高一上學期

多元選修課程

資訊安全與概論

の學習紀錄

聶宜帆

目錄

—課前感想-為什麼選擇這門課—

—9/15編碼與解碼—

—9/22密碼學概論—

—10/13隱寫術—

—10/20 Linux簡介—

—10/27高中跨校資安實務素養挑戰賽-個人賽—

—11/03人工智慧AI簡介—

—11/10高中跨校資安實務素養挑戰賽-團體賽—

—11/17模型逆向攻擊MNIST訓練模型—

—11/24 SQL Injection學習與實作—

—12/01封包攔截-Burp Suite攔截—

—12/15資安推廣影片與期末報告海報討論—

為什麼選擇這門課

「編碼與解碼」、「現代密碼學」、「網路封包實戰」，課程大綱映入我眼簾。小時候，應該沒有人沒幻想過自己使用神秘的摩斯密碼與人溝通，或是當個網路上帥氣的駭客。現在的我，看著眼前的「攻擊實務演練」，想像著與駭客的程式戰，心中默默憧憬著--或許這門課能讓我真正踏進過去只存在於想像中的世界。

那些看似複雜的技術名詞，背後究竟藏著什麼秘密？駭客的攻擊是如何被發現、被阻擋的？比起電影裡幾秒鐘完成的情節，我更好奇這些攻擊背後所運用的原理與邏輯。

帶著這種半期待、半好奇的心情，我選擇了這門課。希望在未來的課堂上，能逐漸看懂那些在我眼中原本像魔法般的指令與工具，理解它們的用途與意義，看見資安世界真正的樣貌。

這門課對我來說是學習一項未知的技術，也是一個讓我接觸另一種思考方式的入口。我期待能在這堂課裡，將那些曾經只存在於幻想中的酷炫場景，一步步化為真實。

09/15

編碼與解碼

- ASCLL學習
- Morse Code摩斯密碼學習
- Base64學習

ASCII學習

-將「HELLO」編碼

-將「66 114 101 97 107 65 76 76 67 84 70 123 65 109 118 48 117 68 121 101 114 118 80 116 109 86 114 57 83 83 83 75 125」解碼

使用的網站： https://www.aperisolve.com/

Morse code摩斯密碼學習

-將「I love you」編碼

-將「.. -. ..-. --- ... . -.-. ..-. .-.. .- --. .. ... -- --- .-. ... .. -. --.」解碼

使用的網站： https://www.aperisolve.com/

Base64學習

Man的Base64編碼

文字	M								a								n							
ASCII編碼	77								97								110							
位元	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
索引	19								22				5				46							
Base64編碼	T								W				F				u							

-將字串「BreakallCTF{happyhackinghighhaaha}」轉換成 Base64

BreakallCTF{happyhackinghighhaaha}

清空 加密 解密 解密為 UTF-8字符

QnJlYWthbGxDVEZ7aGFwcHloYWNraW5naGlnaGhhYWhhfQ==

複製

-將Base64 字串「QnJlYWtBTExDVEZ7NTN1c1pRM2hXVzI1ZGNoWjdkWGV9」解碼

QnJlYWtBTExDVEZ7NTN1c1pRM2hXVzI1ZGNoWjdkWGV9

清空 加密 解密 解密為 UTF-8字符

BreakALLCTF{53usZQ3hWW25dchZ7dXe}

複製

使用的網站： https://www.aperisolve.com/

利用Python解碼

將下列十六進位編碼字串解碼：
6236343a20615735305a584a755a58526659323975646d567963326c76626c
3930623239736331397962324e72

10 分

您的回答

10 分

請將前一題結果再進行解碼

您的回答

[2]
0 秒

```
import base64
hex_string = '6236343a20615735305a584a755a58526659323975646d567963326c76626c3930623239736331397962324e72'
decoded_bytes = bytes.fromhex(hex_string)
decoded_string = decoded_bytes.decode('utf-8')
print(decoded_string)
```

```
b64: aW50ZXJuZXRfY29udmVyc2lvbl90b29sc19yb2Nr
```

[4]
0 秒

```
import base64
base64_string = 'aW50ZXJuZXRfY29udmVyc2lvbl90b29sc19yb2Nr'
decoded_bytes = base64.b64decode(base64_string)
decoded_string = decoded_bytes.decode('utf-8')
print(decoded_string)
```

```
internet_conversion_tools_rock
```

△利用Chat gpt寫出程式後再使用google AI更正錯誤

09/22

密碼學概論

- 密碼學學習
  - 使用軟體進行訊息加密
  - 使用該軟體解題

密碼學學習

現代密碼學分類 (type of key)

對稱式密碼
symmetric key algorithms
Private-key cryptography

密鑰

加密與解密都是使用同一把key

https://en.wikipedia.org/wiki/Cipher

非對稱式密碼
asymmetric key algorithms
Public-key cryptography

https://en.wikipedia.org/wiki/Outline_of_cryptography

危險的對稱式密碼 - XOR ciphers

- 使用XOR 邏輯運算建構簡易的密碼(容易被破)

A	B	A AND B	A OR B	A ⊕ B	(A ⊕ B)⊕ B	(A ⊕ B)⊕ A
0	0	0	0	0	0	0
0	1	0	1	1	0	1
1	0	0	1	1	1	0
1	1	1	1	0	1	1

⊕ Boolean exclusive-OR operation

-使用軟體進行訊息加密

Message to encrypt:
Hello
Caesar Output:
Khoor
5 characters, 1 line

Message to encrypt:
Hello
AES Output:
CA 14 5D 08 CD E6 31 6D 1C 6E 6A 7F E4 7E 81 27
47 characters, 1 line

-使用該軟體解題

請使用XOR加密以下內容， * 10 分

Cryptography is wonderful.

key為

99 AA BB CC DD EE FF

答案為

DA D8 C2 BC A9 81 98 EB CB CB A4 A4 CE 96

Message to encrypt/decrypt:

Cryptography is wonderful.

XOR Output:

DA D8 C2 BC A9 81 98 EB CB CB A4 A4 CE 96 EA 8A CC A3 B3 8A 9A EB CC CE A0 F3

77 characters, 1 line

請使用RSA解密以下內容，N=299, e=23 * 10 分

60 00 C1 00 3E 00 F9 00 76 00 93 00 C1 00 02 01 FD 00 4B 00 56 00 56 00 93 00 74 00 A0 00 69 00 7A 00 C1 00 3D 00 93 00 4B 00 C1 00 02 01 3E 00 67 00 F9 00 A0 00 69 00 74 00 68 00 56 00 FD 00 B8

答案為

I love asymmetric key algorithms.

Plaintext:

I love asymmetric key algorithms.

33 characters, 1 line

Decrypted Text:

10/13

隱寫術

-使用網站解密解密隱藏在片中的文字

Decode image

To decode a hidden message from an image, just choose an image and hit the **Decode** button.

Neither the image nor the message that has been hidden will be at any moment transmitted over the web, all the magic happens within your browser.

選擇檔案 noble_images.png

Decode

Hidden message

CTF{I love noble wilds}□□□

-使用網站解密解密隱藏在片中的文字 / 照片

使用的網站： https://www.aperisolve.com/

10/20

Linux 簡介

-使用CDX Lunux模擬環境

-練習各種指令

△練習「ls -al」

△練習「hostname -s」

使用的網站： https://cdx.nchc.org.tw

-使用picoCTF解題

▽題目

▽利用linux解出答案

```
tfg11431037-picoctf@webshell:~$ wget https://mercury.picoctf.net/static/2d24d50b4ebed90c704575627f1f57b2/flag
--2025-10-20 04:03:26--  https://mercury.picoctf.net/static/2d24d50b4ebed90c704575627f1f57b2/flag
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34 [application/octet-stream]
Saving to: 'flag'

flag                                                    100%[=============================================

2025-10-20 04:03:26 (32.0 MB/s) - 'flag' saved [34/34]

tfg11431037-picoctf@webshell:~$ cat flag
picoCTF{s4n1ty_v3r1f13d_f28ac910}
```

10/27

參加

高中跨校資安實務

素養挑戰賽-個人賽

參賽！

-利用因才網參賽的自學資源影片進行學習

-解題

使用的工具： https://www.dcode.fr/caesar-cipher

-參賽證明

ISIP 證-114-高教師-114120019

教育部跨領域資通訊安全人才培育計畫

參賽證明

聶宜帆 同學

於114年10月26日至11月24日參加本計畫「第二屆高中跨校資安實務素養挑戰賽」之個人賽，特此證明。

個人賽任務內容：

1. 自主學習資訊安全基礎概念與CTF (Capture of The Flag) 資安實務競賽解題
2. 自主完成CTF資安實務競賽解題任務，完成度為 100 %。

教育部跨領域資通訊安全人才培育計畫推動辦公室

鄧惟中 Wei-Zhong Deng

許建隆 Chien-Lung Hsu

黃正達 Cheng-Ta Huang

翁麒耀 Qi-yao Weng

教育部 推動辦公室 跨領域資通訊安全人才培育計畫

114 年 12 月 9 日

PDF檔： https://niehmini.github.io/InfoSec/個人賽參賽證明.pdf

11/03

人工智慧AI簡介

-看老師製作的YT影片學習人工智慧大概架構

影片連結： https://www.youtube.com/watch?v=5U2l2LTIY5M&list=PLUFIbeASyXmfqQOctNEfVT3BVyV_3eo-E

-授課老師教學簡報

-實作—訓練模型

請Chat GPT幫忙寫訓練程式

請用MNIST資料集寫一個卷積神經網路CNN的訓練程式

當然可以 👍
以下是一個使用 **TensorFlow / Keras** 建立並訓練 CNN 的完整範例，用於 **MNIST 手寫數字辨識**。

這個程式會：

1. 載入 MNIST 資料集
2. 建立卷積神經網路（CNN）模型
3. 編譯與訓練模型
4. 評估模型準確率

📘 程式碼範例

python　　複製程式碼

```python
# 匯入必要的套件
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical

# 1 載入 MNIST 資料集
```

最好畫出ACCURY正確率跟Epoch關係圖 以及lass和Epoch的關係圖

非常好 👍
我們可以在訓練後使用 `matplotlib` 將 **準確率 (accuracy)** 與 **損失 (loss)** 隨著訓練 epoch 的變化畫出圖表。

以下是完整程式（包含畫圖部分）：

🧠 CNN + 訓練曲線 (MNIST)

python　　複製程式碼

```python
# 匯入必要的套件
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt

# 1 載入 MNIST 資料集
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# 2 資料預處理
x_train = x_train.reshape((x_train.shape[0], 28, 28, 1)).astype('float32') / 255
```

實作結果

```
Epoch 1/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 49s 55ms/step - accuracy: 0.8516 - loss: 0.4688 - val_accuracy: 0.9853 - val_loss: 0.0506
Epoch 2/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 47s 55ms/step - accuracy: 0.9818 - loss: 0.0583 - val_accuracy: 0.9893 - val_loss: 0.0387
Epoch 3/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 45s 54ms/step - accuracy: 0.9886 - loss: 0.0370 - val_accuracy: 0.9887 - val_loss: 0.0398
Epoch 4/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 47s 56ms/step - accuracy: 0.9922 - loss: 0.0279 - val_accuracy: 0.9885 - val_loss: 0.0403
Epoch 5/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 81s 55ms/step - accuracy: 0.9924 - loss: 0.0223 - val_accuracy: 0.9883 - val_loss: 0.0384
Epoch 6/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 82s 55ms/step - accuracy: 0.9941 - loss: 0.0182 - val_accuracy: 0.9895 - val_loss: 0.0356
Epoch 7/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 45s 53ms/step - accuracy: 0.9954 - loss: 0.0152 - val_accuracy: 0.9897 - val_loss: 0.0367
Epoch 8/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 46s 55ms/step - accuracy: 0.9962 - loss: 0.0112 - val_accuracy: 0.9920 - val_loss: 0.0304
Epoch 9/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 82s 54ms/step - accuracy: 0.9974 - loss: 0.0092 - val_accuracy: 0.9905 - val_loss: 0.0421
Epoch 10/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 81s 53ms/step - accuracy: 0.9969 - loss: 0.0092 - val_accuracy: 0.9903 - val_loss: 0.0460
```

正確率隨訓練次數增加　　失誤率隨訓練次數減少

```python
# 匯入必要的套件
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt

# 1 載入 MNIST 資料集
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# 2 資料預處理
x_train = x_train.reshape((x_train.shape[0], 28, 28, 1)).astype('float32') / 255
x_test = x_test.reshape((x_test.shape[0], 28, 28, 1)).astype('float32') / 255

y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)

# 3 建立 CNN 模型
model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.Flatten(),
        layers.Dense(64, activation='relu'),
        layers.Dense(10, activation='softmax')
])

# 顯示模型架構
model.summary()

# 4 編譯模型
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# 5 訓練模型
history = model.fit(x_train, y_train, epochs=10, batch_size=64,
                    validation_split=0.1, verbose=1)

# 6 評估模型
test_loss, test_acc = model.evaluate(x_test, y_test, verbose=0)
print(f"\n✅ 測試集準確率: {test_acc * 100:.2f}%")

# 7 繪製訓練曲線
plt.figure(figsize=(12, 5))

# 左圖：Accuracy vs Epoch
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='訓練準確率', marker='o')
plt.plot(history.history['val_accuracy'], label='驗證準確率', marker='s')
plt.title('Accuracy vs Epoch')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)

# 右圖：Loss vs Epoch
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='訓練損失', marker='o')
plt.plot(history.history['val_loss'], label='驗證損失', marker='s')
plt.title('Loss vs Epoch')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()
```

Colab 程式網址：

https://colab.research.google.com/drive/1HX9CJC9GJBD1IJ_yCiCFB8hc4X7slAKK#scrollTo=NvxUTM6LwN5R

11/10

參加

高中跨校資安實務

素養挑戰賽-團體賽

Challenge

粗心的程式設計師

20

描述：粗心的程式設計師，常常會不小心在程式中遺留下重要資料。

提示：請觀察index2.html網頁。

Key格式：CTF{/}

index2.html

Flag

Submit

```
            padding: 10px 15px;
            cursor: pointer;
            border-radius: 3px;
        }
        input[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>

<form action="/submit" method="post">
    <h2>登入</h2>
    <label for="username">帳號:</label>
    <input type="text" id="username" name="username" required>

    <label for="password">密碼:</label>
    <input type="password" id="password" name="password" required>

    <input type="submit" value="送出">

            <! --   admin/V@1ue230 -->
</form>
     <p hidden> --   admin/V@1ue230 </p>
<script defer src="https://static.cloudflareinsights.com/beacon.min.js/vcd15cbe7772f49c399c6a5babf22c124171
</body>
```

答案!

Challenge

魚目混珠

20

資訊人員Crystal發現公司文件中的文件被駭客篡改。請您幫忙找出那些文件是被篡改過？

Key 格式： CTF{ }

WinMD5Free應用程式 線索：WinMD5Free

CKD-202510...

Flag

Submit

代到文件16的時候發現不一樣

打開看看，發現...

Challenge

世外桃源

20

描述： 請問原始檔案的第一行文字為何?

提示：

1. 眼見不為憑
2. 線上工具：hexed.it

Key格式：CTF{}

DB_.pptx

Flag

Submit

打開題目給的連結、放入題目給的檔案

解出來"7z"→壓縮檔

題目給的網站資源:https: //hexed.it/

解壓縮

答案!

DB_目錄 - 記事本

檔案 編輯 檢視

資料庫的核心理論與實務7／e

https://www.books.com.tw/products/0010779579

第 一 篇 概論
第 1 章 資料庫論述

第 二 篇 設計與使用篇
第 2 章 實體關係模式－基本概念
第 3 章 實體關係模式－進階練習
第 4 章 關聯模式
第 5 章 關聯模式的運算
第 6 章 基本的SQL
第 7 章 進階的SQL
第 8 章 正規化

第 三 篇 系統篇
第 9 章 資料儲存結構
第 10 章 基本的查詢處理與最佳化
第 11 章 進階的查詢處理與最佳化
第 12 章 交易管理

第 四 篇 進階篇
第 13 章 擴充實體關係模式
第 14 章 XML與資料庫
第 15 章 雲端運算與資料庫

第 五 篇 實務篇(置於本書書籍服務平台)
第 16 章 常見的資料庫管理系統
第 17 章 Web程式語言和資料庫應用
第 18 章 資料倉儲

第 2 行，第 1 欄 100% Windows (CRLF) UTF-8

Challenge

馬雅數字

20

請運用馬雅數字建立Key，再用Vigenère傳統式加密法解密文字。

提示

1. 先請開啟英文字母順序表網頁 abc.html
2. 參考附圖馬雅數字對應到英文字母找出解密鑰匙，馬雅數字為20進位
3. 加密鑰匙英文字母為由左至右。對應到馬雅數字為由上至下，字母均為小寫。
4. 運用Vigenère傳統式加密法線上工具。
5. 使用解密鑰匙將密文 cmzvwiuixadlhvetztxreapmzl 轉換成明文。

線索：vigenere.html

Key格式： CTF{明文}

Mayan_numb...

Flag

Submit

數字	大寫	小寫	音標	數字	大寫	小寫	音標	數字	大寫	小寫	音標
01	A	a	[ɑ]	10	J	j	[ʤɛ]	19	S	s	[ɛs]
02	B	b	[bi:]	11	K	k	[kɛi]	20	T	t	[ti:]
03	C	c	[si:]	12	L	l	[ɛl]	21	U	u	[ju:]
04	D	d	[di:]	13	M	m	[ɛm]	22	V	v	[vi:]
05	E	e	[i:]	14	N	n	[ɛn]	23	W	w	[ˈdʌbəlju:]
06	F	f	[ɛf]	15	O	o	[əu]	24	X	x	[ɛks]
07	G	g	[ʤi:]	16	P	p	[pi:]	25	Y	y	[wɑi]
08	H	h	[ɛitʃ]	17	Q	q	[kju:]	26	Z	z	[zi:] [zɛd]
09	I	i	[ɑi]	18	R	r	[ɑ:]				

對照後的解密鑰匙：temple

Text To Encode

jinglebellshorseopensleigh

答案!

Text To Decode

cmzvwiuixadlhvetztxreapmzl

KEY = TEMPLE

Random Key | Encode | Decode | Remove Spaces | Add Spaces | 4

Message:
- CMZVWIUIXADLHVETZTXREAPMZL

Keys: *TEMPLE*

7	22	14	11	15	22

AS:
- jinglebellshorseopensleigh

題目給的網站資源：https://www.ifreesite.com/abc/abc.html

把題目的檔案丟進去

題目給的網站：https://futureboy.us/stegano/decinput.html

解出來的東西

Q1RGe1RoZSBNYW55IEFkdmVudHVyZXMgb2YgV2lubmllIHRoZSBQb29ofQ==

拿去給Base64解

答案！

網站：https://www.utilities-online.info/base64

全部完成！

心得感想

在這次的資安挑戰賽後，我覺得自己學到了很多。過去相較於團體賽，我更喜歡個人賽。對我來說，憑自己的實力比賽，輸了也能心服口服；但若與他人一組，可能會遇到有人「躺分」或拖慢進度，讓我覺得不太公平。

然而，這次的比賽改變了我的想法。在解題的過程中，我體會到團隊合作的樂趣：題目解不出來時一起苦惱，解出來後互相分享思路、交流技巧，不僅讓彼此進步，也能學習不同的解題方法。即使有組員最後沒有成功解出題目，他們在過程中依然努力思考，只是方向可能與正解不同而已。那份共同努力、彼此鼓勵的過程，讓我深刻感受到團隊精神的重要以及和他人合作的快樂。

另外，這次比賽的分組是由老師隨機安排，我與組員們在合作前都不認識。和陌生人一起解題、討論的過程讓我覺得很新奇有趣。相較於個人賽的單打獨鬥，團體賽讓我體驗到更多元的想法與合作的成就感。

此外，當我成功解出題目，送出答案後看到螢幕上出現“Correct”的那一刻讓我非常感動。老實說，在課堂學習時，我曾覺得資安解題似乎只是把密文丟進網站轉換，感覺自己沒有學到什麼。但經過這次比賽我才發現，工具只是輔助，真正重要的是能否判斷題目的方向與選擇合適的方法。如果一開始抓錯重點、用錯網站，就算嘗試再多次也得不到結果。當我經過多次嘗試、終於成功解出答案時，那種成就感是前所未有的。

這次資安挑戰賽讓我重新認識了團隊合作，也讓我真正體會到「自己動腦解出問題」的樂趣。它不僅是一次比賽，更是一場讓我成長、學會信任夥伴的寶貴經驗。

參賽證明

ISIP 證-114-高教師-114121483

教育部跨領域資通訊安全人才培育計畫

結業證明

聶宜帆 同學

於114年10月26日至11月24日參加本計畫「第二屆高中跨校資安實務素養挑戰賽_團體賽」已於因材網平台上完成本課程包之全部指定內容，包含線上教材瀏覽、主題練習、單元任務、課後評量等學習活動，經系統審核合格，特頒此證。。

課程包內容包含：

1. 核心素養導向教材
2. 多元情境式任務與互動練習
3. 單元重點回顧與強化活動
4. 總結性評量／學習成果驗證

教育部跨領域資通訊安全人才培育計畫推動辦公室

鄧惟中 Wei-Zhong Deng

許建隆 Chien-Lung Hsu

黃正達 Cheng-Ta Huang

翁麒耀 Qi-yao Weng

教育部 推動辦公室 跨領域資通訊安全人才培育計畫

114 年 12 月 9 日

PDF檔： https://niehmini.github.io/InfoSec/團體賽結業證明.pdf

11/17

模型逆向攻擊

MNIST訓練模型

-利用Chat GPT寫出程式碼於google colab執行

請幫我做模型逆向攻擊MNIST訓練模型的colab程式碼

ChatGPT 上午10:28

下面給你 可直接貼進 Google Colab 執行 的完整程式碼範例，示範：

✅ 在 MNIST 訓練好的模型上

- 建立模型 (簡單 CNN)
- 進行模型逆向攻擊 (Model Inversion Attack)：
  透過梯度上升，從特定標籤反推出讓模型信心最高的「代表性影像」。

🧪 Colab：模型逆向攻擊 (Model Inversion) MNIST 範例

你只要整段貼到 Colab 執行即可。

python 複製程式碼

```python
# ==========================================
# 1. 安裝/匯入套件
# ==========================================
import torch
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F
from torchvision import datasets, transforms
import matplotlib.pyplot as plt

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print("Using device:", device)

# ==========================================
# 2. 建立簡單 CNN 模型
# ==========================================
```

+ 傳送訊息給 @ChatGPT

ChatGPT 可能會出錯。請查證重要資訊。新組對話給不會使用你的個人 ChatGPT 記憶。

▷ 與Chat gpt的部分對話紀錄

◁ 部分程式碼內容

```python
train_dataset = datasets.MNIST(root='./data', train=True, download=True, transform=transform)
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=64, shuffle=True)

# ==========================================
# 4. 訓練模型
# ==========================================
optimizer = optim.Adam(model.parameters(), lr=0.001)
criterion = nn.CrossEntropyLoss()

epochs = 2  # 為了快速示範，只跑 2 epoch

print("Training...")
for epoch in range(epochs):
    for data, target in train_loader:
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        output = model(data)
        loss = criterion(output, target)
        loss.backward()
        optimizer.step()
    print(f"Epoch {epoch+1}/{epochs} Done")


# ==========================================
# 5. 模型逆向攻擊 (Model Inversion)
#    針對特定標籤，反推出最佳代表影像
# ==========================================
def model_inversion(model, target_label, steps=300, lr=0.1):
    model.eval()

    # 從隨機噪音開始
    img = torch.randn((1, 1, 28, 28), device=device, requires_grad=True)

    optimizer = optim.Adam([img], lr=lr)

    target_tensor = torch.tensor([target_label], device=device)

    for i in range(steps):
        optimizer.zero_grad()
```

```
Using device: cpu
100%|██████████| 9.91M/9.91M [00:00<00:00, 38.2MB/s]
100%|██████████| 28.9k/28.9k [00:00<00:00, 1.01MB/s]
100%|██████████| 1.65M/1.65M [00:00<00:00, 10.0MB/s]
100%|██████████| 4.54k/4.54k [00:00<00:00, 6.89MB/s]
Training...
Epoch 1/2 Done
Epoch 2/2 Done
Running model inversion for label 7 ...
```

Model inversion result for label 7

▷ 執行結果

colab完整程式碼連結： https://colab.research.google.com/drive/1xH2Fq9WWYErWSDK3Ffgd_o0BSFzV_slx?usp=sharing

11/24

SQL Injection

學習與實作

使用教學簡報學習

-SQL Injection 簡介

什麼是 SQL Injection？

- 一種駭客攻擊手法，藉由輸入惡意 SQL 指令竄改資料庫查詢語句。
- 可能導致：
  - 登入繞過驗證
  - 瀏覽/竄改/刪除資料
  - 控制整個資料庫

◁老師上課簡報

▽簡單語法舉例與能運用在生活中保護自己的個資的部分

簡單例子（登入繞過）

- 假設有登入語法如下：

```
SELECT * FROM users WHERE username='$user'
AND password='$pass';
```

- 使用者輸入：

username: ' OR '1'='1';/*

password: */ --　（可輸入任意或不填）

如何防範？

- 永遠不要直接把使用者輸入放進 SQL
- 例如：使用 參數化查詢（Prepared Statements）

```
$stmt = $pdo->prepare("SELECT * FROM users
WHERE username = ? AND password = ?");
```

先寫好一段查詢資料庫的SQL語法草稿，但先空下兩個位置（用？表示），等等再把真正的帳號密碼填進去。

```
$stmt->execute([$user, $pass]);
```

把使用者輸入的帳號 $user 和密碼 $pass 填到剛預留的兩個？裡。

SQL Inject安全版的登入系統實作

-使用XAMPP進行SQL Injection的實作

▽ 在MySQL中建立資料表

login.php - 記事本

檔案　編輯　檢視

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Login Page</title>
</head>
<body>
    <h2>登入系統</h2>
    <form action="login.php" method="POST">
        <label>帳號：</label>
        <input type="text" name="username" required><br><br>

        <label>密碼：</label>
        <input type="password" name="password" required><br><br>

        <input type="submit" value="登入">
    </form>
</body>
</html>

<?php
$host = "localhost";
$user = "root";     // 如果你有設定密碼請修改
$pass = "";         // XAMPP 預設空白
$dbname = "login_test";

// 連線資料庫
$conn = new mysqli($host, $user, $pass, $dbname);
if ($conn->connect_error) {
    die("資料庫連線失敗：" . $conn->connect_error);
}

// 接收表單資料
$username = $_POST['username'] ?? '';
$password = $_POST['password'] ?? '';

// 使用 Prepared Statement 防止 SQL injection
$stmt = $conn->prepare("SELECT * FROM users WHERE username = ? AND password = ?");
$stmt->bind_param("ss", $username, $password);
$stmt->execute();

$result = $stmt->get_result();

// 判斷帳密
if ($result->num_rows > 0) {
    echo "登入成功！歡迎你，" . htmlspecialchars($username);
} else {
    echo "帳號或密碼錯誤！";
}

$stmt->close();
$conn->close();
?>
```

◁ 請 Chat gpt 幫忙生成的程式碼

→ 存在記事本裡

儲存成php檔

另存新檔

本機 › Data (D:) › xampp › htdocs › login

搜尋 login

組合管理　新增資料夾

名稱	修改日期	類型	大小
login.php	2025/11/24 上午 11:35	PHP 檔案	2 KB

檔案名稱(N): login.php

存檔類型(T): 所有檔案 (*.*)

隱藏資料夾　編碼方式: UTF-8　存檔(S)　取消

在瀏覽器網址列上打上 localhost/login/login.php
成功出現登入表單

依照老師的教學簡報
在瀏覽器中打上網址

在顯示出的網站介面中輸入設定的帳號密碼

登入系統

帳號： test

密碼： ••••

登入

帳號或密碼錯誤！

登入系統

帳號：

密碼：

登入

登入成功！歡迎你，test

成功了！！

SQL Inject不安全版的登入系統實作

```php
<?php
$host = "localhost";
$user = "root"; // XAMPP/WAMP 預設帳號
$pass = "";
$dbname = "login_test";

// 連線資料庫
$conn = new mysqli($host, $user, $pass, $dbname);
if ($conn->connect_error) {
    die("資料庫連線失敗：" . $conn->connect_error);
}

// 接收前端表單輸入
$username = $_POST['username'] ?? '';
$password = $_POST['password'] ?? '';

// ❌❌ 脆弱、不安全的 SQL 拼接（為了教學實驗）
$sql = "SELECT * FROM users WHERE username = '$username' AND password = '$password'";

// 顯示 SQL（方便你觀察注入效果）
echo "<p><b>執行的 SQL：</b> $sql</p>";

$result = $conn->query($sql);

// 驗證結果
if ($result && $result->num_rows > 0) {
    echo "<h2 style='color: green;'>登入成功！（此頁面可被 SQL Injection 攻破）</h2>";
} else {
    echo "<h2 style='color: red;'>登入失敗（輸入正確帳密或試著做 SQL Injection）</h2>";
}

$conn->close();
?>


<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Login Page</title>
</head>
<body>
    <h2>登入系統</h2>
    <form action="login.php" method="POST">
        <label>帳號：</label>
        <input type="text" name="username" required><br><br>

        <label>密碼：</label>
        <input type="password" name="password" required><br><br>

        <input type="submit" value="登入">
    </form>
</body>
</html>
```

漏洞在這！

◁ SQL有漏洞的程式碼

登入失敗（輸入正確帳密或試著做 SQL Injection）

登入系統

帳號：' OR '1'='1

密碼：••••••••••

登入

輸入漏洞的帳號密碼也會登入成功

執行的 SQL： SELECT * FROM users WHERE username = 'test' AND password = 'abcd'

登入成功！（此頁面可被 SQL Injection 攻破）

登入系統

帳號：

密碼：

登入

▽ 與正常的程式碼對照

帳號或密碼錯誤

登入系統

帳號：' OR '1'='1

密碼：••••••••••

登入

使用正常的程式碼就不能登入

帳號或密碼錯誤

登入系統

帳號：

密碼：

登入

12/01

封包攔截--

Burp Suite攔截

利用Burp Suite監控所輸入的密碼

-試著監控看看上禮拜做的網站

輸入之後

剛剛輸入的帳號密碼!

-試試二代校務行政系統

輸入我的帳號密碼

我的帳號


```
{
  "account":"fg11431037",
  "password":"[redacted]",
  "relayState":null,
  "samlRequest":null
}
```


OMG居然成功了?!!

真的可以登入...

-再試一次

```
21 {
   "account":"test",
   "password":"test",
   "relayState":null,
   "samlRequest":null
}
```


真的看的到...

心得:

國中時的輔導課總會請講師來教導我們資安的重要，例如不要亂連不明來源的網站，密碼很容易就此被盜走之類的，那時候我總覺得這些狀況離現實很遠，系統怎麼可能那麼容易被入侵？

但在今天的課程過後，我才真正體會到網路世界到底有多危險。如果一不小心讓這類程式連接到我的電腦的話，駭客想要看到密碼，只需動動手指就行。這讓我意識到，資安並不是誇張的劇情，而是每個人都可能遇到的真實問題。

今天的課讓我體會到資安世界的有趣與危險，也更明白平時提高警覺、養成良好習慣的重要性。

-換個網站試試

◁巴哈姆特動畫瘋網站的登入介面

也看的到，只不過只看的到使用者名稱看不到密碼

說明：

這次選擇嘗試的是我日常會使用的網站。在照片中我們可以看到輸入捨用者名稱以後，Burp Suite中就會跳出所輸入的內容。(如果輸入了密碼的話Burp Suite就會跳掉，所以我沒有截到圖。但我覺得就算只有暴露帳號，還是蠻危險的)

12/15

資安推廣影片與期末報告海報討論

拍攝資安推廣影片

1. 決定主題

2. 請Chat gpt幫忙生成腳本

3. 修改腳本成最佳版本

4. 拍片

PHOTO

課後感想

起初，我帶著一點憧憬、一點想像，跨進這門課的大門。那時候的我只覺得「資安」這兩個字背後有種神祕的魅力--似乎藏著某種只有少數人懂得的語言。沒想到，真正踏上這條路後，我看到的，是一個比想像中更廣、更深、更有趣的世界。

從最一開始的「編碼與解碼」、「密碼學概論」開始，那些看似亂碼，又似某種秘密對話的開端，彷彿是資訊世界的另一種魔法。

在「隱寫術」的那堂課，是我第一次意識到訊息可以被藏得如此安靜、如此優雅，默默存在於細節深處。

「Linux」中課程，不同的指令與介面讓我感到陌生，但也因此更想探索。

兩次跨校資安實務素養體驗賽，是讓我對資安這個領域真正燃起興趣的時刻。個人賽的緊張、團體賽的默契與感動，讓我了解到原來學習不只是在吸收，而也在等待被挑戰、被驗證的瞬間中，獲得另一種成就。

在「人工智慧簡介」中，我們看見技術與未來的交點；「SQL Injection」、「Burp Suite攔截」課程，則讓我第一次真正理解，原來資訊世界的脆弱與堅強，一線之隔。

在這學期的學習過程中，我遇過困惑、卡關，也遇過突然恍然大悟的時刻。這些過程就像一種溫柔的磨礪：跌倒使人學會重新站起，好奇使人前進，這門課讓我看到資訊世界的另一面，也使我開始思考，原來那些被我們視為理所當然的安全，都是一套套演算法與邏輯默默構築出來的。

也許我一開始是因為那份神祕與帥氣而靠近，但結束這學期時，吸引我留下的，是看見日常背後的祕密的感動，也讓我在資訊的世界裡，看見了比想像中更美麗的風景。