財團法人電信技術中心出國報告

美國拉斯維加斯受訓 SANS

單位名稱：資通安全組
姓名職稱：蔡銘峯、陳坤裕
派赴國家：美國
出國期間：107/9/22-1079/30
報告日期：107/10/30

摘要

為提升財團法人電信技術中心資安技術能量，本次參加訓練課程 SANS Network Security 2018 FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques 與 SEC573: Automating Information Security with Python，學習惡意程式逆向工程分析技術與 如何使用 python 撰寫高效率之自動化資安腳本。

FOR610 課程內容分五大部分：

1. 惡意程式分析基礎知識介紹：學習如何建立分析環境及基礎靜動態分析技巧。
2. 分析惡意程式碼：以反組譯工具 IDA-Pro 為主，學習分析反組譯之組合語言。
3. 惡意網頁與檔案：學習如何從惡意網頁或檔案中，擷取出惡意程式。
4. 進階惡意程式分析：學習惡意程式加解殼技術。
5. 探討具自我防禦機制之惡意程式：介紹惡意程式防禦機制，及破解技術。

上述五點內容，將分析惡意程式各層面知識由淺入深介紹。實際效益上，可幫助電信技術中心建立以 windows 平台惡意程式為目標之靜動態分析環境與分析流程。未來，也可以此基礎，建立行動裝置惡意程式之分析環境。

SEC573 課程內容分五大部分：

1. Python 簡介與語法介紹 : 學習使用 Python 提供工具，快速撰寫腳本。
2. 防禦 : 學習使用正則表示式，有效率地擷取資料。
3. 取證 : 使用 Python 或第三方提供之套件，針對各類型檔案取出感興趣的資料。
4. 攻擊 : 學習使用 socket 向伺服器要求資料，並使用 socket 建立後門。
5. 搶旗競賽 : 運用前五天課程所習得之知識，解決實際問題。

針對上述五點 SEC573 課程內容，對於如何使用 Python 撰寫高效率的資安自動化腳本皆有詳盡的描述與實際操作操作。在成本效益上，可以提升資安組在各項計畫執行效率，進而降低所需花費之時間成本。

目錄

圖目錄

表目錄

壹、目的

SANS Institute (以下簡稱 SANS) 是國際知名資訊安全教育訓練組織，成立於 1989 年，成立至今已培訓超過 165,000 名資訊安全領域專家，該組織對於各種資安議題、分析與防護技術皆有深入的研究，諸如資安稽核、網路安全管理、惡意程式分析、安全程式設計等，也經常與各大企業資訊安全領域之專家合作，並交流資訊安全實務知識與技術。因此，其所提供之訓練課程與教材，除學術理論亦整合實務經驗，更由深具資訊安全實務經驗之專家擔任講師。除提供資訊安全教育訓練課程外，SANS 也有嚴謹認證機制，提供每一位參與教育訓練學員進行資訊安全能力認證。因此，SANS 為公認之最具資訊安全教育資源與公信力之資安培訓組織。

此次 SANS Network Security 2018 於美國拉斯維加斯舉辦，時間為美西時間 9 月 23 日至 9 月 30 日，主要針對四大主題 (滲透測試、資安稽核、資訊安全管理與電腦鑑識) 開設超過 25 門課程，提供來自世界各地資訊安全從業人員，或有興趣者參與。

電信技術中心此次派員參與 2 門課程，分別為 FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques 與 SEC573: Automating Information Security with Python。FOR610: Reverse-Engineering Malware: Malware Analysis Techniques 課程共 6 天，課程內容為惡意程式分析基礎知識介紹、分析惡意程式碼、惡意網頁與檔案、進階惡意程式分析與探討具自我防禦機制之惡意程式，共 5 大部分。SEC573:Automating Information Security with Python 亦是 6 天課程，課程內容可分成 Python 簡介與語法介紹、防禦、取證、攻擊以及搶旗競賽共 5 項，在搶旗競賽中，講師規劃一系列題目，期望學員能透過競賽方式，充分發揮 5 天所學之知識與技術，已增加開發滲透測試工具經驗。

此次透過參與 SANS 教育訓練課程，使電信技術中心能夠吸取國外最新惡意程式分析相關技術，以提升電信技術中心惡意程式分析能力與效率，並且藉由參與 SEC573 課程提升電信技術中心資安測試效率及開法測試工具的能力。

貳、過程

一、議程

本次 SANS Network Security 2018 共開設超過 25 門教育訓練課程，其中主要分為四大主題，分別為滲透測試、資安稽核、資訊安全管理、電腦鑑識，表 1 為 SANS Network Security 2018 訓練課程總表。

表 1　SANS Network Security 2018 訓練課程總表

課程編號	課程名稱
SEC301	Introduction to Cyber Security
SEC401	Security Essentials Bootcamp Style
SEC440	Critical Security Controls: Planning, Implementing, and Auditing
SEC455	SIEM Design & Implementation
SEC460	Enterprise Threat and Vulnerability Assessment
SEC487	Open-Source Intelligence Gathering (OSINT) and Analysis
SEC501	Advanced Security Essentials – Enterprise Defender
SEC503	Intrusion Detection In-Depth
SEC504	Hacker Tools, Techniques, Exploits, and Incident Handling
SEC505	Securing Windows and PowerShell automation
SEC506	Securing Linux/Unix
SEC511	Continuous Monitoring and Security Operations
SEC524	Cloud Security Fundamentals
SEC542	Web App Penetration Testing and Ethical Hacking
SEC545	Cloud Security Architecture and Operations
SEC546	IPv6 Essentials
SEC550	Active Defense, Offensive Countermeasures and Cyber Deception
SEC555	SIEM with Tactical Analytics
SEC560	Network Penetration Testing and Ethical Hacking
SEC562	CyberCity Hands-on Kinetic Cyber Range Exercise

SEC564	Red Team Operations and Threat Emulation
SEC566	Implementing and Auditing the Critical Security Controls – In-Depth
SEC567	Social Engineering for Penetration Testers
SEC573	Automating Information Security with Python
SEC579	Virtualization and Software-Defined Security
SEC580	Metasploit Kung Fu for Enterprise Pen Testing
SEC599	Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses
SEC617	Wireless Penetration Testing and Ethical Hacking
SEC642	Advanced Web App Penetration Testing, Ethical Hacking, and Exploitation Techniques
SEC660	Advanced Penetration Testing, Exploit Writing, and Ethical Hacking
SEC760	Advanced Exploit Development for Penetration Testers
FOR500	Windows Forensic Analysis
FOR508	Advanced Digital Forensics, Incident Response, and Threat Hunting
FOR518	Mac and iOS Forensic Analysis and Incident Response
FOR526	Memory Forensics In-Depth
FOR572	Advanced Network Forensics: Threat Hunting, Analysis, and Incident Response
FOR578	Cyber Threat Intelligence
FOR585	Advanced Smartphone Forensics
FOR610	Reverse-Engineering Malware: Malware Analysis Tools and Techniques
MGT305	Technical Communication and Presentation Skills for Security Professionals
MGT414	SANS Training Program for CISSP Certification
MGT415	A Practical Introduction to Cyber Security Risk Management

MGT433	SANS Security Awareness: How to Build, Maintain, and Measure a Mature Awareness Program
MGT512	SANS Security Leadership Essentials For Managers with Knowledge Compression
MGT514	Security Strategic Planning, Policy, and Leadership
MGT516	Managing Security Vulnerabilities: Enterprise and Cloud
MGT517	Managing Security Operations: Detection, Response, and Intelligence
MGT525	IT Project Management, Effective Communication, and PMP exam p Prep
MGT535	Incident Response Team Management
DEV522	Defending Web Application Security Essentials
DEV531	Defending Mobile Applications Security Essentials
DEV534	Secure DevOps: A Practical Introduction
DEV540	Secure DevOps and Cloud Application Security
DEV541	Secure Coding in Java/JEE: Developing Defensible Applications
DEV543	Secure Coding in C & C++
DEV544	Secure Coding in .NET: Developing Defensible Applications
AUD507	Auditing & Monitoring Networks, Perimeters & Systems
AUD566	Implementing and Auditing the Critical Security Controls – In-Depth
LEG523	Law of Data Security and Investigations
ICS410	ICS/SCADA Security Essentials
ICS456	Essentials for NERC Critical Infrastructure Protection
ICS515	ICS Active Defense and Incident Response

資料來源：SANS 課程網站 [1]

目前組內正研發 APK 惡意程式自動化分析引擎。分析過程中，需要經常利用不同工具，透過靜動態分析方法了解惡意程式之行為。此外，也隨著惡意程式不斷演化，攻擊手法推陳出新，資安組惡意程式分析技術須更進一步強化，故此次藉由參加業務相關之 FOR610: Reverse Engineering Malware: Malware Analysis Tools and Techniques，希望強化分析惡意程式相關議題之理解深度，提升分析能力與效率。

此外，組內目前執行之各項專案，如滲透測試、惡意程式自動化分析引擎……等，皆需要撰寫自動化腳本實現功能。Python 是一種高階腳本語言， 由 Guido van Rossum 創造，自 1991 年發展至今，目前同時存在 Python 2.7.15 與 Python 3.7.1 兩種版本，由於具備簡潔清晰之語法且擁有大量威力強大的套件，Python 長期備受資安人員青睞，此次藉由參加業務相關之 SEC573:Automating Information Security with Python，希望能習得使用 Python 撰寫高效率自動化資安腳本之技巧，提升專案執行效率。

二、工作內容

(一) FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques

本課程為期 6 天，授課講師為 Lenny Zeltser，其專長為惡意程式分析。FOR610 將課程主題分為惡意程式分析基礎知識介紹、分析惡意程式碼、惡意網頁與檔案、進階惡意程式分析與探討具自我防禦機制之惡意程式，共五大主題。最後一天，講師設計一系列題目，主要期望學員透過解答題目，充分運用且融會貫通前 5 日所學之技術，以增加學員在惡意程式分析之經驗。課程完整議程詳見下表 2。

表 2 FOR610 訓練課程議程

日期	課程內容
9/23 09:00 ~ 17:00	• Assembling a toolkit for effective malware analysis • Examining static properties of suspicious programs

Malware Analysis Fundamentals	• Performing behavioral analysis of malicious Windows executables • Performing static and dynamic code analysis of malicious Windows executables • Interacting with malware in a lab to derive additional behavioral characteristics
9/24 09:00 ~ 17:00 Reversing Malicious Code	• Understanding core x86 assembly concepts to perform malicious code analysis • Identifying key assembly logic structures with a disassembler • Following program control flow to understand decision points during execution • Recognizing common malware characteristics at the Windows API level (registry manipulation, keylogging, HTTP communications, droppers) • Extending assembly knowledge to include x64 code analysis
9/25 09:00 ~ 17:00 Malicious Web and Document Files	• Interacting with malicious websites to assess the nature of their threats • De-obfuscating malicious JavaScript using debuggers and interpreters • Analyzing suspicious PDF files • Examining malicious Microsoft Office documents, including files with macros • Analyzing malicious RTF document files
9/26	• Recognizing packed malware

09:00 ~ 17:00 In-Depth Malware Analysis	• Getting started with unpacking • Using debuggers for dumping packed malware from memory • Analyzing multi-technology and file-less malware • Code injection and API hooking • Using memory forensics for malware analysis
9/27 09:00 ~ 17:00 Examining Self-Defending Malware	• How malware detects debuggers and protects embedded data • Unpacking malicious software that employs process hollowing • Bypassing the attempts by malware to detect and evade the analysis toolkit • Handing code misdirection techniques, including SHE and TLS Callbacks • Unpacking malicious executable by anticipating the packer's actions
9/28 09:00 ~ 17:00 Malware Analysis Tournament	• Behavioral malware analysis • Dynamic malware analysis (using a debugger) • Static malware analysis (using a disassembler) • JavaScript de-obfuscation • PDF document analysis • Office document analysis • Memory analysis

資料來源：SANS Network Security 2018 FOR610 課程網頁[2]

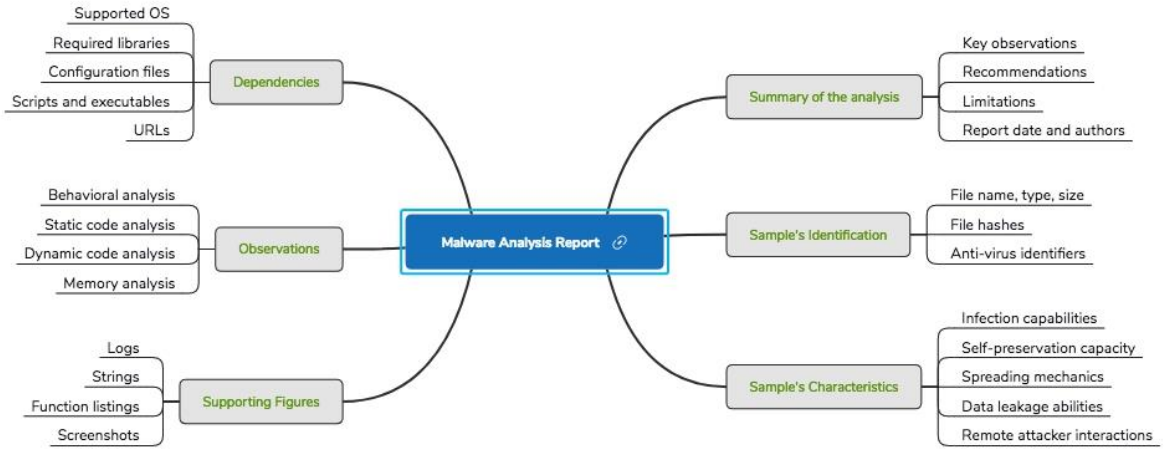
1. Malware Analysis Fundamentals

第一天課程共分為六大部分，分別是 Introduction to Malware Analysis, Malware Analysis Lab, Static Properties Analysis, Behavioral Analysis Essentials, Code Analysis Essentials 以及 Interactive Behavioral Analysis。

在 Introduction to Malware Analysis 中，講師定義何為 reverse-engineering 與 malware analysis 之差別，前者專注在分析惡意程式碼，而後者專注在檢視惡意程式之基本資訊。而在此課程中，我們專注在 windows 樣本上。

惡意程式分析分為四個階段，由簡而難依序為 fully automated analysis, static properties analysis, interactive behavioral analysis 與 manual code reversing。此四階段在後續課程中皆會完整介紹。

惡意程式分析完畢後，便需撰寫分析報告。對分析人員而言，通常需要有 verbal reports, suspicious files, file system image, memory image, network logs, anomaly observations 才有辦法進行分析。分析完畢後，則可回答下列問題 1. What malware does? 2. How to identify it? 3. Attacker's profile. 4. IR recommendations. 5. Reports and IOCs. 6. Malware trends. 若以心智圖來看，分析報告所需涵蓋內容由圖 1 所示。



資料來源：本報告自行整理

圖 1 分析報告所需涵蓋內容

此外，分析時善用免費分析資源亦相當重要。這些資源包含：Malware data repositories: VirusTotal, #totalhash; Multi-engine antivirus scanners: VirusTotal, metadefender, VirSCAN, AVCaesar; File reputation: Malware Hash Registry, HashSets; Automated sandboxed: Malwr, Hybrid Analysis; Website investigation: urlQuery, vURL, Quttera; Other threat intelligence: PassiveTotal, Censys, Open Threat Exchange 等皆可協助我們進行初步判斷，或取得公開情資，以利進階分析。然而，自動化分析與公開情資雖好用，但不可全然信之，必須自主分析驗證後才可寫入報告。

因此，我們必須建立分析環境，以利進階驗證初級情資。分析環境必須嚴格控管，並與外部網路隔絕。課程中，講師建議使用虛擬化技術，虛擬機器來建立分析環境。因為虛擬化環境，可高度客製化，並可快速還原分析環境。

在此課堂中，由於分析標的為 windows 惡意程式，因此我們分析環境需要兩種不同作業系統之虛擬機器。1. Windows 10. 2. Ubuntu 。在 Windows 10 作業系統上，我們會安裝 PeStudio, strings, CFF Explorer, peframe, Detect It Easy, HxD 等工具以利 static properties analysis。安裝 Process hacker, Process monitor, Regshot, wireshark, fakedns, TcpLogView 等工具以利 Behavioral analysis。我們也會安裝 IDA pro, x64dbg/x32dbg, OllyDumpEx, jmp2it, Scylla 等工具以利 Code analysis。

在 static properties analysis 中，我們以惡意程式樣本 brbbot.exe 來實戰。分析時，我們會關注之 static properties 有 file and section hashes, packer identification, embedded resources, imports/exports, crypto references, digital certificates 及 interesting strings。上述資訊，皆可協助分析人員在分析初始階

段建立分析 roadmap。而當 roadmap 建立後，分析人員可開始進行驗證或進階探索工作。

從字串分析中，我們或可拼湊出幾項有用分析資訊。1. IOCs for detection. 2. Persistence mechanism 3. Communication mechanism。在 Linux 環境下，我們可使用 工具 pestr 查找有趣字串。而在 Windows 環境下，我們則可使用 BinText 或 Strings2 來完成相同任務。

針對其他 static properties 分析，在 Windows 下，也可使用 PeStudio 來完成。PeStudio 可協助查看 file's section, imports, 及 VirusTotal 檢查結果。在 Linux 環境下，則可用 perframe 及 pescanner.py 來完成相同任務。另外，也可使用 Detect It Ea o PE 來檢查 PE header details 及加殼工具。sy (DIE) 或 ExeInf

另外，尚有開源工具可參考使用。1. Signsrch, 2. Pescan/ portex 3. MASTIFF 4. Exiftool 5. Trid 6. Viper。

取得 static properties 後，我們便可建構出分析 roadmap 以及做出部分猜測。若要驗證猜測是否正確，則需進一步分析。Behavioral analysis 則是其中一種驗證猜測之方法。

在此分析階段，我們可使用 Process hacker, Process Monitor, Regshot, ProcDOT 與 wireshark 進行驗證。在分析前，我們會先以上述工具觀察尚未被感染之作業系統，將之感染後，再觀察並比對有哪些差異。由此當作分析切入點。

而在 Behavioral analysis 後，有時初步的行為分析還是無法回答所有問題。此時，則必須進行 code analysis。Code analysis 階段，我們會使用到

disassembler (IDA pro/radare2) 與 debugger (x64dbg/Windbg) 。而上述之 static properties 與 behavioral analysis 則可協助判斷要分析哪部分的 code。

Code analysis 分析完，有時在驗證過程中，我們會使用 Interactive Behavioral analysis 進行驗證。此方法則是在建立分析環境後，試圖與樣本互動，並觀察樣本反應是否如我們所預測。

綜觀第一天上課之內容，講師運用一隻惡意程式樣本，並搭配課程介紹之所有分析方法進行分析，讓受訓人員充分理解惡意程式分析流程。然而，第一天所介紹之分析手法只能對付相對容易分析之惡意程式。後續課程將針對惡意程式進階分析手法，做更詳細之介紹。

2. Reversing Malicious Code

第二天課程共分為四大部分，分別是 Core Reversing Concepts, Reversing Functions, Control Flow In-Depth 以及 API Patterns in Malware。

在 Core Reversing Concepts 中，介紹了程式碼編譯成二進制檔的 lifecycle。接著介紹分析組合語言之主力軟體 IDA pro。講師分別介紹 IDA pro 中各種基本操作，讓受訓者得以對該軟體有初步理解。接著從 IDA pro 分析中，介紹組合語言之基礎，例如：Instructions 、registers 等。

在 Reversing Functions 中，則介紹了如何鎖定特定 function，追蹤是哪一段程式碼呼叫了該 function，function 之 input (value passed in), body (code to perform the task) 與 return (value passed back)，以及 function 與 stack 之間互動關係。 並以一簡單範例將上述概念完整介紹。

在 Control Flow In-Depth 中，則介紹了程式碼中的控制流程。例如：if else statement, if elseif else statement, loops (for, while), AND, OR 及 Switch statement 等。每一個 control flow 皆有 C 語言範例與組合語言範例，交互參照。

而在 API Patterns in Malware 中，我們探討惡意程式常用之可疑 API。從 IDA pro 中的 Import Table 為分析切入點，尋找可疑之 API。而猜測之可疑 API 則可從惡意程式行為下手。例如：監控使用者輸入之資訊 (GetKeyState, OpenClipboard, GetClipboardData, CloseClipboard, GetWindowTextA 與 GetAsyncKeyState) 等。以此為分析切入點。

3. Malicious Web and Document Files

第三天課程共分為六大部分，分別是 Interacting with malicious sites, deobfuscating scripts using debuggers, deobfuscating scripts using interpreters, malicious pdf document analysis, macros in malicious office documents 及 malicious RTF documents。雖然本課程專注在二進制檔分析上，但有時在分析過程中，我們須與 web 或是其他可能藏有惡意程式的檔案互動，才有辦法取得真正的惡意程式。因此，第三天課程主要專注在惡意程式藏身處，並試圖從中取得真正關鍵之二進制檔。

在 Interacting with malicious sites 中，介紹了如何與惡意網頁互動，取得真正關鍵之二進制檔。首先，介紹了 TOR 與 VPN 工具，讓惡意程式分析師得以隱藏真實網路身份，在分析時不至於曝光。另外，也建議分析人員使用虛擬機器分析，並修改虛擬機器相關設定，隱藏分析環境。在訪問 malicious websites 時，也可使用 wget, curl, pinpoint, scout 或直接使用分析環境中之瀏覽器。

在訪問 malicious websites 之同時，開啟 wireshark 與 fiddler 紀錄網路流量與 HTTP/HTTPS session。觀察 log 後，我們或可從中發現分析環境在背後下載了什麼檔案，或連線到哪些地方。其中，我們亦可使用 CapTipper 與 Network Miner 協助分析網路流量。

由於與 malicious websites 互動同時，使用者可能會不知不覺下載了一些 scripts (powershell 或 javascripts)。而這些 scripts 為了防止被分析，因此都會採用混淆手法，增加分析難度。因此，在 Deobfuscating scripts using debuggers 中，介紹了如何使用 debugger 來「解」經過混淆的 scripts。課程中，講師引導我們使用 windows Internet Explorer 內建 debugger 功能， 讓瀏覽器直接執行 Javascripts，並設定斷點，讓我們得以直接查看經過瀏覽器「解」出來的 Javascripts，以利後續分析。

「解」混淆過之 scripts 也可用 interpreter 的方式來處理。因此，在 Deobfuscating script using interpreters 中，講師介紹了如何使用 SpiderMonkey, CScript 與 V8 interpreter 來處理混淆過之 scripts。

在 Malicious PDF Documents analysis 中，講師介紹了 PDF 結構、駭客如何在 PDF 中藏惡意程式碼，以及如何找到並分析之。其中，我們可以使用 REMnux 中預載之程式 peepdf.py, pdfid.py 與 pdf-parser.py 協助我們快速爬梳 PDF 內容，鎖定可疑部分。

惡意程式亦可藏在微軟 Office 文件的巨集中。因此，在 Macros in Malicious Office Documents 中，講師介紹了何為巨集，駭客如何在巨集中藏惡意程式碼，以及如何找到並分析之。其中，我們可以使用 olevba.py, oledump.py, xor-kpa.py 協助我們快速爬梳 Office Documents 內容，鎖定可疑部分，並分析之。

不僅如此，惡意程式許多時候也會藏身在 Rich Text File (RTF) 格式的文件中，該文件亦可用微軟 office word 開啟。因此，在 Malicious RTF Documents 中，講師介紹了 RTF 格式文件之結構、駭客如何在 RTF 文件中藏惡意程式碼，以及如何找到並分析之。其中，我們可以使用 rtfdump.py, x64dbg/x32dbg, rtfobj.py 協助我們快速爬梳 RTF 文件內容，鎖定可疑部分，並分析之。

4. In-Depth Malware Analysis

第四天課程共分為七大部分，分別是 Recognizing Packed Malware, Getting Started with Unpacking, Using debuggers for dumping, debugging packed malware, analyzing multi-technology malware, code injection and API hooking 及 malware memory forensics。部分商業軟體，為了防止被竄改或商業機密曝光。因此會使用 pack 技術 (加固/加殼) 保護該軟體。駭客為了避免惡意程式被分析，也會使用加固技術。因此，第四天課程主要專注在如何辨識加解固樣本等議題上。

在 Recognizing Packed Malware 中，講師首先介紹了加固技術原理，並介紹了常見加固工具，例如：UPX, Armadillo, FSG, Themida。接著講師介紹如何人工辨識加固樣本。可從以下四點切入：1. The file contains few readable strings. 2. Byte values in the file or some of its sections are too random (high entropy). 3. The file has few imports or recognizable functions. 4. Embedded strings sometimes reveal the name of the packer。另外，也可使用 Bytehist, pescanner.py, Detect It Easy, trid, file, pepack, packerid, pescan, ProtectionID, RDG Packer Detector, CFF Explorer 或 Exeinfo PE 等工具協助辨識加固樣本。

接著，在 Getting Started with Unpacking 中，講師介紹如何解固樣本。解固時，需注意先使用 PE editor, CFF Explorer 或 setdllcharacteristics 工具處理 ASLR

技術，因此技術會忽略 ImageBase，增加分析困難度。另外，講師亦提到另一解固方法，該方法即是直接在分析環境中執行樣本，讓樣本運行後，自行解固運行之 process 到記憶體中，之後再使用 Scylla 從記憶體中將該 process dump 出來。而在 Using Debuggers for dumping 中，講師亦介紹如何使用 x64dbg 進行樣本解固。

若使用 process dump 出來之方法，在分析時程緊湊下可能太耗時。因此，在 Debugging Packed Malware 中，講師介紹如何使用 debugger，讓樣本運作，自行解固，並分析解固結果亦是提升分析效率之方法。

惡意程式攻擊手法日異月新，許多惡意程式更是結合不同手法，達到感染系統之目的。因此，在 Analyzing Multi-Technology Malware 中，講師介紹了惡意程式如何結合 scripting languages 與 windows internals 來感染系統。

在 Code Injection and API Hooking 中，講師介紹了惡意程式如何在被感染系統中，注入惡意程式碼到其他 processes 中。其中，Windows API 例如：CreateToolhelp32Snapshot, EnumProcesses, OpenProcess, VirtualALLocEX, WriteProcessMemory 與 CreateRemoteThread 可能會在惡意程式執行 code injection 時被呼叫。因此，我們可以此為切入點，辨識樣本是否有使用到上述 API ，並使用 IDA pro 開始分析。

而在 Malware Memory Forensics 中，講師介紹了記憶體鑑識技巧，讓分析師得以更了解樣本行為。分析師可使用 WinPMEM, Comae Memory Toolkit (DumpIt), KnTDD 以及 BelkaSoft Live RAM Capturer 捕捉記憶體。並搭配 vol.py 分析 memory image。

5. Debugger Detection and Data Protection

第五天課程共分為四大部分，分別是 Debugger Detection and Data Protection, Unpacking Process Hollowing, Detecting the Analysis Toolkit 以及 Handling Misdirection Techniques。惡意程式為了防止被分析，採用了很多手法，例如：偵測 Debugger 與資料防護、讓解固過程失效、偵測分析工具、誤導分析等。因此，第五天課程主要專注在上述惡意程式防禦手法上。

在 Debugger Detection and Data Protection 中，講師介紹一隻惡意程式，其中包含有 debugger 偵測功能，該惡意程式使用了 API IsDebuggerPresent 偵測是否正被 debugger 分析。接著講師介紹如何 Patch 該 binary 檔，使得惡意程式防禦機制失效，以利後續分析。此外，駭客亦可使用其他 API 例如：OutputDebugString, CheckRemoteDebuggerPresent, NtQueryInformationProcess 及 ZwQueryInformationProcess。來偵測是否正被 debugger 分析。另一有趣偵測手法，駭客使用執行時間來偵測是否被 debugger 分析。因被 debugger 分析，時間可能會拖長。因此，若該段程式碼執行速度很快，而執行時間過長，則表示該惡意程式可能正被 debugger 分析中。此外，分析人員亦可使用 x64dbg/x32dbg 中的 ScyllaHide plugin，協助隱藏 debugger 特徵，以利後續分析。

駭客亦可能會使用簡易加密手法，將 IP 或其他敏感資訊加密，增加分析困難度。最常見之加密方法為 XOR 加密。分析人員可使用 XORSearch, brxor.py 或 bbcrack.py 來協助尋找疑似經過 XOR 加密之字串，並解密之。此方法不見得每次奏效，但值得一試。此外，駭客亦可能使用 stack string 方式增加分析複雜度。此時，可使用 strdeob.pl 自動 decode stack string。其他工具例如：floss。亦是處理加密字串的好幫手。

在 Unpacking Process Hollowing 中，駭客躲避惡意程式被分析時，可分為四個階段。1. 創造一看似無害之 Process。2. 掏空該 Process 所分配到之記憶體內容。3. 注入惡意 Process 到該記憶體位置。4. 執行該 Process。講師介紹如何使用 x64dbg/x32dbg，在惡意 Process 被 unpack 且注入記憶體位置後，將該段記憶體 dump 出來，並分析之。

在 Detecting the Analysis Toolkit 中，惡意程式為避免被分析，也會利用一些軟硬體特徵來辨識。例如：VMware 中之預設 MAC address、video controller 名稱、特定 ip 位置、或從 file path 中尋找 VMware 關鍵字等。或是硬體狀態，例如：辨識滑鼠動作，若只有點擊，但沒有鬆開的動作，則表示被分析的機率高。因此，建立分析環境時或撰寫自動化分析腳本時，我們皆需修改環境變數，避免被惡意程式偵測出，進而被無法觸發惡意行為。

在 Handling Misdirection Techniques 中，介紹駭客如何使用 SEH (Structured Exception Handling) 方法複雜化程式運作流程，擾亂分析人員。講師也介紹如何在 x64dbg/x32dbg 中分析使用該手法之惡意程式。

(二) SEC573: Automating Information Security with Python

SEC573: Automating Information Security with Python(以下簡稱 SEC573)主要目的為「如何在沒有任何資安分析工具能運用之情況下，使用 Python 撰寫高效率滲透測試腳本」，課程講師 Mark Baggett 擅長滲透測試，擁有 28 年以上資安業界相關經驗，SEC573 課程主題可分成 Python 簡介與語法介紹、防禦、取證、攻擊以及搶旗競賽共 5 項，在搶旗競賽中，講師規劃一系列題目，期望學員能透過競賽方式，充分發揮 5 天所學之知識與技術，已增加開發滲透測試工具經驗，課程規劃內容請詳見表 3 SEC573 訓練課程議程。

表 3 SEC573 訓練課程議程

日期	課程內容
9/23 9:00 ~ 17:00 Essentials Workshop with pyWars	▪ Python Syntax ▪ Variables ▪ Math Operators ▪ Strings ▪ Functions ▪ Modules ▪ Control Statements ▪ Introspection
9/24 9:00 ~ 17:00 Essentials Workshop with MORE pyWars	▪ Lists ▪ Loops ▪ Tuples ▪ Dictionaries ▪ The Python Debugger ▪ Coding Tips ▪ Tricks and Shortcuts ▪ System Arguments ▪ ArgParser Module
9/25 9:00 ~ 17:00	▪ File Operations ▪ Python Sets

Defensive Python	▪ Regular Expressions ▪ Log Parsing ▪ Data Analysis tools and techniques ▪ Long Tail/Short Tail Analysis ▪ Geolocation acquisition ▪ Blacklists and whitelists ▪ Packet Analysis ▪ Packet reassembly ▪ Payload extraction
9/26 9:00 ~ 17:00 Forensics Python	▪ Acquiring Images from disk ▪ Memory and the network ▪ File Carving ▪ The STRUCT module ▪ Raw Network Sockets and protocols ▪ Image Forensics and PIL ▪ SQL Queries ▪ HTTP Communications with Python built in Libraries ▪ Web communications with the Requests module
9/27	▪ Network Socket Operations ▪ Exception Handling

9:00 ~ 17:00 Offensive Python	▪ Process execution ▪ Blocking and Non-blocking Sockets ▪ Asynchronous operations ▪ The select module ▪ Python objects ▪ Argument packing and unpacking
9/28 9:00 ~ 17:00 Capture the flag	In this final section you will be placed on a team with other students. Working as a team, you will apply the skills you have mastered in a series of programming challenges. Participants will exercise the skills and code they have developed over the previous five days as they exploit vulnerable systems, break encryption cyphers, analyze packets, parse logs, and automate code execution on remote systems. Test your skills! Prove your might!

資料來源：SANS Network Security 2018 SEC-573 課程網頁 [3]

1. Essentials Workshop with pyWars

第一天課程可分為 Python 語法介紹與設定 Online judge 共 2 項，課程操作環境詳見表 4 課程操作環境:

表 4 課程操作環境

課程環境	名稱
作業系統	Linux
程式編譯器	Python 2.7.15 與 Python 3.7.1
Online judge	pyWars

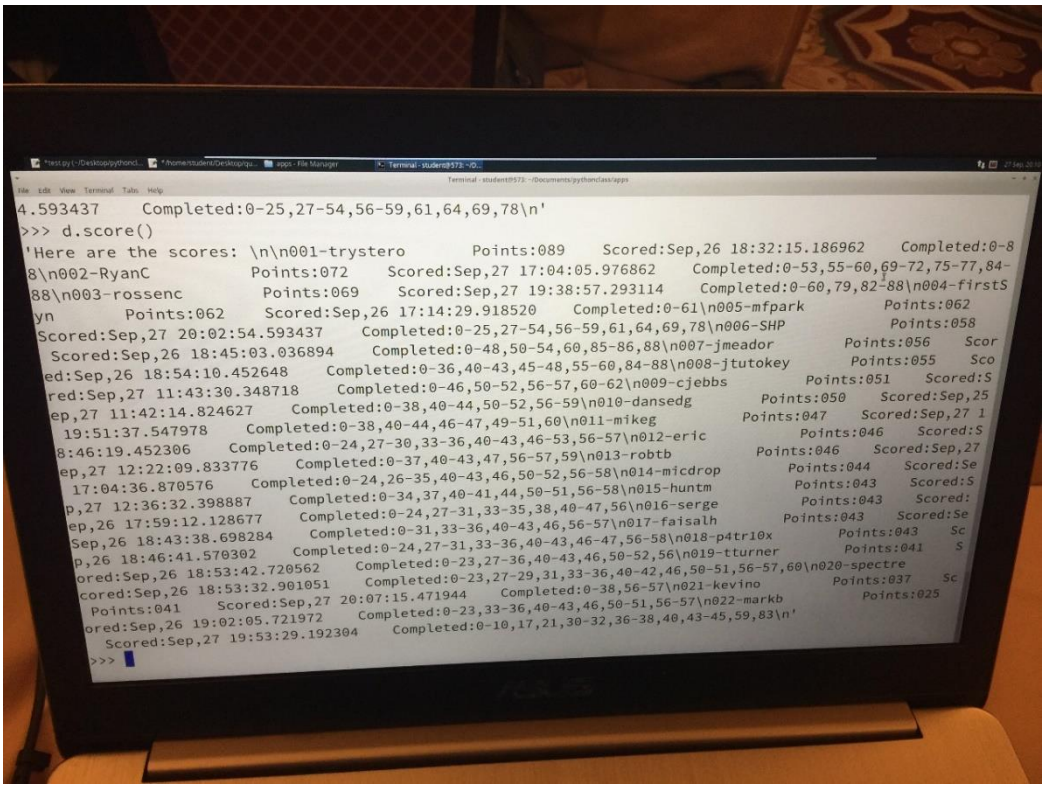
資料來源 : 本報告自行整理

(1) Python 語法介紹

本課程在第一天詳細介紹 Python 語法規則、變數、數學運算符號、字串、函式、模組撰寫、條件式流程以及內部自省模式，其中內部自省模式對於撰寫腳本十分有幫助，原因在於 Python 定義元素諸如變數、字串……等，都是以物件表示，每個物件都有其對應的內部函式可使用，因此如何針對物件找尋所需的內部函式十分重要，而內部自省模式正能提供此功能。

(2) 設定 Online judge

pyWars 為 SANS 自製 Online judge，可針對學員上傳的程式進行測試並告知答案是否正確，pyWars 提供 80 道題目，其中課程練習題約佔 40 題，剩餘 40 道題目取自 CTF(Capture The Flag)競賽，另外 pyWars 也提供記分板功能，上頭紀錄每個人解題的編號與題數，可以驗證每個學員的學習成效以及激起所有人的競爭心理，記分板請詳見圖 2 pyWars 記分板。



資料來源 : 本報告自行整理

圖 2 pyWars 記分板

2. Essentials Workshop with MORE pyWars

第二天課程為第一天課程的延伸，重點著重在 List、Tuples 及 Dictionaries 操作、Python Debugger 使用以及輸入參數設計共 3 種，以下將說明上述 3 種課程主題。

(1) List、Tuples、Dictionaries 操作

Python 針對多筆資料儲存方式可分成 List、Tuples、Dictionaries 共 3 類，其中 List 與 Tuples 差異在於，List 能對資料進行新增、刪除與修改，而 Tuples 僅只能新增，Dictionaries 屬於關聯式陣列，資料結構為 {'key':'value'}，其中 key 為鍵值，value 為數值，這表示每個鍵值都有其對應的數值，因為其資料結構的特性使得 Dictionaries 擁有快速查找某個鍵值對應數值的優點，其搜尋速度為 O(1)。

Mark 在課程中提到一個新手常犯的錯誤，在於當需要額外複製 List、Tuples 或是 Dictionaries 存取的資料時，經常忽略使用 copy 語法進行深度複製，單純只使用“=”承接資料，這會造成變數間資料共用，一旦資料改變

，所有承接資料的變數內部的資料通通都會改變，這會造成一個難以發覺的 bug，因此不得不小心。

(2) Python Debugger

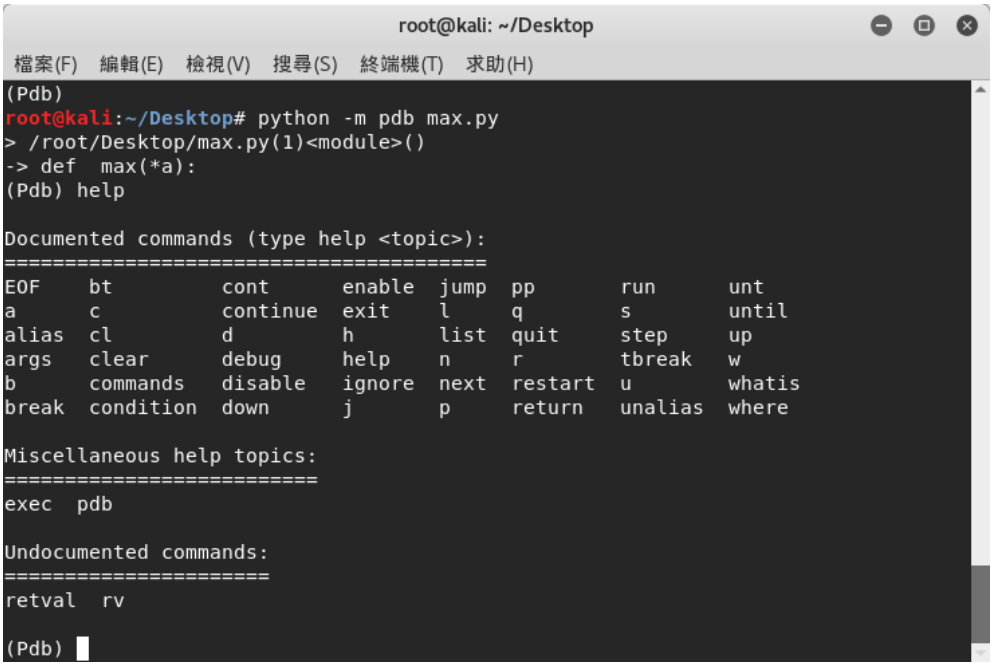
Python 內部提供一個除錯用模組稱為 Python Debugger(簡稱 PDB)，使用方法如下，詳見圖 3 進入 PDB 方式。

```
root@kali:~/Desktop# python -m pdb 檔案名稱.py
```

資料來源 : 本報告自行整理

圖 3 進入 PDB 方式

PDB 也有提供內部自省模式，如圖 4 所示，可以供查詢所需使用函數名稱，SEC573 課程僅針對幾個 Debug 時較常用的指令進行教學，如表 5 所示。



資料來源 : 本報告自行整理

圖 4 PDB 內部自省模式

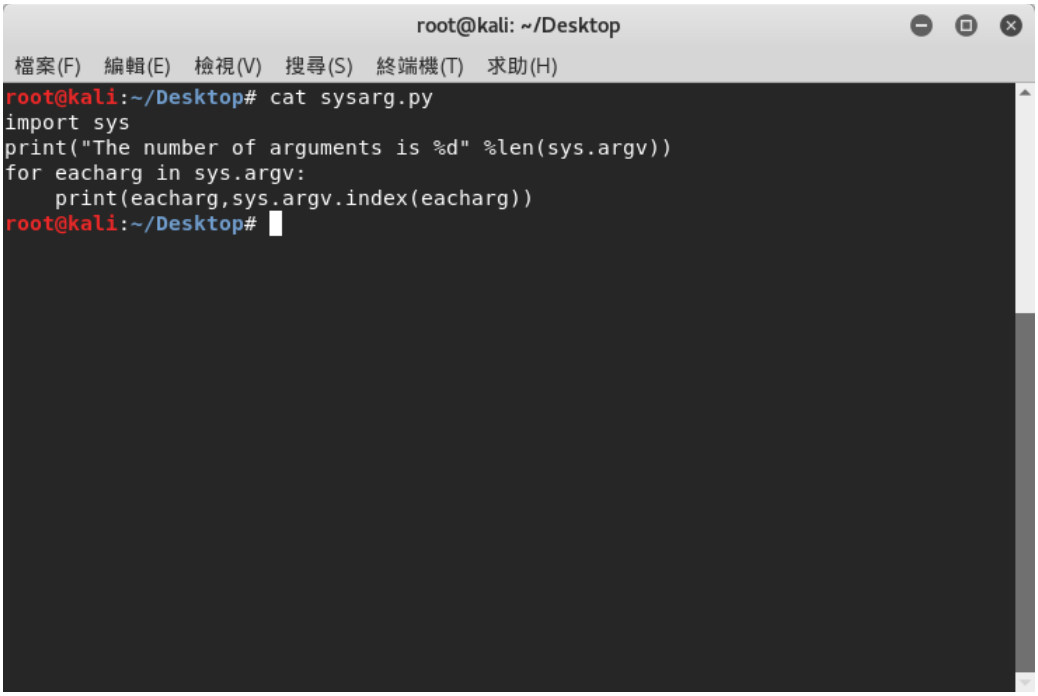
表 5 PDB 常用指令列表

指令	功能
b+數字	設置中斷點
r	繼續執行，直到當前函式返回
c	繼續執行程式
n	執行下一行程式
s	進入函式
p+變數名稱	印出變數
l	印出目前的程式片段
q	離開

資料來源 : 本報告自行整理

(3) 輸入參數設計

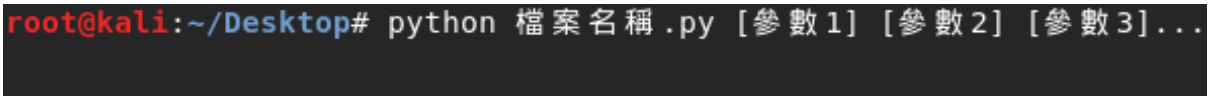
在撰寫滲透測試工具時，如何傳遞外部參數一直是很重要的問題，Python 提供相當簡潔的輸入參數設計功能，可供使用者在 Command Line(簡稱 CML)傳遞參數，舉例來說，我們撰寫以下程式，如圖 5 所示，此程式功能為列印所輸入參數，其中 sys.argv 為承接輸入參數的函式。



資料來源 : 本報告自行整理

圖 5 輸入參數列印

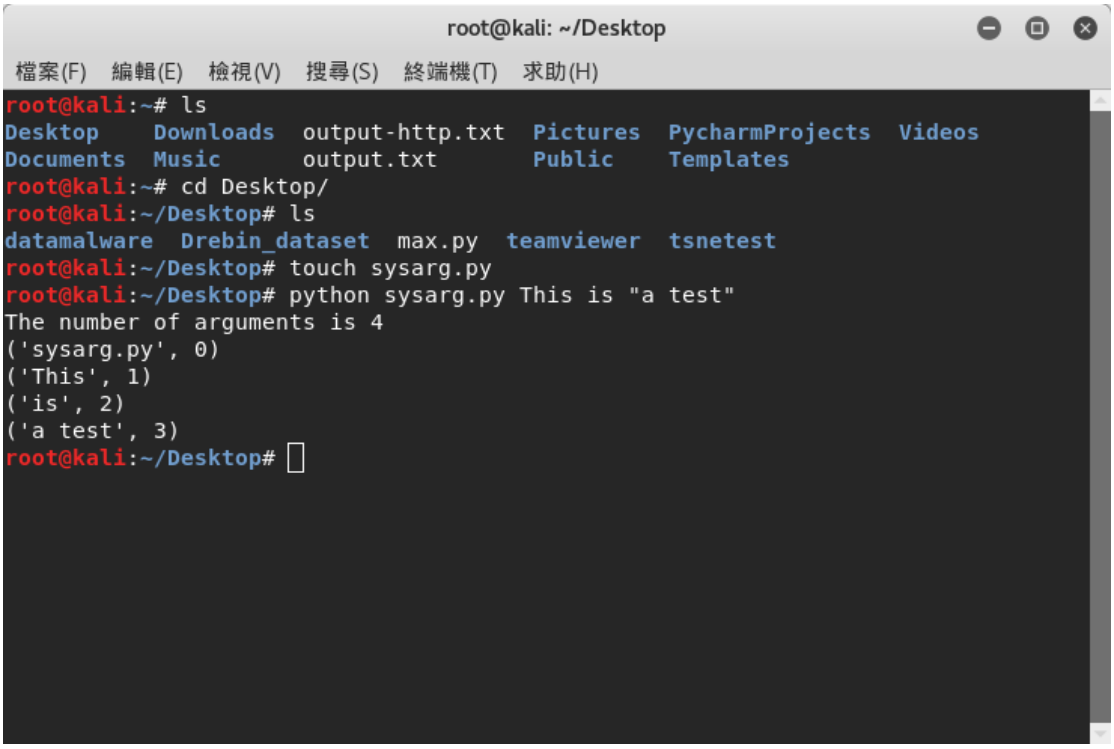
然後，我們執行上述程式，輸入方式如圖 6 所示，參數間以空格為間隔方式。



資料來源 : 本報告自行整理

圖 6 參數輸入方式

執行程式後如圖 7 所示，我們可以發現“檔名.py”竟也在輸入參數內，怎麼回事? 原因在於 sys.argv 是從輸入“python”之後開始承接後面訊息，所以也把檔名視為參數的一種，存在記憶體位置 sys.argv[0]中，接下來遇到空格後再將記憶體位置往下挪一格，將“This”存入 sys.argv[1]中，以此類推。



資料來源 : 本報告自行整理

圖 7 範例程式執行

當然，如果只單純接受參數數值，這樣會造成指令可讀性不足，面對此問題 Python 也提供 argparse 模組，用以解析 sys.argv，然後自動產生幫助與使用之提示訊息，可以讓參數使用文字指令，以達到更加方便的操作。

3. Defensive Python

第三天課程主題為防禦方式，如何有效且快速發覺系統是否被入侵，以及找出被入侵的跡象後研擬防範機制是防禦的主要用意，課程內容為主要探討檔案分析，檔案分析大致可分為文檔分析與網路封包分析共兩種，此外，在解析檔案時，如何有效率的從中獲得自己有興趣的資料是十分重要的問題，為了解決此類問題，科學家發展了正則表示式(Regular expression)，而在課程中，Mark 也針對正則表示式使用極大的篇幅說明，以下將針對正則表示式、文檔分析與網路封包分析進行說明。

(1) 正則表示式

無論是系統日誌檔(Log file)或是網頁原始碼(Page source code)，上頭每筆資料皆以特定格式存放，正則表示式可使用內部定義的符號建構資料的特定格式，然後再用比對的方式擷取符合格式的資料，正則表示式內部定義的符號如表 6 所示。

表 6 常用正則表示式符號

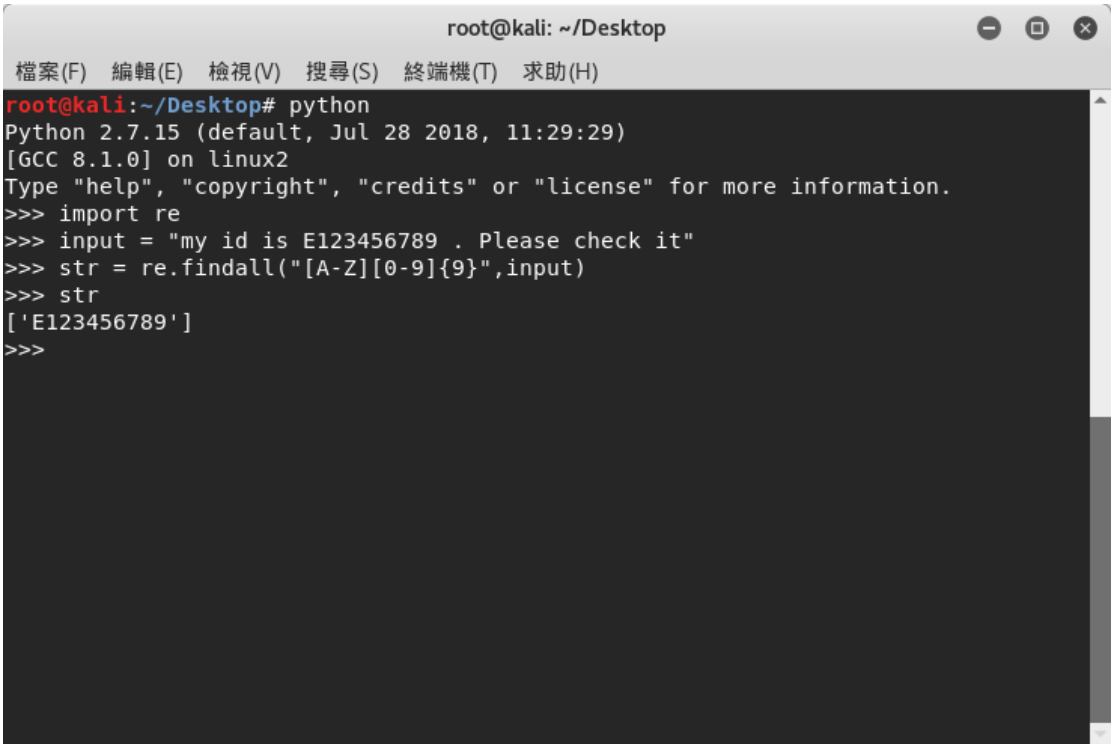
符號	說明
\	避開特殊字元。
^	符合輸入字串的開始位置。
$	符合輸入字串的結束位置。
*****	符合前面的子運算式零次或多次。
+	符合前面的子運算式一次或多次。
?	符合前面的子運算式零次或一次。
{n}	*n* 是一個非負整數。符合確定的 *n* 次。
{n,}	*n* 是一個非負整數。至少符合 *n* 次。
{n,m}	*m* 和 *n* 均為非負整數，其中 *n*<=*m*。最少符合 *n* 次且最多符合 *m* 次。
.	比對任何一個字元（但換行符號不算）。
(x)	取得符合 x 的子字串。
x|y	符合「x」或「y」字元。

[xyz]	比對中括弧內的任一個字元。
[^xyz]	比對不在中括弧內出現的任一個字元。
[a-z]	比對在中括弧內指定範圍的任一個字元。
[^a-z]	比對不在中括弧內指定範圍的任一個字元。
\b	比對英文字的邊界。
\B	比對非「英文字的邊界」。
\cx	比對控制字元（Control character），其中 X 是一個控制字元。
\d	符合一個數字字元。
\D	符合一個非數字字元。
\f	符合一個換頁符號。
\n	符合一個換行符。
\r	符合一個 Enter 符號。
\s	符合任何空白字元。
\S	符合任何非空白字元。
\t	符合定位字元（Tab）。
\v	比對垂直定位字元（Vertical tab）。
\w	比對數字字母字元（Alphanumerical characters）或底線字母（”_”）。

\W	比對非「數字字母字元或底線字母」。

資料來源 : 本報告自行整理

Python 有提供相對應的正則表示式模組 re， 針對 re 的使用範例如圖 8 所示，範例程式功能為擷取字串身分證字號，程式一開始呼叫 re 模組，然後針對字串“my id is E123456789 . Please check it”擷取身份證字號“E123456789”，這裡設定的正則表示式為“[A-E][0-9]{9}”，此式子表達的意思為，擷取資料格式為開頭為大寫的英文字母後面連結 9 個數字的資料，然後使用 findall 函式找到符合定義的資料後，用變數 str 承接，並列印內容。



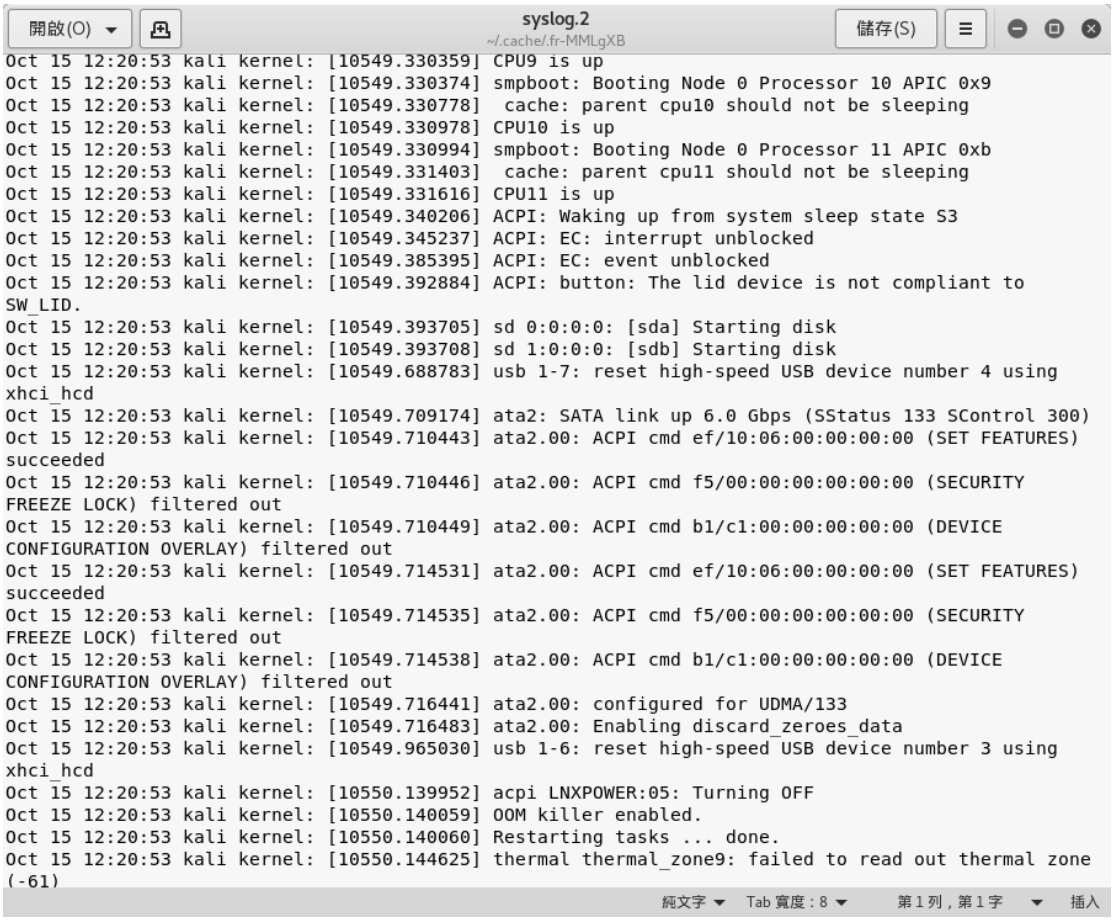
資料來源 : 本報告自行整理

圖 8 re 模組使用範例

(2) 文檔分析

舉例來說我們察覺某系統被駭客入侵，則我們必須解讀系統日誌，查找駭客入侵足跡並規劃防範機制。系統日誌記錄著系統登入時間、主機 ID、CPU 運作情況等……系統操作訊息，在資安事件處理過程中，系統日誌是協助追查出入侵者種種跡象的重要關鍵，課程內容以/var/lig/syslog.2.gz 為例子，其

中.gz 檔為一種壓縮檔格式，Python 針對壓縮檔處理也有提供 gzip 套件可使用，當我們在讀取壓縮檔後，可以發現 syslog.2，如圖 9 所示，這就是系統日誌檔，接著我們便可以使用正則表示式來擷取我們較感興趣的資訊。



資料來源：本報告自行整理

圖 9　Linux Logfile 節錄

(3)　網路封包分析

除了文檔分析外，分析網路封包也是防禦最主要的一環，舉例來說，當電腦被植入後門(Backdoor)後，駭客可以在受害者毫無察覺的情況下，竊取被害者的訊息、系統的主控權或者是電腦會被當成殭屍網路的一環，此時分析網路包流向、協定及其他資訊，有助於取得更有益於研擬防範機制的相關參考證據，針對網路封包的分析，Python 對此有提供 scapy 套件，可以用於多種協定網路封包，目前比較有名跟它類似的工具像是 Wireshark，scapy 除了可以配合 Python 寫出俱有特殊目的的封包分析方式，也可以像 Wireshark 一樣擁有嗅探(Sniffer)功能，也可以下條件式過濾封包。

4. Forensics Python

第四天的課程主要探討數據取證，從蒐集資料、定義資料格式、讀取資料，直到取出感興趣的資料，都有詳細說明。

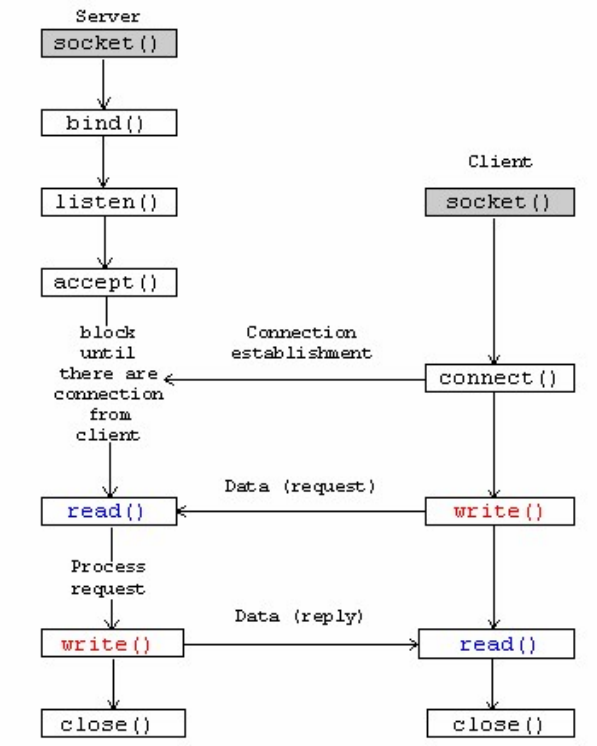
資料蒐集途徑大概可以分成硬碟、記憶體、網路共三者，課程以 Windows 與 Linux 兩系統為例，針對上述三者進行操作。在硬體儲存方面，必須先了解 Physical Drive 與 Logical Drive 定義，Physical Drive 為硬體本身，Logical Drive 是由系統在 Physical Drive 分割的虛擬硬碟，在 Windows 與 Linux 對於 Physical Drive 路徑的定義為\\.\PhysicalDrive0 以及/dev/sda，反之 Logical Drive 為\\.\c:以及/dev/sda1。針對記憶體蒐集資料，Windows 會使用內建的 Winpmem.exe 建立一個稱為\\.\pmem 檔案可以供使用者讀取目前記憶體資訊，Linux 則需要安裝第三方套件 FMEM 建立/dev/mem 才能供使用者讀取目前記憶體資訊。針對使用網路蒐集資料，Windows 與 Linux 皆可使用 Socket 蒐集資料。

蒐集完資料後，則可以開始讀取資料內容，每個資料都有其專屬的資料格式，對此 Python 均有相對應的套件可以解析資料格式，例如在第三天學到的 scapy 可以針對 TCP 或 UDP 封包進行資料格式解析，gzip 可以解析壓縮檔，也可使用第三方套件 PIL 讀取圖檔。

在開始讀取資料時，針對資料內容，可以使用正則表示式定義感興趣資料的擷取語法，擷取資料。另外對於圖檔，SEC573 使用 PIL 解析圖片格式，解析圖片格事後可以在特定的資料欄位取出資料，例如 TAG 34853 存放的是拍照時的 GPS 位置。

5. Offensive Python

第五天課程主要探討 Socket 的使用方式並使用 socket 製作簡單後門(Backdoor)，Socket 是不同程序(Process)之間的溝通方式，其傳遞資料的方式，除了可用程序與程序之間傳遞資料，也可以很輕鬆地使用 TCP 或 UDP 協定跨越主機傳遞資料，要建立一個 Socket 連線，須建立服務端(Server)與用戶端(Client)，服務端與用戶端溝通方式如圖 10 所示。



資料來源：TCP Socket Programming 學習筆記[4]

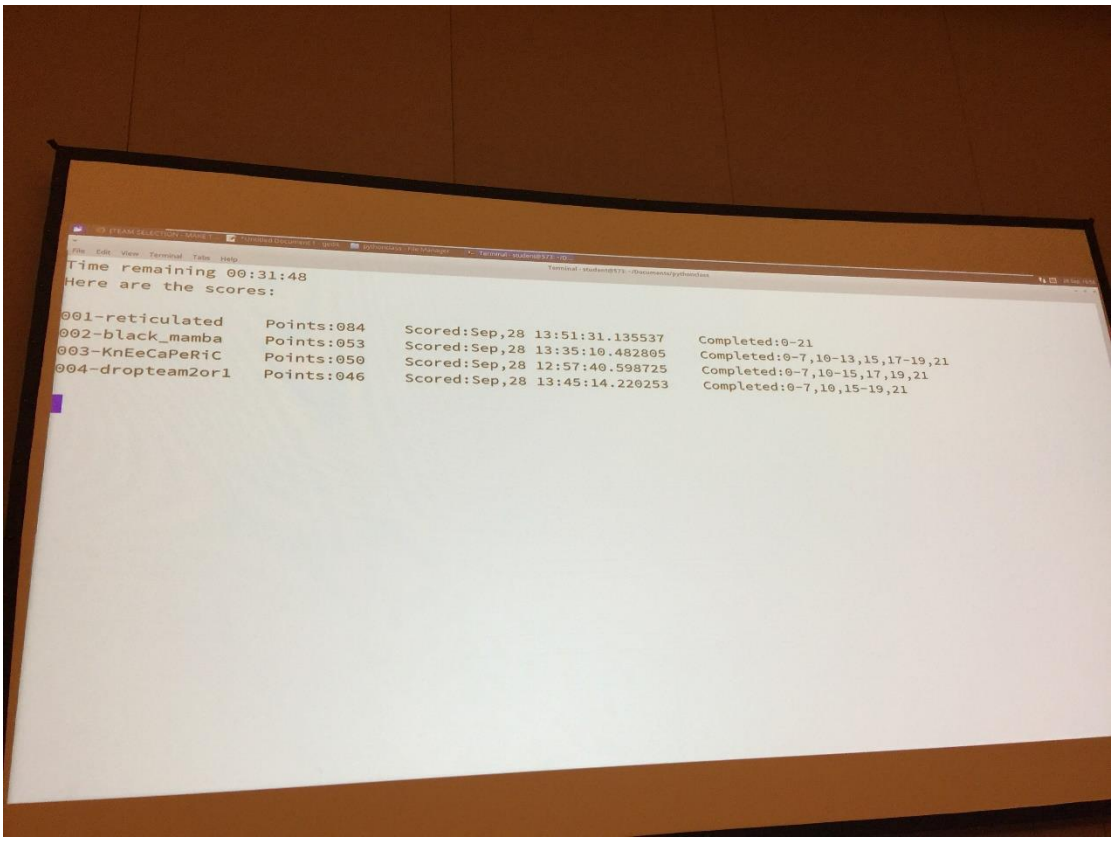
圖 10 Client 與 Server 通訊流程

建立連線前，Server 需綁定(Bind)主機的 IP 位址，之後開始進行傾聽(Listen)是否有用戶端要求連結，當傾聽到用戶端的連結要求後，便可以建立連線(Accept)進行溝通。

在 SEC573 課程中，Mark 使用 socket 製作一個簡單的後門，其中服務端放在作業系統為 Linux 的機器上，用戶端則放在作業系統為 Windows 的機器，然後在運用自製的指令透過用戶端向服務端要求傳輸檔案，傳輸檔案為 syslog.2.gz，當 Client 收到檔案後，便能使用正則表示式擷取資料後，進行分析。Mark 提及此後門雖然相當粗糙，功能也不多，但是只要持續改善並增加功能，此後門也可能變成強大的武器。

6. Capture the flag

第六天是搶旗競賽，學員們必須組織隊伍，每一隊伍共 3~5 人，每隊學員必須運用 5 天所學之知識與技術，透過一連串的小組討論與合作，在 4 小時內解決 CTF 伺服器上公布的 22 道問題，每道題目會依題目困難度給予相對應的分數，簡單題目 1 分，最困難的題目 8 分，每隊隊伍解決問題的進度將會公布在計分板上，記分板如圖 11 所示。



資料來源：本報告自行整理

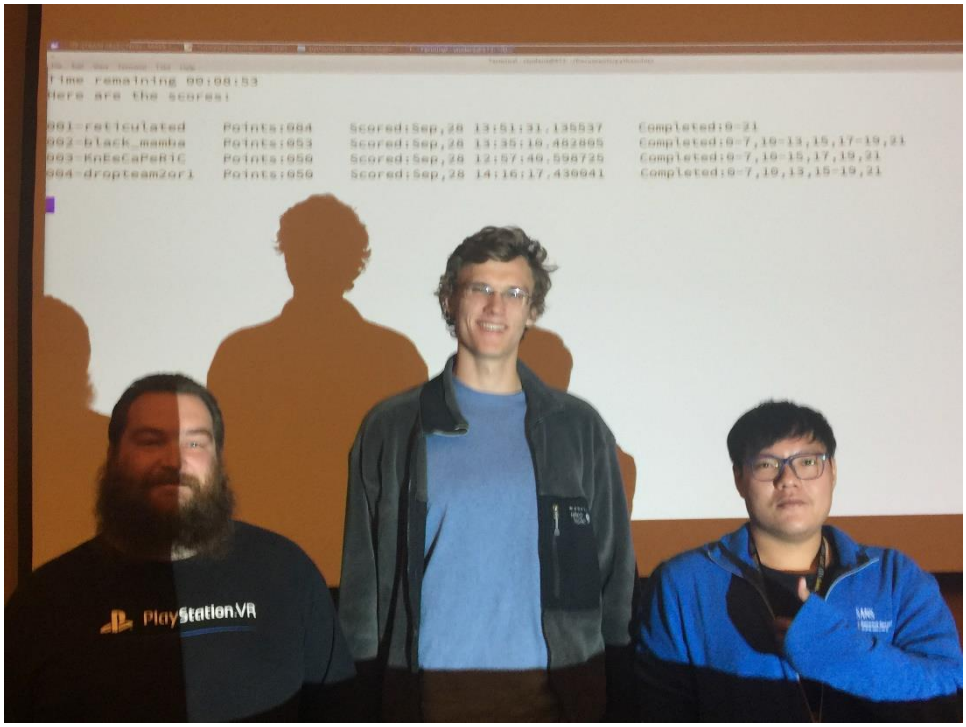
圖 11　搶旗競賽記分板

獲勝的隊伍或是 22 道題目全解完之隊伍可以獲得一個 SEC573 課程獎章，獎章如圖 12 所示。

資料來源：本報告自行整理

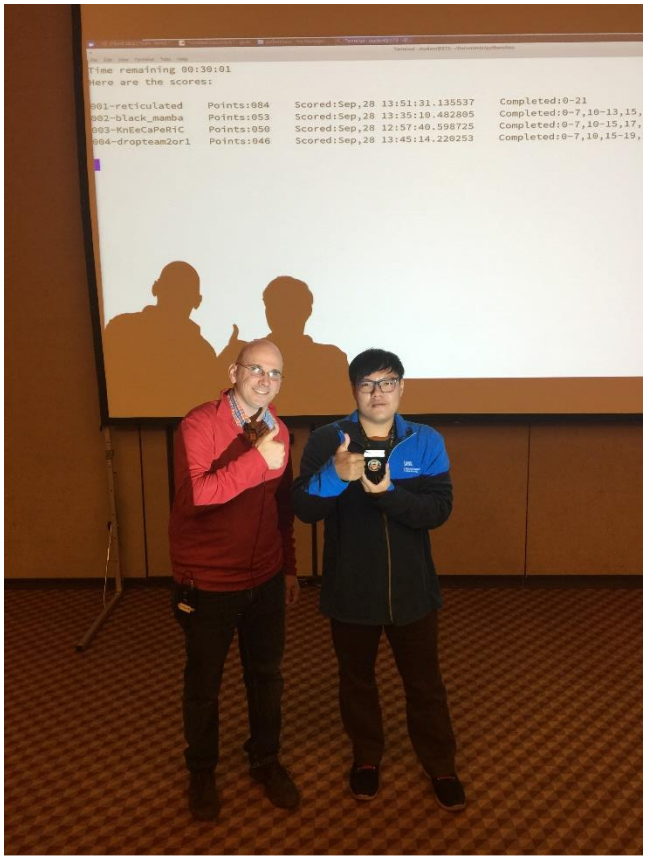
圖 12　搶旗競賽獎章

我們的隊伍名稱為 “reticulated”，組員共 3 個人，其他 2 人分別為 Gerolstein 以及 Ryan，我們共計花費 3.5 個小時順利解決全部題目，並拿到獎章，與組員與課程導師合照如圖 13-14 所示。



資料來源：本報告自行整理

圖 13　Gerolstein(左) 、Ryan(中)與我(右)



資料來源：本報告自行整理

圖 14　Mark(左)與我(右)

參、心得與建議

本次參加 SANS Network Security 2018 中 FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques 與 SEC573: Automating Information Security with Python 兩門課程，除了對 windows 惡意程式逆向工程相關技術獲益良多外，也對於如何使用 Python 撰寫高效率資安自動化腳本有更實際的操作。反思目前工作內容與資安組業務後，有兩點心得與建議陳述如下：

一、鼓勵同仁撰寫技術報告，營造資安知識管理與分享之環境

綜觀 SANS 課程，可發現講師授課內容皆是靠平日大量閱讀、咀嚼、歸納後，才能在課堂上提供精彩授課內容。然而，若無平日保留研究筆記，做好知識管理，則肯定無法以系統化方式將內容呈現給學員，提昇學員學習效率。所產出之技術報告，依據反饋與新技術之進步而精進，對撰寫者自己獲益更甚。因此，希望在資安組內，鼓勵同仁將日常分析惡意程式或其他資安技術心得，撰寫小篇幅筆記，並分享給其他同仁，營造樂於分享的環境，提升彼此技術能量。

二、只有分享觀點才能對解決問題產生直接性效益

在得知要參與 SEC573 課程，令我十分興奮，雖然我經常使用 Python 開發程式，但是一些相當基礎的開發知識確實不夠，SEC573 課程的導師 Mark 有相當深厚的技術背景，上課方式幽默風趣，當我向他提出一些問題，Mark 總是能有條不紊的回答並解決我的困惑。

從這六天的課程，我見識到老外上課的風格，台灣與之相比相差甚遠，舉例來說，每當 Mark 針對課程章節提出問題時，每個學員對於問題皆踴躍分享觀點並針對觀點討論，最後課堂上的每個人得以受惠，但是在台灣，由於民族性或是其他因素，幾乎很少討論，使得很多非常優秀的觀點只能埋藏心底，著實可惜。

肆、參考文獻

[1] SANS 課程網站, https://www.sans.org/event/network-security-2018.

[2] SANS Network Security 2018 FOR610,https://www.sans.org/event/network-security-2018/course/reverse-engineering-malware-malware-analysis-tools-techniques.

[3] SANS Network Security 2018 SEC-573, https://www.sans.org/course/automating-information-security-with-python.

[4] TCP Socket Programming 學習筆記, http://zake7749.github.io/2015/03/17/SocketProgramming.