source == URANDOM) {
    int fd = open(source == RANDOM ? "/dev/random" : "/dev/urandom", O_RDONLY);
    if (fd < 0) {
      free(iv);
      raise_warning("Cannot open source device");
      return false;
    }
    int read_bytes;
    for (read_bytes = 0; read_bytes < size && n >= 0; read_bytes += n) {
      n = read(fd, iv + read_bytes, size - read_bytes);
    }
    n = read_bytes;
    close(fd);
    if (n < size) {
      free(iv);
      raise_warning("Could not gather sufficient random data");
      return false;
    }
  } else {
    n = size;
    while (size) {
      iv[--size] = (char)(255.0 * rand() / RAND_MAX);
    }
  }
  return String(iv, n, AttachString);
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/ext_mcrypt.cpp)

MEDIUM: Potentially Unsafe Code - memcpy

Line: 106 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(key_s, key.data(), use_key_length);

MEDIUM: Potentially Unsafe Code - memcpy

Line: 110 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(key_s, key.data(), MIN(key.size(), key_length_sizes[0]));

MEDIUM: Potentially Unsafe Code - memcpy

Line: 122 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(key_s, key.data(), MIN(key.size(), use_key_length));

MEDIUM: Potentially Unsafe Code - memcpy

Line: 138 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(iv_s, iv.data(), iv_size);

MEDIUM: Potentially Unsafe Code - memcpy

Line: 159 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(data_s, data.data(), data.size());

MEDIUM: Potentially Unsafe Code - memcpy

Line: 164 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(data_s, data.data(), data.size());

MEDIUM: Potentially Unsafe Code - memcpy

Line: 212 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(data_s, data.data(), data.size());

MEDIUM: Potentially Unsafe Code - memcpy

Line: 217 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(data_s, data.data(), data.size());

STANDARD: Potentially Unsafe Code - rand

Line: 379 - D:\Question_code\ext_mcrypt.cpp

rand() is not cryptographically secure.

iv[--size] = (char)(255.0 * rand() / RAND_MAX);

MEDIUM: Potentially Unsafe Code - memcpy

Line: 588 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(key_s, key.data(), key.size());

MEDIUM: Potentially Unsafe Code - memcpy

Line: 594 - D:\Question_code\ext_mcrypt.cpp

Function appears in Microsoft's banned function list. Can facilitate buffer overflow conditions and other memory mis-management situations.

memcpy(iv_s, iv.data(), iv_size);

STANDARD: Potential Memory Mis-management. Variable Name: key_sizes

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

The use of malloc() and free() functions in C++ code is not recommended and can result in errors that would otherwise have been avoided with new and delete.

Line: 543 FileName: D:\Question_code\ext_mcrypt.cpp

STANDARD: Potential Memory Mis-management. Variable Name: iv

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

The use of malloc() and free() functions in C++ code is not recommended and can result in errors that would otherwise have been avoided with new and delete.

Line: 543 FileName: D:\Question_code\ext_mcrypt.cpp

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

The use of malloc() and free() functions in C++ code is not recommended and can result in errors that would otherwise have been avoided with new and delete.

Line: 372 FileName: D:\Question_code\ext_mcrypt.cpp

STANDARD: Potential Memory Mis-management. Variable Name: name

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

The use of malloc() and free() functions in C++ code is not

recommended and can result in errors that would otherwise have been avoided with new and delete.

Line: 543 FileName: D:\Question_code\ext_mcrypt.cpp

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

The use of malloc() and free() functions in C++ code is not recommended and can result in errors that would otherwise have been avoided with new and delete.

Line: 372 FileName: D:\Question_code\ext_mcrypt.cpp

2 free

Multiple frees detected. Check code paths manually to ensure that variables cannot be freed more than once.

The use of malloc() and free() functions in C++ code is not recommended and can result in errors that would otherwise have been avoided with new and delete.

Line: 530 FileName: D:\Question_code\ext_mcrypt.cpp

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal5.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/536/qid/16690) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.

```
#include "hphp/runtime/base/base-includes.h"
#include "hphp/runtime/base/runtime-error.h"

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define NON_FREE
#define MCRYPT2
#include <mcrypt.h>

namespace HPHP {

////////////////////////////////////////////////////////////////////////////////

class MCrypt : public SweepableResourceData {
public:
  explicit MCrypt(MCRYPT td) : m_td(td), m_init(false) {
  }

  ~MCrypt() {
    MCrypt::close();
  }

  void sweep() FOLLY_OVERRIDE {
    close();
  }

  void close() {
    if (m_td != MCRYPT_FAILED) {
      mcrypt_generic_deinit(m_td);
      mcrypt_module_close(m_td);
      m_td = MCRYPT_FAILED;
    }
  }

  CLASSNAME_IS("mcrypt");
  // overriding ResourceData
  virtual const String& o_getClassNameHook() const { return classnameof(); }

  MCRYPT m_td;
  bool m_init;
};

typedef enum {
  RANDOM = 0,
  URANDOM,
  RAND
} iv_source;

class mcrypt_data {
public:
  std::string algorithms_dir;
  std::string modes_dir;
```

```
};
static mcrypt_data s_globals;
#define MCG(n) (s_globals.n)
#ifndef MIN
#define MIN(a, b) ((a) < (b) ? (a) : (b))
#endif
#define MCRYPT_OPEN_MODULE_FAILED(str) \
 raise_warning("%s(): Module initialization failed", str);

static Variant php_mcrypt_do_crypt(const String& cipher, const String& key,
                                   const String& data, const String& mode,
                                   const String& iv, bool dencrypt,
                                   char *name) {
  // .....
}

static Variant mcrypt_generic(const Resource& td, const String& data,
                              bool dencrypt) {
  MCrypt *pm = td.getTyped<MCrypt>();
  if (!pm->m_init) {
    raise_warning("Operation disallowed prior to mcrypt_generic_init().");
    return false;
  }

  if (data.empty()) {
    raise_warning("An empty string was passed");
    return false;
  }

  String s;
  unsigned char* data_s;
  int block_size, data_size;
  /* Check blocksize */
  if (mcrypt_enc_is_block_mode(pm->m_td) == 1) { /* It's a block algorithm */
    block_size = mcrypt_enc_get_block_size(pm->m_td);
    data_size = (((data.size() - 1) / block_size) + 1) * block_size;
    s = String(data_size, ReserveString);
    data_s = (unsigned char *)s.bufferSlice().ptr;
    memset(data_s, 0, data_size);
    memcpy(data_s, data.data(), data.size());
  } else { /* It's not a block algorithm */
    data_size = data.size();
    s = String(data_size, ReserveString);
    data_s = (unsigned char *)s.bufferSlice().ptr;
    memcpy(data_s, data.data(), data.size());
  }

  if (dencrypt) {
    mdecrypt_generic(pm->m_td, data_s, data_size);
  } else {
    mcrypt_generic(pm->m_td, data_s, data_size);
  }
  s.setSize(data_size);
  return s;
}

////////////////////////////////////////////////////////////////////////////////

Variant HHVM_FUNCTION(mcrypt_module_open, const String& algorithm,
                                          const String& algorithm_directory,
                              const String& mode, const String& mode_directory) {
```

```
116   // .....
117 }
118
119 bool HHVM_FUNCTION(mcrypt_module_close, const Resource& td) {
120   td.getTyped<MCrypt>()->close();
121   return true;
122 }
123
124 Array HHVM_FUNCTION(mcrypt_list_algorithms,
125                     const String& lib_dir /* = null_string */) {
126   // .....
127 }
128
129 Array HHVM_FUNCTION(mcrypt_list_modes,
130                     const String& lib_dir /* = null_string */) {
131   // .....
132 }
133
134 int64_t HHVM_FUNCTION(mcrypt_module_get_algo_block_size,
135                                  const String& algorithm,
136                                  const String& lib_dir /* = null_string */) {
137   String dir = lib_dir.empty() ? String(MCG(algorithms_dir)) : lib_dir;
138   return mcrypt_module_get_algo_block_size((char*)algorithm.data(),
139                                            (char*)dir.data());
140 }
141
142 int64_t HHVM_FUNCTION(mcrypt_module_get_algo_key_size, const String& algorithm,
143                                  const String& lib_dir /* = null_string */) {
144   String dir = lib_dir.empty() ? String(MCG(algorithms_dir)) : lib_dir;
145   return mcrypt_module_get_algo_key_size((char*)algorithm.data(),
146                                          (char*)dir.data());
147 }
148
149 Array HHVM_FUNCTION(mcrypt_module_get_supported_key_sizes,
150                     const String& algorithm,
151                     const String& lib_dir /* = null_string */) {
152   // .....
153 }
154
155 bool HHVM_FUNCTION(mcrypt_module_is_block_algorithm_mode, const String& mode,
156                                 const String& lib_dir /* = null_string */) {
157   String dir = lib_dir.empty() ? String(MCG(modes_dir)) : lib_dir;
158   return mcrypt_module_is_block_algorithm_mode((char*)mode.data(),
159                                                (char*)dir.data()) == 1;
160 }
161
162 bool HHVM_FUNCTION(mcrypt_module_is_block_algorithm, const String& algorithm,
163                                 const String& lib_dir /* = null_string */) {
164   String dir = lib_dir.empty() ? String(MCG(algorithms_dir)) : lib_dir;
165   return mcrypt_module_is_block_algorithm((char*)algorithm.data(),
166                                           (char*)dir.data()) == 1;
167 }
168
169 bool HHVM_FUNCTION(mcrypt_module_is_block_mode, const String& mode,
170                                  const String& lib_dir /* = null_string */) {
171   String dir = lib_dir.empty() ? String(MCG(modes_dir)) : lib_dir;
172   return mcrypt_module_is_block_mode((char*)mode.data(),
173                                      (char*)dir.data()) == 1;
174 }
175
176 bool HHVM_FUNCTION(mcrypt_module_self_test, const String& algorithm,
```

```
                                const String& lib_dir /* = null_string */) {
  String dir = lib_dir.empty() ? String(MCG(algorithms_dir)) : lib_dir;
  return mcrypt_module_self_test((char*)algorithm.data(),
                                 (char*)dir.data()) == 0;
}

Variant HHVM_FUNCTION(mcrypt_create_iv, int size, int source /* = 0 */) {
  if (size <= 0 || size >= INT_MAX) {
    raise_warning("Can not create an IV with a size of less than 1 or "
                  "greater than %d", INT_MAX);
    return false;
  }

  int n = 0;
  char *iv = (char*)calloc(size + 1, 1);
  if (source == RANDOM || source == URANDOM) {
    int fd = open(source == RANDOM ? "/dev/random" : "/dev/urandom", O_RDONLY);
    if (fd < 0) {
      free(iv);
      raise_warning("Cannot open source device");
      return false;
    }
    int read_bytes;
    for (read_bytes = 0; read_bytes < size && n >= 0; read_bytes += n) {
      n = read(fd, iv + read_bytes, size - read_bytes);
    }
    n = read_bytes;
    close(fd);
    if (n < size) {
      free(iv);
      raise_warning("Could not gather sufficient random data");
      return false;
    }
  } else {
    n = size;
    while (size) {
      iv[--size] = (char)(255.0 * rand() / RAND_MAX);
    }
  }
  return String(iv, n, AttachString);
}
```

(https://github.com/2longAGO/Projet_Synth/blob/main/ext_mcrypt.cpp)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal5.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/536/qid/16690) == "2"))

### Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

◯ oui
◯ non

### Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [r98q0]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

## Il y a-t-il une erreur dans ce code? (dh_IESCipher.java)

Vous pouvez employer n'importe quelle stratégie pour répondre à cette question. Cliquer sur l'image vous donne accès au fichier du code.

**{if(is_empty(randVal.NAOK),rand(1,2),randVal.NAOK)}**

```java
package org.bouncycastle.jcajce.provider.asymmetric.dh;
import java.io.ByteArrayOutputStream;
import java.security.AlgorithmParameters;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.Key;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.spec.AlgorithmParameterSpec;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.CipherSpi;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.ShortBufferException;
import javax.crypto.interfaces.DHKey;
import javax.crypto.interfaces.DHPrivateKey;
import javax.crypto.interfaces.DHPublicKey;

import org.bouncycastle.crypto.BlockCipher;
import org.bouncycastle.crypto.InvalidCipherTextException;
import org.bouncycastle.crypto.KeyEncoder;
import org.bouncycastle.crypto.agreement.DHBasicAgreement;
import org.bouncycastle.crypto.digests.SHA1Digest;
import org.bouncycastle.crypto.engines.AESEngine;
import org.bouncycastle.crypto.engines.DESedeEngine;
import org.bouncycastle.crypto.engines.IESEngine;
import org.bouncycastle.crypto.engines.OldIESEngine;
import org.bouncycastle.crypto.generators.DHKeyPairGenerator;
import org.bouncycastle.crypto.generators.EphemeralKeyPairGenerator;
import org.bouncycastle.crypto.generators.KDF2BytesGenerator;
import org.bouncycastle.crypto.macs.HMac;
import org.bouncycastle.crypto.paddings.PaddedBufferedBlockCipher;
import org.bouncycastle.crypto.params.AsymmetricKeyParameter;
import org.bouncycastle.crypto.params.DHKeyGenerationParameters;
import org.bouncycastle.crypto.params.DHKeyParameters;
import org.bouncycastle.crypto.params.DHParameters;
import org.bouncycastle.crypto.params.DHPublicKeyParameters;
import org.bouncycastle.crypto.params.IESParameters;
import org.bouncycastle.crypto.params.IESWithCipherParameters;
import org.bouncycastle.crypto.parsers.DHIESPublicKeyParser;
import org.bouncycastle.jcajce.provider.asymmetric.util.DHUtil;
import org.bouncycastle.jcajce.provider.asymmetric.util.IESUtil;
import org.bouncycastle.jcajce.util.BCJcaJceHelper;
import org.bouncycastle.jcajce.util.JcaJceHelper;
import org.bouncycastle.jce.interfaces.IESKey;
import org.bouncycastle.jce.spec.IESParameterSpec;
import org.bouncycastle.util.BigIntegers;
import org.bouncycastle.util.Strings;
public class IESCipher
    extends CipherSpi
{
    private final JcaJceHelper helper = new BCJcaJceHelper();

    private IESEngine engine;
    private int state = -1;
    private ByteArrayOutputStream buffer = new ByteArrayOutputStream();
    private AlgorithmParameters engineParam = null;
    private IESParameterSpec engineSpec = null;
    private AsymmetricKeyParameter key;
    private SecureRandom random;
    private boolean dhaesMode = false;
    private AsymmetricKeyParameter otherKeyParameter = null;
    public IESCipher(IESEngine engine)
    {
        this.engine = engine;
    }

    public IESCipher(OldIESEngine engine)
    {
        this.engine = engine;
    }

    public int engineGetBlockSize()
    {
        if (engine.getCipher() != null)
```

```
        {
            return engine.getCipher().getBlockSize();
        }
        else
        {
            return 0;
        }
    }
    public int engineGetKeySize(Key key)
    {
        if (key instanceof DHKey)
        {
            return ((DHKey)key).getParams().getP().bitLength();
        }
        else
        {
            throw new IllegalArgumentException("not a DH key");
        }
    }

    public byte[] engineGetIV()
    {
        return null;
    }
    public AlgorithmParameters engineGetParameters()
    {
        if (engineParam == null && engineSpec != null)
        {
            try
            {
                engineParam = helper.createAlgorithmParameters("IES");
                engineParam.init(engineSpec);
            }
            catch (Exception e)
            {
                throw new RuntimeException(e.toString());
            }
        }
        return engineParam;
    }
    public void engineSetMode(String mode)
        throws NoSuchAlgorithmException
    {
        String modeName = Strings.toUpperCase(mode);
        if (modeName.equals("NONE"))
        {
            dhaesMode = false;
        }
        else if (modeName.equals("DHAES"))
        {
            dhaesMode = true;
        }
        else
        {
            throw new IllegalArgumentException("can't support mode " + mode);
        }
    }
    public int engineGetOutputSize(int inputLen)
    {
        int len1, len2, len3;
        if (key == null)
        {
            throw new IllegalStateException("cipher not initialised");
        }
        len1 = engine.getMac().getMacSize();
        if (otherKeyParameter == null)
        {
            len2 = 1 + 2 * (((DHKeyParameters)key).getParameters().getP().bitLength() + 7) / 8;
        }
        else
        {
            len2 = 0;
        }
        if (engine.getCipher() == null)
        {
            len3 = inputLen;
        }
        else if (state == Cipher.ENCRYPT_MODE || state == Cipher.WRAP_MODE)
        {
            len3 = engine.getCipher().getOutputSize(inputLen);
        }
        else if (state == Cipher.DECRYPT_MODE || state == Cipher.UNWRAP_MODE)
        {
            len3 = engine.getCipher().getOutputSize(inputLen - len1 - len2);
        }
        else
```

```
        {
            throw new IllegalStateException("cipher not initialised");
        }
        if (state == Cipher.ENCRYPT_MODE || state == Cipher.WRAP_MODE)
        {
            return buffer.size() + len1 + len2 + len3;
        }
        else if (state == Cipher.DECRYPT_MODE || state == Cipher.UNWRAP_MODE)
        {
            return buffer.size() - len1 - len2 + len3;
        }
        else
        {
            throw new IllegalStateException("IESCipher not initialised");
        }
    }
    public void engineSetPadding(String padding)
        throws NoSuchPaddingException
    {
        String paddingName = Strings.toUpperCase(padding);
        // TDOD: make this meaningful...
        if (paddingName.equals("NOPADDING"))
        {
        }
        else if (paddingName.equals("PKCS5PADDING") || paddingName.equals("PKCS7PADDING"))
        {
        }
        else
        {
            throw new NoSuchPaddingException("padding not available with IESCipher");
        }
    }
    // Initialisation methods
    public void engineInit(
        int opmode,
        Key key,
        AlgorithmParameters params,
        SecureRandom random)
        throws InvalidKeyException, InvalidAlgorithmParameterException
    {
        AlgorithmParameterSpec paramSpec = null;
        if (params != null)
        {
            try
            {
                paramSpec = params.getParameterSpec(IESParameterSpec.class);
            }
            catch (Exception e)
            {
                throw new InvalidAlgorithmParameterException("cannot recognise parameters: " + e.toString());
            }
        }
        engineParam = params;
        engineInit(opmode, key, paramSpec, random);
    }
    public void engineInit(
        int opmode,
        Key key,
        AlgorithmParameterSpec engineSpec,
        SecureRandom random)
        throws InvalidAlgorithmParameterException, InvalidKeyException
    {
        // Use default parameters (including cipher key size) if none are specified
        if (engineSpec == null)
        {
            this.engineSpec = IESUtil.guessParameterSpec(engine.getCipher());
        }
        else if (engineSpec instanceof IESParameterSpec)
        {
            this.engineSpec = (IESParameterSpec)engineSpec;
        }
        else
        {
            throw new InvalidAlgorithmParameterException("must be passed IES parameters");
        }

        // Parse the recipient's key
        if (opmode == Cipher.ENCRYPT_MODE || opmode == Cipher.WRAP_MODE)
        {
            if (key instanceof DHPublicKey)
            {
                this.key = DHUtil.generatePublicKeyParameter((PublicKey)key);
            }
            else if (key instanceof IESKey)
            {
```

```
                IESKey ieKey = (IESKey)key;
                this.key = DHUtil.generatePublicKeyParameter(ieKey.getPublic());
                this.otherKeyParameter = DHUtil.generatePrivateKeyParameter(ieKey.getPrivate());
            }
            else
            {
                throw new InvalidKeyException("must be passed recipient's public DH key for encryption");
            }
        }
        else if (opmode == Cipher.DECRYPT_MODE || opmode == Cipher.UNWRAP_MODE)
        {
            if (key instanceof DHPrivateKey)
            {
                this.key = DHUtil.generatePrivateKeyParameter((PrivateKey)key);
            }
            else if (key instanceof IESKey)
            {
                IESKey ieKey = (IESKey)key;
                this.otherKeyParameter = DHUtil.generatePublicKeyParameter(ieKey.getPublic());
                this.key = DHUtil.generatePrivateKeyParameter(ieKey.getPrivate());
            }
            else
            {
                throw new InvalidKeyException("must be passed recipient's private DH key for decryption");
            }
        }
        else
        {
            throw new InvalidKeyException("must be passed EC key");
        }
        this.random = random;
        this.state = opmode;
        buffer.reset();
    }
    public void engineInit(
        int opmode,
        Key key,
        SecureRandom random)
        throws InvalidKeyException
    {
        try
        {
            engineInit(opmode, key, (AlgorithmParameterSpec)null, random);
        }
        catch (InvalidAlgorithmParameterException e)
        {
            throw new IllegalArgumentException("can't handle supplied parameter spec");
        }

    }
    // Update methods - buffer the input
    public byte[] engineUpdate(
        byte[] input,
        int inputOffset,
        int inputLen)
    {
        buffer.write(input, inputOffset, inputLen);
        return null;
    }
    public int engineUpdate(
        byte[] input,
        int inputOffset,
        int inputLen,
        byte[] output,
        int outputOffset)
    {
        buffer.write(input, inputOffset, inputLen);
        return 0;
    }
    // Finalisation methods
    public byte[] engineDoFinal(
        byte[] input,
        int inputOffset,
        int inputLen)
        throws IllegalBlockSizeException, BadPaddingException
    {
        if (inputLen != 0)
        {
            buffer.write(input, inputOffset, inputLen);
        }
        byte[] in = buffer.toByteArray();
        buffer.reset();

        // Convert parameters for use in IESEngine
        IESParameters params = new IESWithCipherParameters(engineSpec.getDerivationV(),
            engineSpec.getEncodingV(),
```

```
            engineSpec.getMacKeySize(),
            engineSpec.getCipherKeySize());

        DHParameters dhParams = ((DHKeyParameters)key).getParameters();

        byte[] V;
        if (otherKeyParameter != null)
        {
            try
            {
                if (state == Cipher.ENCRYPT_MODE || state == Cipher.WRAP_MODE)
                {
                    engine.init(true, otherKeyParameter, key, params);
                }
                else
                {
                    engine.init(false, key, otherKeyParameter, params);
                }
                return engine.processBlock(in, 0, in.length);
            }
            catch (Exception e)
            {
                throw new BadPaddingException(e.getMessage());
            }
        }
        if (state == Cipher.ENCRYPT_MODE || state == Cipher.WRAP_MODE)
        {
            // Generate the ephemeral key pair
            DHKeyPairGenerator gen = new DHKeyPairGenerator();
            gen.init(new DHKeyGenerationParameters(random, dhParams));
            EphemeralKeyPairGenerator kGen = new EphemeralKeyPairGenerator(gen, new KeyEncoder()
            {
                public byte[] getEncoded(AsymmetricKeyParameter keyParameter)
                {
                    byte[] Vloc = new byte[(((DHKeyParameters)keyParameter).getParameters().getP().bitLength() + 7) / 8];
                    byte[] Vtmp = BigIntegers.asUnsignedByteArray(((DHPublicKeyParameters)keyParameter).getY());
                    if (Vtmp.length > Vloc.length)
                    {
                        throw new IllegalArgumentException("Senders's public key longer than expected.");
                    }
                    else
                    {
                        System.arraycopy(Vtmp, 0, Vloc, Vloc.length - Vtmp.length, Vtmp.length);
                    }
                    return Vloc;
                }
            });
            // Encrypt the buffer
            try
            {
                engine.init(key, params, kGen);
                return engine.processBlock(in, 0, in.length);
            }
            catch (Exception e)
            {
                throw new BadPaddingException(e.getMessage());
            }
        }
        else if (state == Cipher.DECRYPT_MODE || state == Cipher.UNWRAP_MODE)
        {
            // Decrypt the buffer
            try
            {
                engine.init(key, params, new DHIESPublicKeyParser(((DHKeyParameters)key).getParameters()));
                return engine.processBlock(in, 0, in.length);
            }
            catch (InvalidCipherTextException e)
            {
                throw new BadPaddingException(e.getMessage());
            }
        }
        else
        {
            throw new IllegalStateException("IESCipher not initialised");
        }
    }
    public int engineDoFinal(
        byte[] input,
        int inputOffset,
        int inputLength,
        byte[] output,
        int outputOffset)
        throws ShortBufferException, IllegalBlockSizeException, BadPaddingException
    {
        byte[] buf = engineDoFinal(input, inputOffset, inputLength);
        System.arraycopy(buf, 0, output, outputOffset, buf.length);
```

```
422         return buf.length;
423     }
424     /**
425      * Classes that inherit from us
426      */
427     static public class IES
```

(https://github.com/2longAGO/Projet_Synth/blob/main/dh_IESCipher.java)

MEDIUM: Potentially Unsafe Code - AESEngine

Line: 27 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

This class defaults to ECB mode, which is unsafe.

```
import org.bouncycastle.crypto.engines.AESEngine;
```

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 52 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

```
public class IESCipher
```

LOW: Potentially Unsafe Code - Operation on Primitive Data Type

Line: 58 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The code appears to be carrying out a mathematical operation on a primitive data type. In some circumstances this can result in an overflow and unexpected behaviour. Check the code manually to determine the risk.

```
private int state = -1;
```

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 427 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)

\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class IES

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 438 - D:\school_shit\old_project\CryptographicVulnerabilities \CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2) \CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class IESwithDESede

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 450 - D:\school_shit\old_project\CryptographicVulnerabilities \CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2) \CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class IESwithAES

MEDIUM: Potentially Unsafe Code - AESEngine

Line: 458 - D:\school_shit\old_project\CryptographicVulnerabilities

\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

This class defaults to ECB mode, which is unsafe.

new PaddedBufferedBlockCipher(new AESEngine())));

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 465 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class OldIESwithCipher

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 477 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class OldIES

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 488 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It

is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class OldIESwithDESede

POTENTIAL ISSUE: Potentially Unsafe Code - Public Class Not Declared as Final

Line: 497 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

The class is not declared as final as per OWASP recommendations. It is considered best practice to make classes final where possible and practical (i.e. It has no classes which inherit from it). Non-Final classes can allow an attacker to extend a class in a malicious manner. Manually inspect the code to determine whether or not it is practical to make this class final.

static public class OldIESwithAES

MEDIUM: Potentially Unsafe Code - AESEngine

Line: 502 - D:\school_shit\old_project\CryptographicVulnerabilities\CryptographicVulnerabilities_CVE_Analysis\InsecureDefaults(2)\CVE-2016-1000344&1000352\bugged\dh_IESCipher.java

This class defaults to ECB mode, which is unsafe.

super(new AESEngine());

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/455/qid/16171) == "1"))

Ce qui se trouve à la fin du texte sont les messages d'erreurs obtenues d'une analyse statique du code présenté.

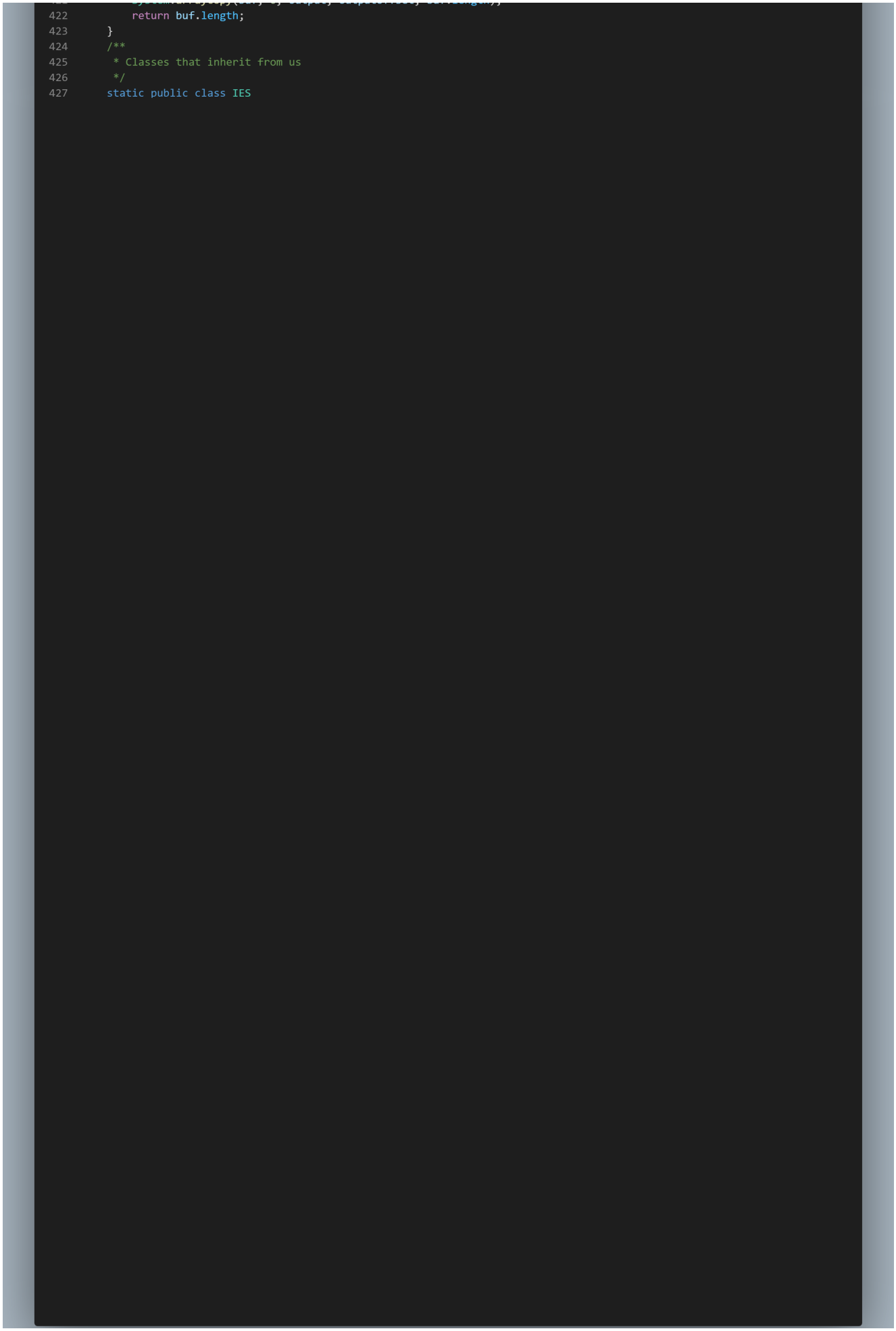
```java
package org.bouncycastle.jcajce.provider.asymmetric.dh;
import java.io.ByteArrayOutputStream;
import java.security.AlgorithmParameters;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.Key;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.spec.AlgorithmParameterSpec;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.CipherSpi;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.ShortBufferException;
import javax.crypto.interfaces.DHKey;
import javax.crypto.interfaces.DHPrivateKey;
import javax.crypto.interfaces.DHPublicKey;

import org.bouncycastle.crypto.BlockCipher;
import org.bouncycastle.crypto.InvalidCipherTextException;
import org.bouncycastle.crypto.KeyEncoder;
import org.bouncycastle.crypto.agreement.DHBasicAgreement;
import org.bouncycastle.crypto.digests.SHA1Digest;
import org.bouncycastle.crypto.engines.AESEngine;
import org.bouncycastle.crypto.engines.DESedeEngine;
import org.bouncycastle.crypto.engines.IESEngine;
import org.bouncycastle.crypto.engines.OldIESEngine;
import org.bouncycastle.crypto.generators.DHKeyPairGenerator;
import org.bouncycastle.crypto.generators.EphemeralKeyPairGenerator;
import org.bouncycastle.crypto.generators.KDF2BytesGenerator;
import org.bouncycastle.crypto.macs.HMac;
import org.bouncycastle.crypto.paddings.PaddedBufferedBlockCipher;
import org.bouncycastle.crypto.params.AsymmetricKeyParameter;
import org.bouncycastle.crypto.params.DHKeyGenerationParameters;
import org.bouncycastle.crypto.params.DHKeyParameters;
import org.bouncycastle.crypto.params.DHParameters;
import org.bouncycastle.crypto.params.DHPublicKeyParameters;
import org.bouncycastle.crypto.params.IESParameters;
import org.bouncycastle.crypto.params.IESWithCipherParameters;
import org.bouncycastle.crypto.parsers.DHIESPublicKeyParser;
import org.bouncycastle.jcajce.provider.asymmetric.util.DHUtil;
import org.bouncycastle.jcajce.provider.asymmetric.util.IESUtil;
import org.bouncycastle.jcajce.util.BCJcaJceHelper;
import org.bouncycastle.jcajce.util.JcaJceHelper;
import org.bouncycastle.jce.interfaces.IESKey;
import org.bouncycastle.jce.spec.IESParameterSpec;
import org.bouncycastle.util.BigIntegers;
import org.bouncycastle.util.Strings;
public class IESCipher
    extends CipherSpi
{
    private final JcaJceHelper helper = new BCJcaJceHelper();

    private IESEngine engine;
    private int state = -1;
    private ByteArrayOutputStream buffer = new ByteArrayOutputStream();
    private AlgorithmParameters engineParam = null;
    private IESParameterSpec engineSpec = null;
    private AsymmetricKeyParameter key;
    private SecureRandom random;
    private boolean dhaesMode = false;
    private AsymmetricKeyParameter otherKeyParameter = null;
    public IESCipher(IESEngine engine)
    {
        this.engine = engine;
    }

    public IESCipher(OldIESEngine engine)
    {
        this.engine = engine;
    }

    public int engineGetBlockSize()
    {
        if (engine.getCipher() != null)
```

```
        {
            return engine.getCipher().getBlockSize();
        }
        else
        {
            return 0;
        }
    }
    public int engineGetKeySize(Key key)
    {
        if (key instanceof DHKey)
        {
            return ((DHKey)key).getParams().getP().bitLength();
        }
        else
        {
            throw new IllegalArgumentException("not a DH key");
        }
    }

    public byte[] engineGetIV()
    {
        return null;
    }
    public AlgorithmParameters engineGetParameters()
    {
        if (engineParam == null && engineSpec != null)
        {
            try
            {
                engineParam = helper.createAlgorithmParameters("IES");
                engineParam.init(engineSpec);
            }
            catch (Exception e)
            {
                throw new RuntimeException(e.toString());
            }
        }
        return engineParam;
    }
    public void engineSetMode(String mode)
        throws NoSuchAlgorithmException
    {
        String modeName = Strings.toUpperCase(mode);
        if (modeName.equals("NONE"))
        {
            dhaesMode = false;
        }
        else if (modeName.equals("DHAES"))
        {
            dhaesMode = true;
        }
        else
        {
            throw new IllegalArgumentException("can't support mode " + mode);
        }
    }
    public int engineGetOutputSize(int inputLen)
    {
        int len1, len2, len3;
        if (key == null)
        {
            throw new IllegalStateException("cipher not initialised");
        }
        len1 = engine.getMac().getMacSize();
        if (otherKeyParameter == null)
        {
            len2 = 1 + 2 * (((DHKeyParameters)key).getParameters().getP().bitLength() + 7) / 8;
        }
        else
        {
            len2 = 0;
        }
        if (engine.getCipher() == null)
        {
            len3 = inputLen;
        }
        else if (state == Cipher.ENCRYPT_MODE || state == Cipher.WRAP_MODE)
        {
            len3 = engine.getCipher().getOutputSize(inputLen);
        }
        else if (state == Cipher.DECRYPT_MODE || state == Cipher.UNWRAP_MODE)
        {
            len3 = engine.getCipher().getOutputSize(inputLen - len1 - len2);
        }
        else
```

```
        {
            throw new IllegalStateException("cipher not initialised");
        }
        if (state == Cipher.ENCRYPT_MODE || state == Cipher.WRAP_MODE)
        {
            return buffer.size() + len1 + len2 + len3;
        }
        else if (state == Cipher.DECRYPT_MODE || state == Cipher.UNWRAP_MODE)
        {
            return buffer.size() - len1 - len2 + len3;
        }
        else
        {
            throw new IllegalStateException("IESCipher not initialised");
        }
    }
    public void engineSetPadding(String padding)
        throws NoSuchPaddingException
    {
        String paddingName = Strings.toUpperCase(padding);
        // TDOD: make this meaningful...
        if (paddingName.equals("NOPADDING"))
        {
        }
        else if (paddingName.equals("PKCS5PADDING") || paddingName.equals("PKCS7PADDING"))
        {
        }
        else
        {
            throw new NoSuchPaddingException("padding not available with IESCipher");
        }
    }
    // Initialisation methods
    public void engineInit(
        int opmode,
        Key key,
        AlgorithmParameters params,
        SecureRandom random)
        throws InvalidKeyException, InvalidAlgorithmParameterException
    {
        AlgorithmParameterSpec paramSpec = null;
        if (params != null)
        {
            try
            {
                paramSpec = params.getParameterSpec(IESParameterSpec.class);
            }
            catch (Exception e)
            {
                throw new InvalidAlgorithmParameterException("cannot recognise parameters: " + e.toString());
            }
        }
        engineParam = params;
        engineInit(opmode, key, paramSpec, random);
    }
    public void engineInit(
        int opmode,
        Key key,
        AlgorithmParameterSpec engineSpec,
        SecureRandom random)
        throws InvalidAlgorithmParameterException, InvalidKeyException
    {
        // Use default parameters (including cipher key size) if none are specified
        if (engineSpec == null)
        {
            this.engineSpec = IESUtil.guessParameterSpec(engine.getCipher());
        }
        else if (engineSpec instanceof IESParameterSpec)
        {
            this.engineSpec = (IESParameterSpec)engineSpec;
        }
        else
        {
            throw new InvalidAlgorithmParameterException("must be passed IES parameters");
        }

        // Parse the recipient's key
        if (opmode == Cipher.ENCRYPT_MODE || opmode == Cipher.WRAP_MODE)
        {
            if (key instanceof DHPublicKey)
            {
                this.key = DHUtil.generatePublicKeyParameter((PublicKey)key);
            }
            else if (key instanceof IESKey)
            {
```

```
                IESKey ieKey = (IESKey)key;
                this.key = DHUtil.generatePublicKeyParameter(ieKey.getPublic());
                this.otherKeyParameter = DHUtil.generatePrivateKeyParameter(ieKey.getPrivate());
            }
            else
            {
                throw new InvalidKeyException("must be passed recipient's public DH key for encryption");
            }
        }
        else if (opmode == Cipher.DECRYPT_MODE || opmode == Cipher.UNWRAP_MODE)
        {
            if (key instanceof DHPrivateKey)
            {
                this.key = DHUtil.generatePrivateKeyParameter((PrivateKey)key);
            }
            else if (key instanceof IESKey)
            {
                IESKey ieKey = (IESKey)key;
                this.otherKeyParameter = DHUtil.generatePublicKeyParameter(ieKey.getPublic());
                this.key = DHUtil.generatePrivateKeyParameter(ieKey.getPrivate());
            }
            else
            {
                throw new InvalidKeyException("must be passed recipient's private DH key for decryption");
            }
        }
        else
        {
            throw new InvalidKeyException("must be passed EC key");
        }
        this.random = random;
        this.state = opmode;
        buffer.reset();
    }
    public void engineInit(
        int opmode,
        Key key,
        SecureRandom random)
        throws InvalidKeyException
    {
        try
        {
            engineInit(opmode, key, (AlgorithmParameterSpec)null, random);
        }
        catch (InvalidAlgorithmParameterException e)
        {
            throw new IllegalArgumentException("can't handle supplied parameter spec");
        }

    }
    // Update methods - buffer the input
    public byte[] engineUpdate(
        byte[] input,
        int inputOffset,
        int inputLen)
    {
        buffer.write(input, inputOffset, inputLen);
        return null;
    }
    public int engineUpdate(
        byte[] input,
        int inputOffset,
        int inputLen,
        byte[] output,
        int outputOffset)
    {
        buffer.write(input, inputOffset, inputLen);
        return 0;
    }
    // Finalisation methods
    public byte[] engineDoFinal(
        byte[] input,
        int inputOffset,
        int inputLen)
        throws IllegalBlockSizeException, BadPaddingException
    {
        if (inputLen != 0)
        {
            buffer.write(input, inputOffset, inputLen);
        }
        byte[] in = buffer.toByteArray();
        buffer.reset();

        // Convert parameters for use in IESEngine
        IESParameters params = new IESWithCipherParameters(engineSpec.getDerivationV(),
            engineSpec.getEncodingV(),
```

```
            engineSpec.getMacKeySize(),
            engineSpec.getCipherKeySize());

        DHParameters dhParams = ((DHKeyParameters)key).getParameters();

        byte[] V;
        if (otherKeyParameter != null)
        {
            try
            {
                if (state == Cipher.ENCRYPT_MODE || state == Cipher.WRAP_MODE)
                {
                    engine.init(true, otherKeyParameter, key, params);
                }
                else
                {
                    engine.init(false, key, otherKeyParameter, params);
                }
                return engine.processBlock(in, 0, in.length);
            }
            catch (Exception e)
            {
                throw new BadPaddingException(e.getMessage());
            }
        }
        if (state == Cipher.ENCRYPT_MODE || state == Cipher.WRAP_MODE)
        {
            // Generate the ephemeral key pair
            DHKeyPairGenerator gen = new DHKeyPairGenerator();
            gen.init(new DHKeyGenerationParameters(random, dhParams));
            EphemeralKeyPairGenerator kGen = new EphemeralKeyPairGenerator(gen, new KeyEncoder()
            {
                public byte[] getEncoded(AsymmetricKeyParameter keyParameter)
                {
                    byte[] Vloc = new byte[(((DHKeyParameters)keyParameter).getParameters().getP().bitLength() + 7) / 8];
                    byte[] Vtmp = BigIntegers.asUnsignedByteArray(((DHPublicKeyParameters)keyParameter).getY());
                    if (Vtmp.length > Vloc.length)
                    {
                        throw new IllegalArgumentException("Senders's public key longer than expected.");
                    }
                    else
                    {
                        System.arraycopy(Vtmp, 0, Vloc, Vloc.length - Vtmp.length, Vtmp.length);
                    }
                    return Vloc;
                }
            });
            // Encrypt the buffer
            try
            {
                engine.init(key, params, kGen);
                return engine.processBlock(in, 0, in.length);
            }
            catch (Exception e)
            {
                throw new BadPaddingException(e.getMessage());
            }
        }
        else if (state == Cipher.DECRYPT_MODE || state == Cipher.UNWRAP_MODE)
        {
            // Decrypt the buffer
            try
            {
                engine.init(key, params, new DHIESPublicKeyParser(((DHKeyParameters)key).getParameters()));
                return engine.processBlock(in, 0, in.length);
            }
            catch (InvalidCipherTextException e)
            {
                throw new BadPaddingException(e.getMessage());
            }
        }
        else
        {
            throw new IllegalStateException("IESCipher not initialised");
        }
    }
    public int engineDoFinal(
        byte[] input,
        int inputOffset,
        int inputLength,
        byte[] output,
        int outputOffset)
        throws ShortBufferException, IllegalBlockSizeException, BadPaddingException
    {
        byte[] buf = engineDoFinal(input, inputOffset, inputLength);
        System.arraycopy(buf, 0, output, outputOffset, buf.length);
```



```
422         return buf.length;
423     }
424     /**
425      * Classes that inherit from us
426      */
427     static public class IES
```

(https://github.com/2longAGO/Projet_Synth/blob/main/dh_IESCipher.java)

Répondre à cette question seulement si les conditions suivantes sont réunies :
((randVal.NAOK (/index.php/questionAdministration/view/surveyid/814556/gid/455/qid/16171) == "2"))

## Avez-vous trouvé une erreur? *

Veuillez sélectionner une seule des propositions suivantes :

- ◯ oui
- ◯ non

## Donner une courte description de l'erreur trouvé. *

Répondre à cette question seulement si les conditions suivantes sont réunies :
La réponse était 'oui' à la question ' [G02Q04]' (Avez-vous trouvé une erreur?)

Veuillez écrire votre réponse ici :

Envoyer votre questionnaire.
Merci d'avoir complété ce questionnaire.