科技部資訊安全技術研發專案計畫

『系統測試計畫書』

System Test Plan Document

以雲端平台為基礎之智慧型安全分析與管理系統(1/3)

MOST 106-2221-E-110-017-MY3

陳嘉玫

國立中山大學 資訊管理學系

Department of Information Management
National SunYat-sen University

107/07/01

文件版本修正履歷表

編號： MOST 106-2221-E-110-017-MY3

名稱： 以雲端平台為基礎之智慧型安全分析與管理系統(1/3)

修訂次序	核準日期	版本	修訂內容
1	107/05/01	1.00	發行初版

版本: 1.00

目錄

文件版本修正履歷表 I

目錄 II

1. 簡介 4

1.1 測試範圍 6

1.2 接受準則 6

2. 測試環境 8

子計畫一測試環境 8

子計畫二測試環境 10

子計畫三測試環境 13

2.1 硬體規格 14

2.2 軟體規格 16

2.3 測試資料來源 17

3. 測試時程、程序 18

3.1 測試時程 18

3.2 接受測試程序 18

總計畫接受測試程序 19

子計畫一接受測試程序 19

子計畫二接受測試程序 21

子計畫三接受測試程序 23

3.3 整合測試 24

3.4 壓力測試 24

4. 測試案例 25

總計畫之測試案例 25

子計畫一測試案例 27

子計畫二測試案例 30

子計畫三測試案例 32

5. 測試結果與分析 34

總計畫 34

子計畫一 34

子計畫二 37

子計畫三 41

附錄 A 追朔表 43

總計畫之追溯表 43
子計畫一追溯表 43
子計畫二追溯表 44
子計畫三追溯表 45

1. 簡介

本計畫「以雲端平台為基礎之智慧型安全分析與管理系統」主要目的為發展提出一套以 Hadoop 為基礎之智慧型資安事件分析與管理平台，簡稱 HiSAMS（Hadoop intelligent Security Analysis and Management System)。由總計畫及三個子計畫組成。主持人與相關計畫如表一所示。

表1 以雲端平台為基礎之智慧型安全分析與管理系統研究各計畫列表

計畫項目	主持人	計畫名稱	科技部編號
總計畫暨子計畫一	陳嘉玫教授	以雲端平台為基礎之智慧型安全分析與管理系統	MOST 106-2221-E-110-017-MY3
子計畫一		以 Hadoop 雲端平台為基礎之智慧型安全分析與管理系統	
子計畫二	林輝堂教授	應用自律控制於物聯網惡意攻擊之偵防研究	MOST 106-2221-E-006-024-MY3
子計畫三	楊竹星教授	新型態惡意程式之網路流量異常分析與軌跡追溯機制之研究	MOST 106-2221-E-006-025 -

各子計畫簡介分別描述如下：

總計畫簡介：

雲端運算、巨量資料、物聯網等關鍵科技改變組織與企業服務模式與應用。雲端運算日漸成熟，大量敏感數據存放在雲端中，因此雲端安全性成為現今資訊產業必須重視的課題；而物聯網的出現，提供便利服務，但是一般物聯網設備因為功能有限，建置時可能未做良好的安全設定，因此可能會輕忽其可能之資訊安全議題。總計畫系統可彙整多元的工作日誌與網路資料，建立關聯規則和多層次與長時間的異常行為關連性分析，以偵測新型態多元化的攻擊事件，並採用雲端叢集式架構與分散式儲存方式，有效率的分析數據，正確且快速偵測從不同網路層或是物聯網來的入侵行為。

子計畫一簡介

子計畫一的目的為發展並建置一套非 signature-based 的偵測惡意程式之機制。由於駭客常利用 DLL Injection 的技術將惡意程式碼注入到正常的處理程序中，常見手法之一是利用系統合法的 API 來注入惡意程式碼。因此，本系統會從靜態分析中找出執行檔是否有遭到 PE 感染的攻擊，抓出惡意執行檔與正常執行檔的差異性，並辨別出惡意執行檔更改正常執行檔之欄位與特徵。

子計畫二簡介

子計畫二之主要目的為物聯網惡意攻擊之偵防系統。計畫初期針對物聯網設備運算能力、儲存空間和能源的限制，設計一套輕量化且安全的認證機制，經由可信任的第三方機構和 PUF 認證物聯網設備身分，以及加密訊息使用一次性金鑰且提出更新的流程，保證訊息傳輸的完整性和機密性。另外，在物聯網環境中設置蜜罐(honeypot)，紀錄駭客攻擊物聯網設備的行為，根據不同攻擊行為提供警訊給閘道，並持續追蹤後續攻擊動作。最終目的為保護物聯網免於惡意設備的加入，保護訊息安全的傳輸和透過觀測攻擊物聯網的行為，發現未知的漏洞而及時修補與防範。

子計畫三簡介

在大規模網路環境中，單點與單一資安設備的設置，無法有效偵測與防禦新型態攻擊行為的發生，最主要原因為攻擊碼更新緩慢，加上資安設備擺放的位置不當或是資安設備偵測規則設置不當，亦無法有效偵測新型態攻擊行為。這種情況下，在網路閘道口收集網路流量(Netflow)，將能夠含擴大規模網路環境中所有進出的流量加以分析。子計畫三之目的為研發新型態攻擊手法之行為態樣分析模型，協助轉化分析結果為防禦規則或特徵碼，並能加速新型態攻擊之防禦。此外，本計劃也將建立服務主機異常行為之告警，並配合目標性網路封包的側錄與歷史性網路流量的追溯分析，完成攻擊軌跡全貌建構與追溯、感染範圍之判定。

1.1 測試範圍

本文件主要是建置以 Hadoop 為基礎之智慧型資安事件分析與管理平台，簡稱 HiSAMS（Hadoop intelligent Security Analysis and Management System）。在系統整合前，必須先確認所有的設計模組皆可正確運行，並輸出預期的成果。故本計畫著重於及接受度測試(Acceptance Test)。本文件內容會對系統規格進行相關的測試計畫進行詳細說明。並希望透過此文件之描述與實踐，達到順利進行測試工作之目的。

1.2 接受準則

本測試計畫總計畫及各子計畫需要滿足的測試接受準則依計畫別分別詳述如下。且測試程序需要依照本測試計畫所訂定的程序進行，所有測試結果需要能符合預期測試結果方能接受。測試以測試案例為單位，當測試未通過時，則重新檢測系統設計並進行必要之修正後，再重新進行測試，至通過為止。

表2 接受測試需求項目表

測試需求編號	必要性	需求內容
HiSAMS-TR-001	必要	測試避免XSS及SQL Injection之相關網站弱點
HiSAMS-TR-002	必要	子計畫內容可互相分享
PEI-TR-001	必要	PDM輸入格式必須符合系統需求
PEI-TR-002	必要	ADM能夠分辨正常與異常的API距離
PEI-TR-003	必要	RTVM能正確算出Import Table的相似程度
LAD-TR-001	必要	第三方驗證中心資料庫建置並能存取資料
LAD-TR-002	必要	可以讀取物聯網設備PUF並存至第三方的資料庫
LAD-TR-003	必要	確認閘道與第三方驗證中心以及物聯網設備與閘道能否正常進行通訊
LAD-TR-004	必要	閘道和物聯網設備能夠進行雙向身分驗證
LAD-TR-005	必要	訊息傳輸時能夠正確進行加解密和確保訊息完整性
LAD-TR-006	必要	訊息傳遞所使用的加密金鑰能夠正確產生及更新
MTA-TR-001	必要	Input Module可讀取Netflow資訊
MTA-TR-002	必要	Input Module可讀取Host Log
MTA-TR-003	必要	Processing Module可顯示資料
MTA-TR-004	必要	Report Module可顯示攻擊資訊

表3 整合測試需求項目表

測試需求編號	必要性	需求內容
HiSAMS-TR-001	必要	測試避免XSS及SQL Injection之相關網站弱點
PEI-TR-001	必要	PDM輸入格式必須符合系統需求
LAD-TR-001	必要	第三方驗證中心資料庫建置並能存取資料
LAD-TR-002	必要	可以讀取物聯網設備PUF並存至第三方的資料庫
LAD-TR-003	必要	確認閘道與第三方驗證中心以及物聯網設備與閘道能否正常進行通訊
MTA-TR-001	必要	Input Module可讀取Netflow資訊
MTA-TR-002	必要	Input Module可讀取Host Log

表4 效能測試需求項目表

測試需求編號	必要性	需求內容
HiSAMS-PR-001	必要	各子計畫相關之資料庫查詢，需於15秒內完成
PEI-PR-001	必要	拆解到分析可於1分鐘內完成
MTA-PR-001	必要	Report Module可於15秒內回應Last Day之Log

2. 測試環境

對於本系統總計畫及各子計畫進行系統測試的環境說明分別詳述如下：

子計畫一測試環境

本研究主要目標是分辨可執行檔是否遭到 PE 感染，由於此攻擊的手法層出不窮，面對與日俱增的攻擊事件，只針對特定方法進行防衛往往會措手不及。因此，本研究利用兩種 PE 感染應用程式會有的特性進行偵測。API 距離法，受到汙染的應用程式其 API 距離會和預先編譯好的存在一個距離差，當兩兩 API 相減得到的差值便當作本研究第一階段驗證 PE Infection 的門檻值；RVA Import Table Dump，受到 PE 感染所汙染的應用程式的 RVA Import Table 會有重複的部分，如有抓到重複則可認為該執行檔可能為惡意。在本系統中，主要包含下列三個模組：

1. PE 拆解模組(PE Disassembly Module, PDM)
   該模組的主要功能是自動化掃描系統的執行檔，並將執行檔進行拆解，以提供之後模組使用。
2. API 距離分析模組(API Distances Module, ADM)
   該模組主要功能是分析每個 PE 檔案中的 API 距離，然後透過系統運算後會得到一筆差值。運用該差值來作為第一階段驗證 PE Infection 的門檻值。
3. RVA Tablet 差異度分析模組(RVA Table Variance Module, RTVM)
   該模組主要目的是尋找 RVA Import Table 的位置再把其內容匯出來，透過公式進行計算取得實際位置。之後將將取得的訊息轉成特殊特徵值，當作本研究二階段的特徵值。

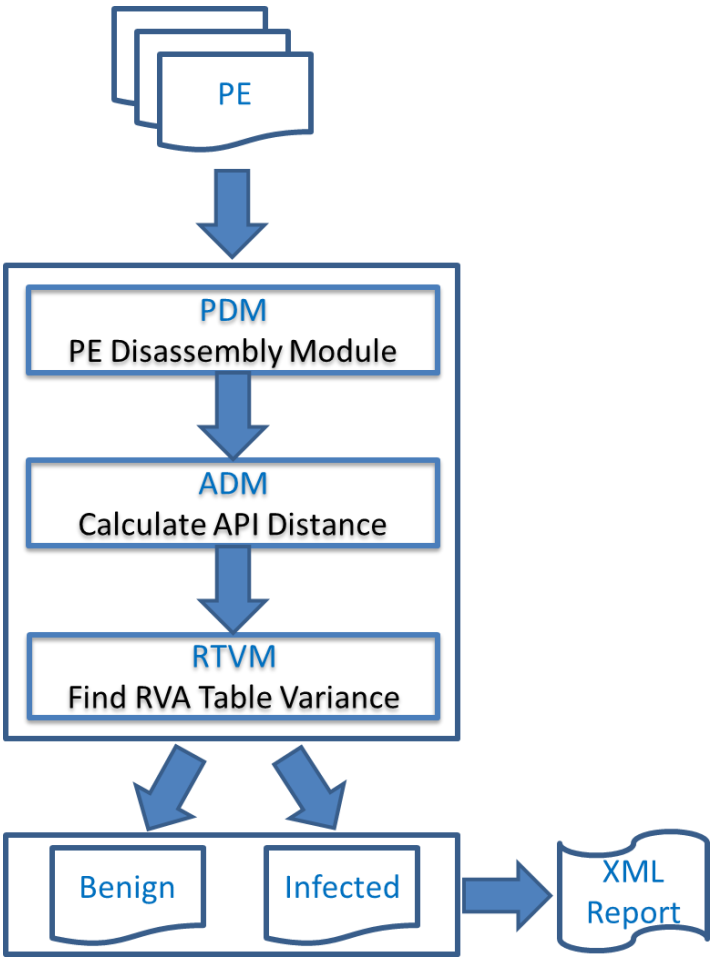


圖1 子計畫一的測試環境

子計畫二測試環境

本研究針對物聯網設備資源上的限制(運算能力、儲存空間、能源)設計輕量化且安全的身分驗證，和加密金鑰建立和更新，以及保障訊息的傳輸安全。由於可信任的第三方機構和 PUF 可以大幅降低物聯網設備所需要的資源，因此我們將基於此，設計一套輕量化的雙向身分驗證流程，保護物聯網不受惡意設備隨意加入網路。另外為了避免金鑰被破解的可能，我們設計金鑰建立和更新流程，定時更新加密金鑰。最後在訊息傳輸的部分，我們確保訊息的機密性和完整性，保證訊息的傳遞安全。

1. 輕量化雙向身分驗證 (Lightweight Bidirectional Authentication, LBA)

首先，物聯網設備在製造完成出廠前，利用設備的 PUF，向驗證中心(Authentication Center, AC)註冊並產生一把初始的對稱式加密金鑰，驗證中心儲存註冊之物聯網設備的 PUF、驗證時使用的運算函式以及加密金鑰，建立此物聯網設備的註冊紀錄，以便作為將來身分驗證之用。

身分驗證機制流程如下：

(1) 首先物聯網設備欲加入到一個物聯網時，設備會先產生一個隨機數值(Nonce)製作管理端節點驗證值。

(2) 將該隨機數值利用與驗證中心共有的加密金鑰進行加密，並製作加入請求並送至物聯網管理節點。

(3) 物聯網管理節點收到加入請求後，對於任何物聯網設備進行身分驗證，可向驗證中心發出請求身分驗證，傳送此身分驗證的請求前，先產生管理端點的數位簽章以及一個隨機數值，再使用驗證中心的公鑰加密，並將來自設備的加密訊息一起傳送至驗證中心。

(4) 驗證中心收到身分驗證請求後，使用驗證中心的私鑰解開，再使用管理節點的公鑰驗證簽章的合法性。

然後再使用與設備的加密金鑰解開來自設備的加密訊息，搜尋其憑證管理資料庫，取得該設備的 PUF 和驗證運算函式，然後利用 PUF 和來自設備所送來的隨機值使用驗證運算函式產生一個管理節點驗證值，並將來自管理節點的隨機數值作為挑戰值，接著將管理端點產生的隨機數值以及來自設備的隨機數值，利用設備的 PUF 和驗證運算函式產生設備驗證值。

最後將管理端節點驗證值以及挑戰值利用與設備共有的加密金鑰進行加密，並且將設備驗證值以及設備所產生的隨機數值用管理節點的公鑰進行加密，之後傳回物聯網管理節點。

(5) 物聯網管理節點收到之後，使用自己的私鑰將加密訊息解開，取得設備驗證值和設備所產生的隨機數值。然後利用自己產生的隨機數

值和設備產生的隨機數值製作一把加密金鑰。最後物聯網管理節點由驗證中心加密後的管理節點驗證值和挑戰值傳送至物聯網設備，請求驗證。

(6) 物聯網設備收到之後，使用與驗證中心共同擁有的金鑰解開取得管理節點驗證值和挑戰值。

(7) 確認來自驗證中心產生的管理節點驗證值是否與設備所產生的相同，接著再結合此挑戰值和自己的PUF經由同樣的驗證運算函式，產生設備驗證值。然後再利用挑戰值與前述所提到設備所產生的隨機數值，製作一把與管理節點相同的加密金鑰。

(8) 將設備驗證值使用上一個步驟所產的對稱式金鑰進行加密，回傳物聯網管理節點，作為挑戰的回覆。

(9) 物聯網管理節點收到後將之解密取得設備驗證值，並與來自驗證中心的設備驗證值比對，看是否相同。如果相同則此物聯網設備通過身分認證，否則身分驗證失敗。

系統測試環境如下圖所示:

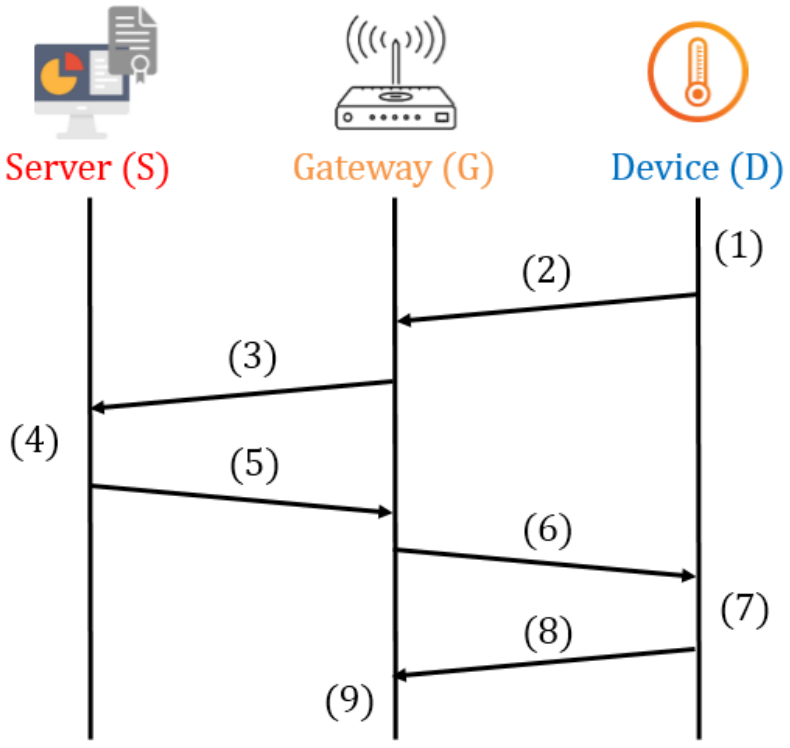


圖2 子計畫二之驗證機制測試環境圖

2. 金鑰建立和更新以及訊息安全 (Key generation and renewal and Information Security, KGRIS)

為了保護加密金鑰不被破解，我們設計金鑰產生及更新機制，讓物聯網設備和閘道定期更新其加密金鑰，以提高資訊傳輸之安全。在訊息傳遞的過程中，我們會以 AES-128 對訊息進行加密，並且使用產生訊息的 HMAC 來確保完整性。

加密金鑰產生和更新流程:

(1)閘道產生隨機值 Nonce_G

(2)閘道使用加密金鑰 K_{GD} 加密 Nonce_G 傳送給物聯網設備

(3)物聯網設備使用 K_{GD} 解密取得 Nonce_G ，並產生隨機值 Nonce_D

(4)物聯網設備使用加密金鑰 K_{GD} 加密 Nonce_D 傳送給閘道

(5)閘道與物聯網設備使用金鑰產生機制(Key Derivation Function,KDF),利用 Nonce_G、Nonce_D 和 K_{GD} 製作一把新的加密金鑰$\widehat{K}GD$ 如下:

$\widehat{K}GD = \text{KDF}[\text{Nonce}_D, \text{Nonce}_G]_{K_{GD}}$

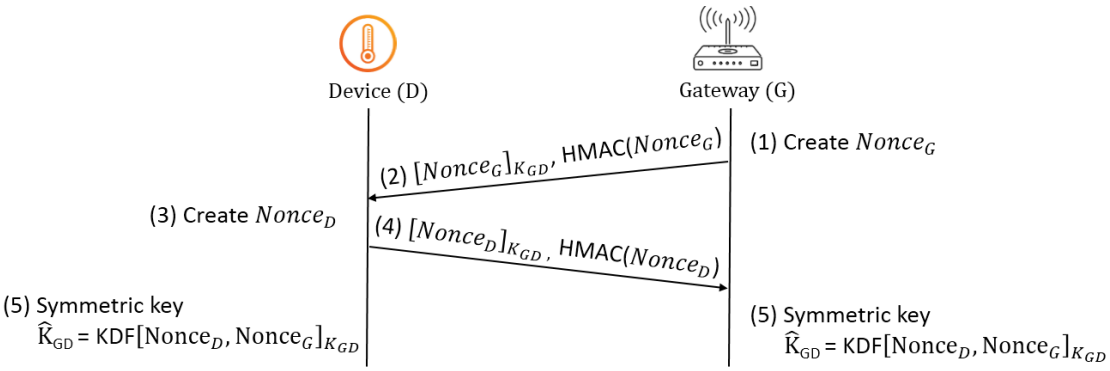


圖3 子計畫二之加密金鑰更新測試環境圖

子計畫三測試環境

本研究擬利用開放源碼為基礎建置大數據資安數據分析系統，並應用此大數據資安數據分析系統，完成(1)建立新型態攻擊手法之行為態樣分析模型，應用於網路流量(Netflow)有效偵測攻擊，(2)建立重點服務主機之正常行為分布曲線，支援服務主機異常行為之告警。(3)目標性網路封包的側錄與歷史性網路流量的追溯分析，完成攻擊軌跡全貌建構與追溯、感染範圍之判定。

計畫第一年預計完成之目標為: 建立以開放源碼為基礎建置大數據資安數據分析系統，開發新型態攻擊手法之行為態樣分析模型。系統中主要模組有三：

Input Module:

Input Module 之功能是建立接收各式日誌之介面，於日誌主機上接收相關日誌。目前可接收之日誌有 HoneyPot、Netflow 日誌及應用程式日誌。

Processing Module:

Processing Module 之功能為處理接收到之日誌，針對收到日誌的格式進行正規化之處理、儲存及比對相對應之黑名單。

Report Module:

Report Module 主要功能為回報及產生相關視覺化之圖表。針對攻擊，此模組亦會產生攻擊之預測。

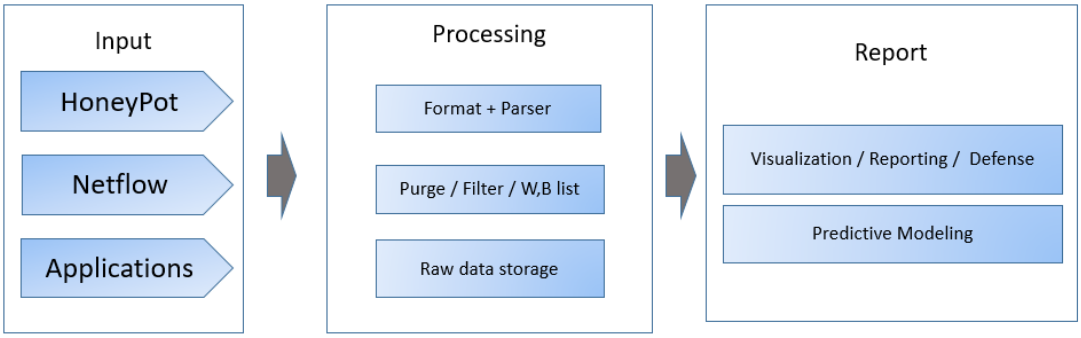


圖4 子計畫三架構

2.1 硬體規格

本系統總計畫及各子計畫關於測試環境所需的硬體規格說明，如下列所示：

總計畫之硬體規格

- 以雲端平台為基礎之智慧型安全分析與管理系統
  - Intel Core i3 2.0GHz 或以上相容的處理器
  - 2 GB 以上的記憶體
  - 1 GB 以上剩餘的磁碟空間(網頁及相關關聯分析程式)
  - 10 GB 以上剩餘的磁碟空間(視資料庫內容增加)

子計畫一硬體規格

- 以 Hadoop 雲端平台為基礎之智慧型安全分析與管理系統
  - Intel Core i3 2.0GHz 或以上相容的處理器
  - 2 GB 以上的記憶體
  - 100 MB 以上剩餘的磁碟空間(系統)
  - 10 GB 以上剩餘的磁碟空間(視樣本數量增加)

子計畫二硬體規格

- 應用自律控制於物聯網惡意攻擊之偵防研究
  - Raspberry pi3
  - CPU: 1.2 GHz 64-bit quad-core ARM Cortex-A53
  - 記憶體: 1GB LPDDR2
  - SD 卡容量:16GB

子計畫三硬體規格

- 惡意流量分析系統
  - Intel Core2Quad 2.66GHz 或以上相容處理器

- 6 GB 以上的記憶體
- 1GB 以上剩餘磁碟空間 (ELK 程式本身)
- 50GB 以上剩餘磁碟空間(視日誌數量增加)

2.2 軟體規格

總計畫軟體規格

- 以雲端平台為基礎之智慧型安全分析與管理系統
  - 作業系統：Ubuntu 14.04 LTS 或更新版本
  - 網頁伺服器：Apache 2.0 以上或相容版本
  - 資料庫管理系統：MySQL 5.0 以上或相容版本
  - 其他軟體元件：PHP 5.0 以上或相容版本

子計畫一軟體規格

- 以 Hadoop 雲端平台為基礎之智慧型安全分析與管理系統
  - 作業系統：Windows Vista x86-64 或更新版本
  - Python 程式語言版本：2.7.0 以上或相容版本

子計畫二軟體規格

- 應用自律控制於物聯網惡意攻擊之偵防研究
  - 作業系統：Raspbian
  - 其他軟體元件: Python3、Mysql

子計畫三軟體規格

- 惡意流量分析系統
  - Elastic Search: 2.3 (建議版本)
  - Logstash: 2.3 (建議版本)
  - Kibana: 4.5 (建議版本)

2.3 測試資料來源

本系統總計畫及各子計畫關於測試期間所需的測試資料來源及數量，說明如下：

子計畫一測試資料來源

本研究資料來源來自於某政府機關於 2015 年被攻擊的 APT 樣本。透過真實蒐集到的惡意樣本，測試能否偵測到最新的 PE 感染。

子計畫二測試資料來源

本研究資料來源全是由本研究所產生。

子計畫三測試資料來源

本研究資料來源是利用真實學校系所網路蒐集之網路流量之 Netflow log 資料，其資料包括系所的所有進出的 log。透過真實學校系所中網路流量分析，偵測相關網路攻擊，並加以分析校園網路中之重要伺服器，偵測網路攻擊，並加以預測。

3. 測試時程、程序

3.1 測試時程

測試時程及查核點為總計畫訂定時間由各子計畫協助完成測試。

時程

1. 各子系統單元測試 (Unit Test) (自 107/02/01 起，應於 107/03/31 完成)
2. 各子系統接受度測試(Acceptance Test) (自 107/04/01 起，應於 107/05/10 完成)

查核點

3. 系統整合與整合測試(107/04/13)
4. 系統測試完成(107/07/1)

3.2 接受測試程序

基於本計畫系統需求規格書內容，本系統須達成以下使用操作場景的需求。各子系統的元件，由各子系統的開發負責人執行，待各子系統整合完成後，由全體人員協同執行整體計畫的整合測試。

總計畫接受測試程序

場景 1：進行網頁弱點掃描及壓力測試

由外部主機對網頁主機進行弱點掃描及壓力測試。使用者利用弱點掃描系統進行主機及 Web Application 之弱點掃描；另於模擬 10 人同時連線之系統效能。如圖五所示。

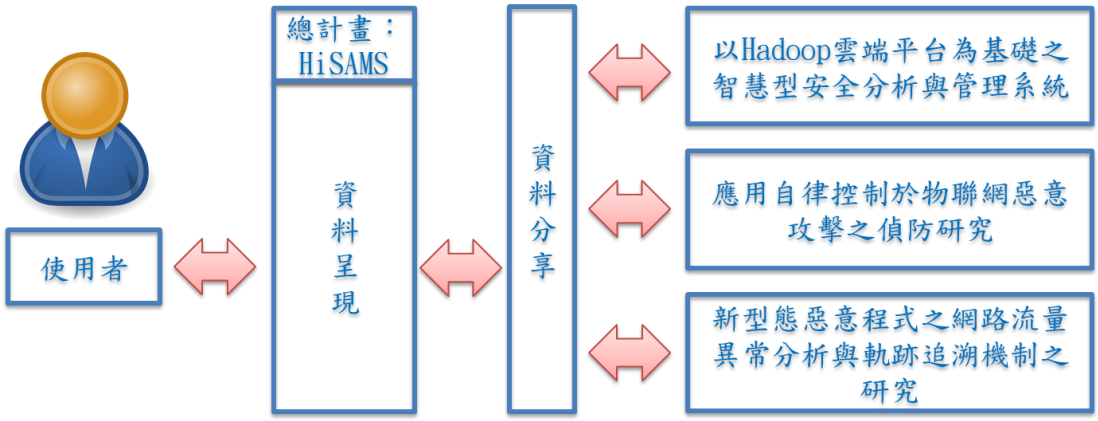


圖5 總計畫測試

子計畫一接受測試程序

場景1：分析電腦中的 PE 檔案

掃描電腦中的執行檔，進行拆解、分析並儲存到系統程式中。如下圖所示：

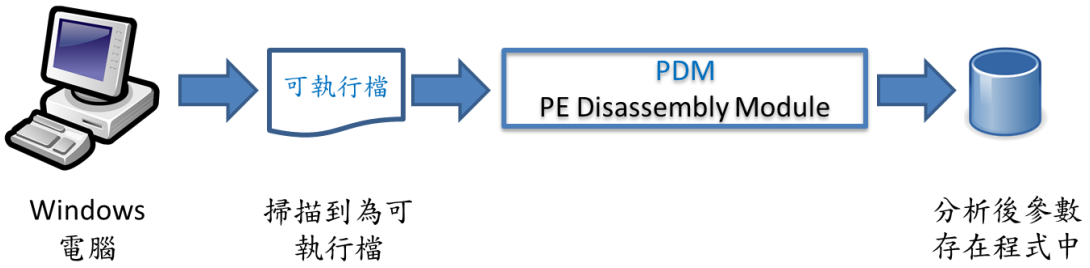


圖6 PE 拆解模組

場景 2：API 距離測量模組

測試 PE 裡每個 API 的距離，如果大於臨界值，則將該 PE 檔放置在暫存資料夾中。如下圖所示：

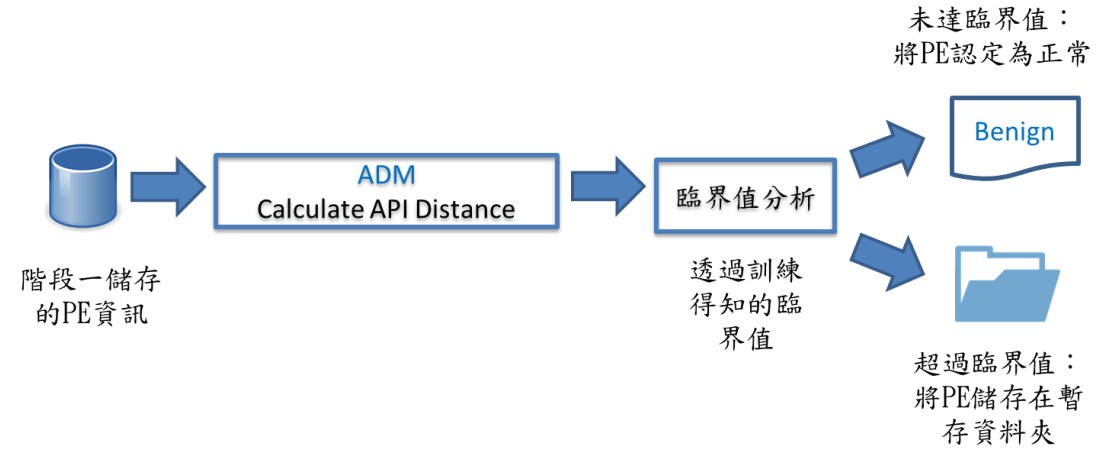


圖7 API 距離分析模組

場景3：RVA Import Table 差異性分析

將 PE 檔案的 RVA 位置匯出，並將該位置的內容轉換成特殊的字串特徵值，藉由分析其相似程度，判斷是否遭到感染。如下圖所示：

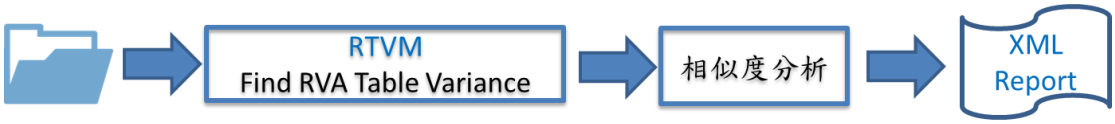
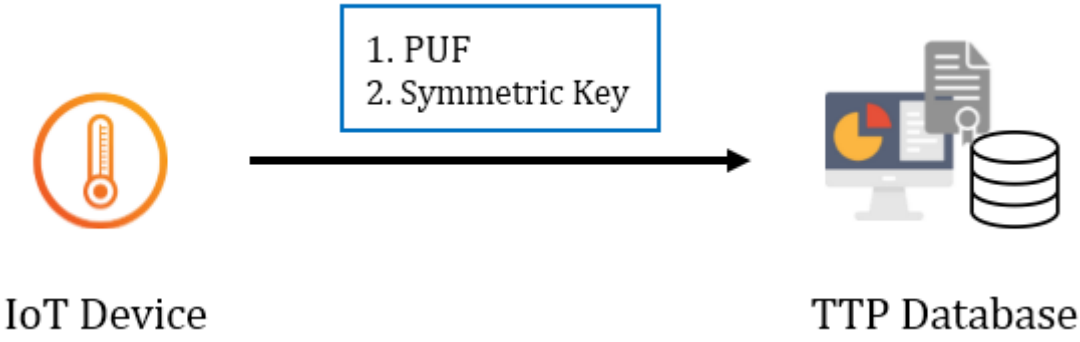


圖8 RVA Import Table 差異性分析

子計畫二接受測試程序

場景 1：儲存物聯網設備 PUF 至第三方資料庫



場景 2：新物聯網設備欲加入物聯網中

閘道對新加入的物聯網設備做驗證，並透過挑戰值判斷設備是否合法的身分:

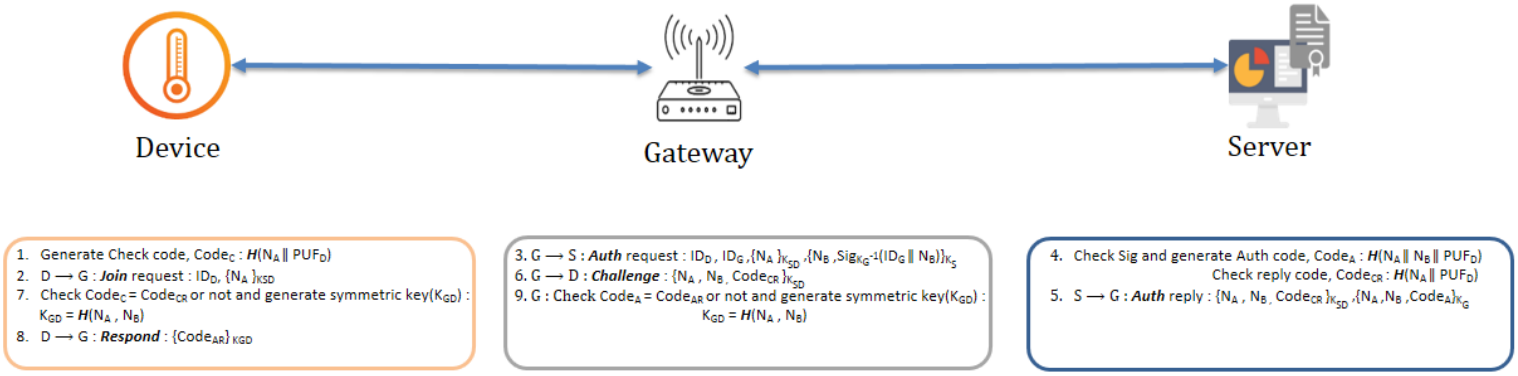


圖9　物聯網設備身分驗證

場景 3:物聯網設備與閘道的加密金鑰更新

閘道與物聯網設備的加密金鑰為一次性，彼此進行更新加密金鑰:

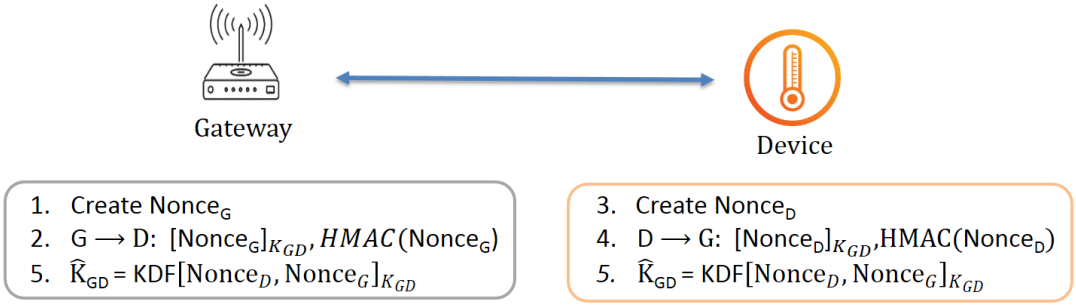


圖10　　物聯網與閘道加密金鑰更新

子計畫三接受測試程序

場景一：收集 Log，並可於 Input Module 中呈現

場景一為收集日誌，於 ELK Server 中透過撰寫日誌檔將收到之日誌存放於 ELK Server 中。

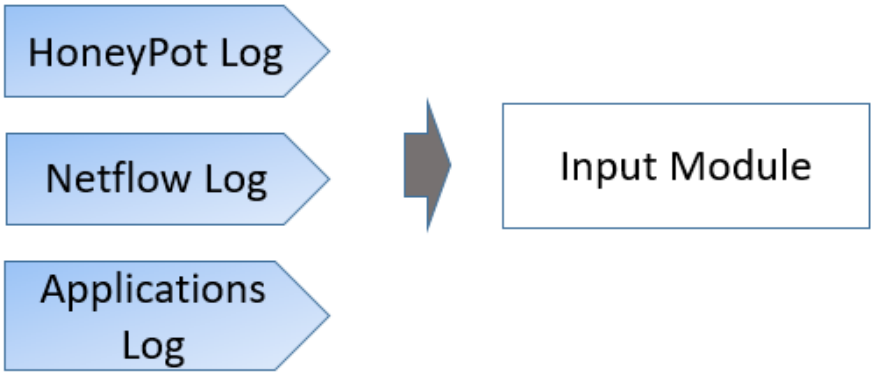


圖11　Input Module

場景二：於 Processing Module 中可呈顯正規化過之資料，並呈現於 Report Module 中

場景二為驗證處理完之資料是否可正常顯示於網頁介面上。

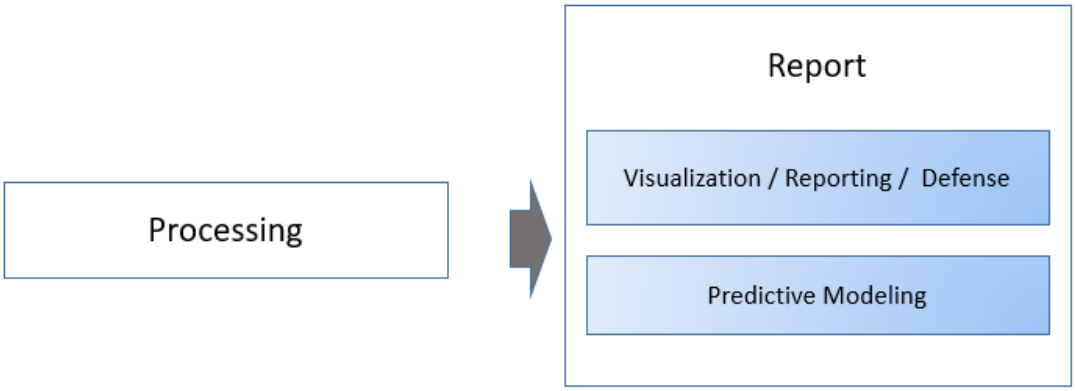


圖12　Processing Module

3.3 整合測試

本計畫為整合型之計畫，各子計畫之開發成果可互享，需於測試中加入整合測試以驗證系統之可行性。要確認每個子計畫之間能互相溝通，並提供完整性檢查，將分析結果正規化表達並匯入到資料庫中。測試之內容為各子計畫間分享資料流及資料呈現。

3.4 壓力測試

計畫為提供網頁服務系統管理員進行資料查詢之作業，由於本系統使用者是系統管理員，所以壓力測試使用者為 10 人，並測試相關錯誤率與回應時間。

4. 測試案例

總計畫之測試案例

HiSAMS-AT-001 Case Test

目的：

- 為了確保各子計畫所上傳之內容已可紀錄於資料庫，互相分享，且不會有 XSS 及 SQL Injection 之情況發生。

✧ 操作說明:

表5 ： HiSAMS-AT-001 Case Test

Identification	HiSAMS-AT-001 Case Test	
Name	正確建立資料庫，各子計畫可分享內容。	
Test Target	資料庫結果正常呈現。	
Requirements	HiSAMS-TR-001、HiSAMS-TR-002	
Severity	1(Critical)	
Instructions	Actor Actions	Instructions
	1. 各子計畫上傳內容及輸入特殊字串，避免安全疑慮	2. 資料顯示於資料庫中
Expected Result	實驗過程中，各子計畫之資料庫皆可正常匯入及分享。	
Cleanup	無	

HiSAMS-AT-002

目的：

- 為了確保同時多人上限可正常呈現,利用測試平台之主機對系統進行多人同時上線之壓力測試。

✧ 操作說明:

表6 HiSAMS-AT-002 Case Test

Identification	HiSAMS-AT-002 Case Test	
Name	主系統壓力測試	
Test Target	於多人連線時可正常回應。	
Requirements	HiSAMS-PR-001	
Severity	1(Critical)	
Instructions	Actor Actions	Instructions
	1. 同時多人連線	2. 網頁與資料庫可正常回應
Expected Result	實驗過程中，網頁與資料庫皆能正常回應。	
Cleanup	無	

子計畫一測試案例

PEI-AT-001 Case Test

目的:

- 為了確保電腦中的 PE 檔案都能夠正確由後續模組分析，所以必須確認拆解後的 PE 訊息符合後續模組的輸入格式

✧ 操作說明:

表7 : PEI-AT-001 Case Test

Identification	PEI-AT-001	
Name	正確拆解 PE 檔案。	
Test Target	確保拆解後的 PE 資訊符合輸入資料	
Requirements	PEI-TR-001、PEI-PR-001	
Severity	1(Critical)	
Instructions	Actor Actions	Instructions
	將 PE 執行檔拆解	正確的建立規格
Expected Result	實驗過程中，本研究正確建立規格，將各式各樣的 PE 檔拆解成後續模組能夠接受的輸入	
Cleanup	無	

PEI-AT-002 Case Test

目的:

- 驗證系統中的 ADM 的有效性，透過異常 PE 與正常 PE 進行比較，查看是否能正常且正確取得 API 距離

✧ 操作說明:

表8 PEI-AT-002 Case Test

Identification	PEI-AT-002	
Name	取得 PE 當中 API 的距離	
Test Target	取得 PE 中 API 的距離，異常與正常的 API 距離有差距	
Requirements	PEI-TR-002、PEI-PR-001	
Severity	1(Critical)	
Instructions	Actor Actions	Instructions
	取得 PE 當中 API 的距離	建立偵測系統的步驟
Expected Result	實驗過程中，本研究將所有 PE 檔案的 API 距離計算出來，並能有效分別正常執行檔以及受到感染的執行檔。	
Cleanup	無	

PEI-AT-003 Case Test

目的:

- 將每個 PE 檔案中的 RVA Import Table 匯出，並取得其 RVA 相似程度。查看正常與異常樣本是否能正確分辨

✧ 操作說明:

表9 PEI-AT-003 Case Test

Identification	PEI-AT-003	
Name	將每個 PE 檔案中的 RVA Import Table 匯出	
Test Target	取得 RVA 相似程度	
Requirements	PEI-TR-001、PEI-PR-001	
Severity	1(Critical)	
Instructions	Actor Actions	Instructions
	將每個 PE 檔案中的 RVA Import Table 匯出	取得 PE 檔案 RVA 相似程度

Expected Result	實驗過程中，將程式的 RVA Import Table 匯出，透過分析其差異性，辨別出是否遭受 PE 感染
Cleanup	無

子計畫二測試案例

LAD-AT-001 Test Case

目的:

- 閘道與物聯網設備雙向身分認證，以提升物聯網的安全性。

表10　LAD-AT-001 Test Case

Identification	LAD-AT-001	
Name	身分驗證機制	
Test Target	閘道及物聯網設備雙向身分驗證	
Requirements	LAD-TR-001~4	
Severity	1(Critical)	
Instructions	IoT devices Actions	Gateway response
	1. 向閘道發送 join request 3. 驗證閘道的合法性	2.驗證 IoT devices 的合法性，決定該設備是否加入物聯網中
Expected Result	閘道和物聯網設備雙向確認身分的合法性	
Cleanup	無	

LAD-AT-002 Test Case

目的:

- 閘道與物聯網設備金鑰的更新，以提高加密金鑰的安全性。

表11　: LAD-AT-002 Test Case

Identification	LAD-AT-002	
Name	加密金鑰更新機制	
Test Target	更新閘道與物聯網設備的加密金鑰	
Requirements	LAD-TR-005~6	
Severity	1(Critical)	
Instructions	Gateway Actions	IoT devices response
	1.向物聯網設備發送亂數值Nonce_G	2.向閘道發送亂數值Nonce_D

	3.產生新的加密金鑰	3.產生新的加密金鑰
Expected Result	閘道與物聯網設備產生新的金鑰	
Cleanup	無	

子計畫三測試案例

MTA-AT-001 Case Test

✧ 操作說明:

表12 : MTA-AT-001 Case Test

Identification	MTA-AT-001	
Name	收集日誌。	
Test Target	是否可正確收集日誌。	
Requirements	MTA-TR-001、MTA-TR-002	
Severity	1(Critical)	
Instructions	Actor Actions	Instructions
	設定接收連接埠號	接收資料
Expected Result	正確接收日誌	
Cleanup	無	

MTA-AT-002 Case Test

✧ 操作說明:

表13 : MTA-AT-002 Case Test

Identification	MTA-AT-002
Name	正規化。
Test Target	日誌經過正規化後，可正常顯示。
Requirements	MTA-TR-001、MTA-TR-002、MTA-TR-003

Severity	1(Critical)	
Instructions	Actor Actions	Instructions
	設定描述檔	正確分割資料
Expected Result	正確顯示接收日誌	
Cleanup	無	

MTA-AT-003 Case Test

✧ 操作說明:

表14 : MTA-AT-003 Case Test

Identification	MTA-AT-001	
Name	顯示攻擊。	
Test Target	是否可依分析之資料顯示攻擊。	
Requirements	MTA-TR-004、MTA-PR-001	
Severity	1(Critical)	
Instructions	Actor Actions	Instructions
	設定攻擊描述	顯示攻擊資料
Expected Result	正確顯示攻擊資料	
Cleanup	無	

5. 測試結果與分析

總計畫

表15　　總計畫測試結果

Test Case #	Results(PASS/FAIL)	Comment
HiSAMS-AT-001	**待測中**	
HiSAMS-AT-002	**待測中**	

子計畫一

表16

Test Case #	Results(PASS/FAIL)	Comment
PEI-AT-001	**PASS**	成功將 PE 拆解
PEI-AT-002	**PASS**	成功算出 API 距離
PEI-AT-003	**PASS**	算出 Import Table 相似度

場景一測試結果：

PDM 能正常地將 PE 拆解，並正常將參數交給 ADM。結果如下圖：

```
C:\Users\Tim\Desktop\PEUPDATE\pefile>python Test.py
('.text\x00\x00\x00', '0x1000', '0x112000', 1120768)
('.data\x00\x00\x00', '0x113000', '0x16000', 69120)
('.tls\x00\x00\x00\x00', '0x129000', '0x1000', 512)
('.rdata\x00\x00', '0x12a000', '0x1000', 512)
('.idata\x00\x00', '0x12b000', '0x4000', 14848)
('.edata\x00\x00', '0x12f000', '0x1a000', 102912)
('.rsrc\x00\x00\x00', '0x149000', '0x24000', 145408)
('.reloc\x00\x00', '0x16d000', '0x14000', 78336)
```

圖13　　拆解 PE

場景二測試結果：

ADM 利用場景一所拆解的 PE，計算出每個拆解 PE 的 API 距離。結果如下圖：

```
[4.0, 8.0, 8.0, 12.0, 16.0, 16.0, 20.0, 20.0, 24.0, 24.0, 28.0, 28.0, 32.0, 32.0
, 36.0, 36.0, 40.0, 40.0, 44.0, 44.0, 48.0, 48.0, 52.0, 52.0, 56.0, 56.0, 60.0,
60.0, 64.0, 68.0, 68.0, 72.0, 72.0, 76.0, 76.0, 80.0, 80.0, 84.0, 84.0, 88.0, 88
.0, 92.0, 96.0, 100.0, 104.0, 108.0, 112.0, 116.0, 120.0, 124.0, 128.0, 132.0, 1
36.0, 140.0, 144.0, 148.0, 152.0, 156.0, 160.0, 164.0, 168.0, 172.0, 176.0, 180.
0, 184.0, 188.0, 192.0, 196.0, 200.0, 204.0, 208.0, 212.0, 216.0, 220.0, 224.0,
228.0, 232.0, 236.0, 240.0, 244.0, 248.0, 252.0, 256.0, 260.0, 264.0, 268.0, 272
.0, 276.0, 280.0, 284.0, 288.0, 292.0, 296.0, 300.0, 304.0, 308.0, 312.0, 316.0,
 320.0, 324.0, 328.0, 332.0, 336.0, 340.0, 344.0, 348.0, 352.0, 356.0, 360.0, 36
4.0, 368.0, 372.0, 376.0, 380.0, 384.0, 388.0, 392.0, 396.0, 400.0, 404.0, 408.0
, 412.0, 420.0, 424.0, 428.0, 432.0, 440.0, 444.0, 448.0, 452.0, 456.0, 460.0, 4
64.0, 468.0, 476.0, 480.0, 484.0, 488.0, 492.0, 496.0, 500.0, 508.0, 512.0, 516.
0, 520.0, 524.0, 528.0, 532.0, 536.0, 540.0, 544.0, 548.0, 552.0, 556.0, 560.0,
564.0, 568.0, 572.0, 576.0, 580.0, 584.0, 588.0, 592.0, 596.0, 600.0, 604.0, 608
.0, 612.0, 616.0, 620.0, 624.0, 628.0, 632.0, 636.0, 640.0, 644.0, 648.0, 652.0,
 656.0, 660.0, 664.0, 668.0, 672.0, 676.0, 680.0, 684.0, 688.0, 692.0, 696.0, 70
0.0, 704.0, 708.0, 712.0, 716.0, 720.0, 724.0, 728.0, 732.0, 736.0, 740.0, 744.0
, 748.0, 752.0, 756.0, 760.0, 764.0, 768.0, 772.0, 776.0, 780.0, 784.0, 788.0, 7
92.0, 796.0, 800.0, 804.0, 808.0, 812.0, 816.0, 820.0, 824.0, 828.0, 832.0, 836.
0, 840.0, 844.0, 848.0, 852.0, 856.0, 860.0, 864.0, 868.0, 872.0, 876.0, 880.0,
884.0, 888.0, 892.0, 896.0, 900.0, 904.0, 908.0, 912.0, 916.0, 920.0, 924.0, 928
.0, 932.0, 936.0, 940.0, 944.0, 948.0, 952.0, 956.0, 960.0, 964.0, 968.0, 972.0,
 976.0, 980.0, 984.0, 988.0, 992.0, 996.0, 1000.0, 1004.0, 1008.0, 1012.0, 1016.
0, 1020.0, 1024.0, 1028.0, 1032.0, 1036.0, 1040.0, 1044.0, 1048.0, 1052.0, 1060.
0, 1064.0, 1068.0, 1072.0, 1076.0, 1080.0, 1084.0, 1088.0, 1092.0, 1096.0, 1100.
0, 1104.0, 1108.0, 1112.0, 1116.0, 1120.0, 1124.0, 1128.0, 1132.0, 1136.0, 1140.
0, 1144.0, 1148.0, 1156.0, 1160.0, 1164.0, 1168.0, 1176.0, 1180.0, 1184.0, 1188.
0, 1192.0, 1196.0, 1204.0, 1208.0, 1212.0, 1216.0, 1220.0, 1224.0, 1228.0, 1232.
0, 1236.0, 1244.0, 1248.0, 1252.0, 1256.0, 1260.0, 1264.0, 1268.0, 1272.0, 1276.
0, 1280.0, 1284.0, 1288.0, 1292.0, 1296.0, 1300.0, 1304.0, 1308.0, 1312.0, 1316.
0, 1320.0, 1324.0, 1328.0, 1332.0, 1336.0, 1340.0, 1344.0, 1348.0, 1352.0, 1356.
0, 1360.0, 1364.0, 1368.0, 1372.0, 1376.0, 1380.0, 1384.0, 1388.0, 1392.0, 1396.
0, 1400.0, 1404.0, 1408.0, 1412.0, 310205.0]

First Step scanning time 7.45599985123
```

圖14　　API 的距離

場景三測試結果：

RTVM 將 PE 的 RVA Import Table 匯出，並轉換成特殊的字串特徵值，之後分析其相似度。

```
Import table starts at file offset 0xab400, and ends at 0xab52c
Features:
84 65 06 00 C8 5B 08 00 00 00 00 00 00 00 00 00
DA 61 08 00 B0 60 06 00 0C 5C 08 00 00 00 00 00
00 00 00 00 5C 62 08 00 F4 60 06 00 80 5C 08 00
00 00 00 00 00 00 00 00 7A 69 08 00 68 61 06 00
00 5F 08 00 00 00 00 00 00 00 00 00 1A 70 08 00
E8 63 06 00 20 5C 08 00 00 00 00 00 00 00 00 00
8A 71 08 00 08 61 06 00 F0 5B 08 00 00 00 00 00
00 00 00 00 FA 71 08 00 D8 60 06 00 18 5B 08 00
00 00 00 00 00 00 00 00 2A 75 08 00 00 60 06 00
```

圖15　　RVA Import Table Dump

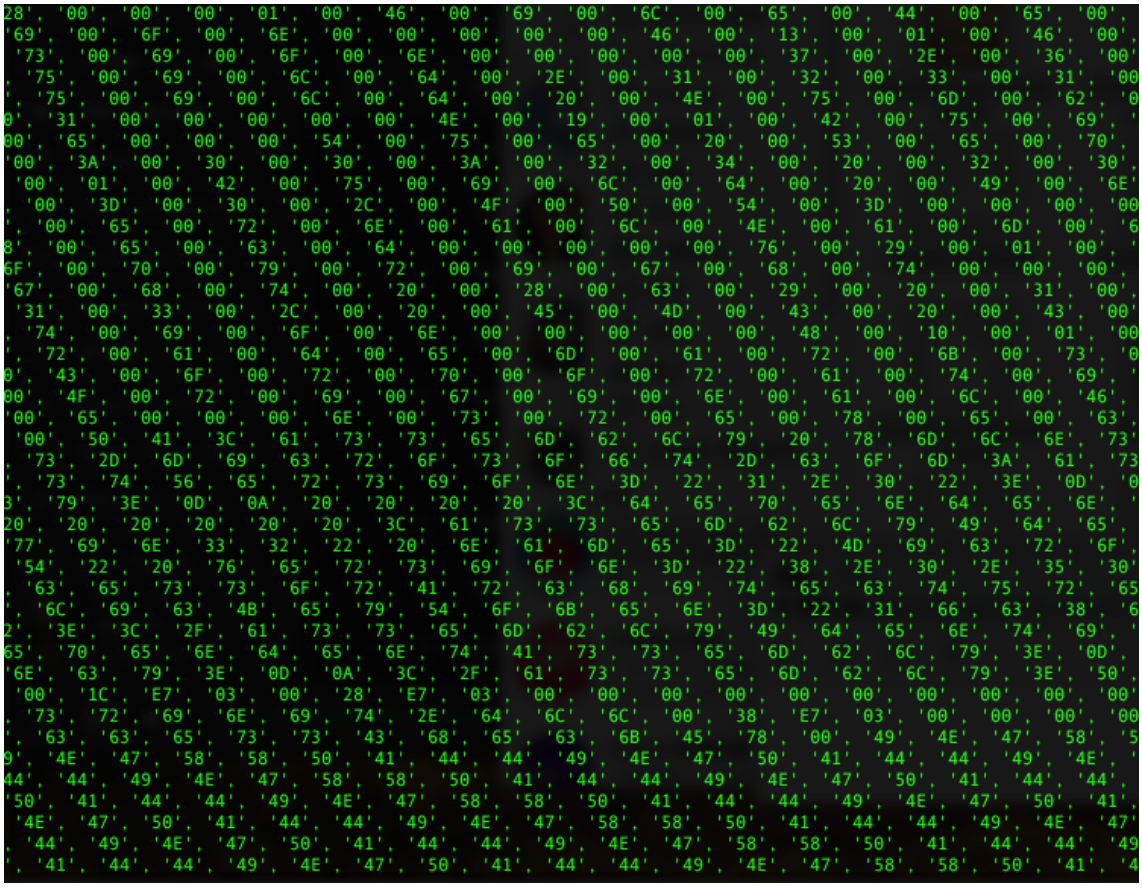


圖16　　轉成特殊字串特徵值

子計畫二

表17

Test Case #	Results(PASS/FAIL)	Comment
LAD-AT-001	**PASS**	
LAD-AT-002	**PASS**	

場景 1 測試結果:

第三方資料庫:物聯網設備資訊

紀錄物聯網設備 device_ID、PUF、key(AC 與物聯網設備的加密金鑰)、hash(製作認證碼使用的 hash function) 、date(紀錄日期)

device_ID	PUF	key	hash	date
nsda00001	1234567890123456	qwerasdftyuighjk	sha256	2018_02_22
nsda00002	zxcvasdfqwer1234	ewjgrnidjwolep	sha256	2018_02_22
nsda00003	edweqwkeidusoekw	sdkoiekdlpfuritj	sha256	2018_02_22
nsda00004	sloeprndjcirktla	vedcjsoieacmdlwd	sha256	2018_02_22

第三方資料庫: Gateway 資訊

紀錄閘道 ID、publickey、date(紀錄時間) 、expiration(到期時間)

ID	publickey	date	expiration
nsda92571	-----BEGIN PUBLIC KEY----- MIGeMA0GCSqGSIb3DQEBAQUAA4GMADCBiAKBgGGWv38EQdoP1tJwn7I11bZqmTPL bqPDDroQQR+3OFRF0mIPpO3IYJR8aiK2lTZz74wws/Q7cbbA1gdetFwOvYFcz18C E9kN5rWqhUE4+Ck9oSP1as5YDo+CMk+jBbPUjA7xalt3wHv2vy1Jt2fjsd3tlzzG t3A+fBtlwNA5Q+gFAgMBAAE= -----END PUBLIC KEY-----	2018_02_22	2020_02_22
ncku116	-----BEGIN PUBLIC KEY----- MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCLVq09Z7Bi7QV3vkCMRU7nZYVJ 6Pfvro785Tf4cXHIgnvn1UjlyJF+FKFBlv+DD/E4j6wKnEWk1/SBXVBLQq4PyU7w q8ksQZWYXttxyuyRmu61b2WsMA8UhObpvMnTjuF7YPUxukA95r7b0nfCmjbEYHV4 Upq59y57kflpmbJPcQIDAQAB -----END PUBLIC KEY-----	2018_02_22	2020_02_22

場景 2 測試結果:

Server:

身分驗證機制 Step 3、Step 6、Step 9

```
========== RESTART: /home/pi/python/Server/Server.py ==========
gateway_ID:
 NSDA92571
<socket.socket fd=7, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 124
2), raddr=('127.0.0.1', 60498)>
 ('127.0.0.1', 60498)
----------step 3----------

device_ID:
 nsda00001
device_nonce:
 cbe60845fa79712d66d85a14fc4b19b397876f5b92c24ee897b9fa3edd0b4aa8
AS public key:
 b'-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCmyCszVDE4+jug1S20wmPf51xa\nurluTF7dTnPt8CW
i/MkFodmBLHgHSgHk/FnPl/6uhu1kcFszvdl3SUGa2OH79wBD\nswmxu5/KrcM0Slgj9zz45rEtwjJlFHgB/gxUYbd3ZbiwikmFaK2DB+t44FFrw
x8N\nvEFdMlh5LWBHkSTY2QIDAQAB\n-----END PUBLIC KEY-----'
gateway_nonce:
 5B[F*J.,SZk2h\88
Signature:
 nsda00001#NSDA92571#cbe60845fa79712d66d85a14fc4b19b397876f5b92c24ee897b9fa3edd0b4aa8#OT86vr8wwIEy0+3I2+FFoM8Eil
KTSVcKpgzss8YYYG44cTgi4VMMYiXf8dVc6UPtqcigR0NDY2juSzDtI939KGFAVb/H4w4cv4h+j1s+IzqQPgUlYLzr6qdSOuVdi/W9o6QHAlx3DE
+aQS8SnX8UHpwCR+l2CGfe5JYb94t7YNQ=#X8ZUJc/hdiUxejqlhI0e3ntHP+2d8nlwav/fVQ9GE2S+zJq/MuxAwEILQM5WeTjls/eDe+JvRISO1
ndJUxr4drTvz9PTA144sXIv7S7FLxnuOU2m04JXcit50FQqvsLbolpAMVUqNd5GmKweqLWAqAjs6B0A/ph2i2C1rr73Isw=
```

```
----------step 6----------

respond to device:
 cbe60845fa79712d66d85a14fc4b19b34a4037ce1714cb6c90307ac9a08c7d5b26beaa736ad67fbec38be3fdf823f419b6392831f3fa8da
bcf58ce3dc0167d542e6b4f72366574261790d156195378859fc0740ee7a3db7138dc298964e93c720e56c48c9988b4a70b4e259051e1303
7
challenge:
 RQIrGgzWrsEc8uC8oX3bKe6jWYshaZ3ptxKRAnGdsRpgDLBOyqwCNgNDTi+aUR/LrAGz6WiRodQxlVjBo/KaVgmPhSWGsCpuIXeK4dC9TkhA1Z6
3rqmIZ7UMj+2gZeCf5GOLUQ8xHmDQkMrOGQwl/FCJdN2CwSlqP/jKIjsdDvo=
nonce_c:
 @fVv%ad@_oIg4[v.
nonce_g:
 5B[F*J.,SZk2h\88
challenge:
 415b8982271447267a88878ae45c8d2628b122913dca2a2ed78937bfb372412b
key_gateway:
 1b295ce1ef53d6f0
----------step 9----------

respond:
 415b8982271447267a88878ae45c8d2628b122913dca2a2ed78937bfb372412b
respond sucessful!!
```

client:

身分驗證機制 Step 1、Step 2、Step 7、Step 8

```
========== RESTART: /home/pi/python/Client/Client.py ==========
device_ID:
 nsda00001
PUF:
 1234567890123456
----------step 1----------

nonce_c:
 @fVv%ad@_oIg4[v.
Challenge:
 557534be5a2381dc5ba01bb0c2c1505da63a030becb8fe0c6b952c9370c3a054
----------step 2----------

message:
 b'nsda00001#cbe60845fa79712d66d85a14fc4b19b397876f5b92c24ee897b9fa3edd0b4aa8'
----------step 7----------

nonce_c:
 @fVv%ad@_oIg4[v.
nonce_g:
 5B[F*J.,SZk2h\88
respond:
 557534be5a2381dc5ba01bb0c2c1505da63a030becb8fe0c6b952c9370c3a054
respond sucessful!!
key_gateway_device:
 1b295ce1ef53d6f0
----------step 8----------

respond to gateway:
 415b8982271447267a88878ae45c8d2628b122913dca2a2ed78937bfb372412b
message:
 b'de8ae8b010b3ead2dfb9e6c4c772acf44c4f9319cbbc07d01e0c08521c0f0febcb0d501216323920efbf5
3f564eb3dbc52f788655c74963e80d963ff3e226cb0c0848cc38419f266ed1cbf390cb83625'
```

AC:

身分驗證機制 Step 4、Step 5

```
<socket.socket fd=8, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('140.116.177.109', 1237), raddr=('140.116.177.117', 34028)>
 ('140.116.177.117', 34028)
b'-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3uBTglgGd0H
hz8g5GTxX0Nz1T\nBu0l+MSz3vvJz1zEokgomxm3m84PC3eiGz2AfBju8yMsPe9IPPVwBpQlVODd65zs
\nEm7USBMcSBnau3sWMspvAPuTu650f/R+xwfL4YIPbojgs0LVokajM0DPUa+/uaLj\nYEyqfkR5sWpl
8uQdvQIDAQAB\n-----END PUBLIC KEY-----'
--------step 4---------
device:
 nsda00001
gateway_ID:
 NSDA92571
device_nonce:
 cbe60845fa79712d66d85a14fc4b19b397876f5b92c24ee897b9fa3edd0b4aa8
gateway_nonce_g:
 5B[F*J.,SZk2h\88
Signature 確認:
 True
device:
 nsda00001
PUF:
 ('1234567890123456',)
device_key
 qwerasdftyuighjk
--------step 5---------
Challenge:
 415b8982271447267a88878ae45c8d2628b122913dca2a2ed78937bfb372412b
Respond:
 557534be5a2381dc5ba01bb0c2c1505da63a030becb8fe0c6b952c9370c3a054
message:
 b'cbe60845fa79712d66d85a14fc4b19b34a4037ce1714cb6c90307ac9a08c7d5b26beaa736ad67
fbec38be3fdf823f419b6392831f3fa8dabcf58ce3dc0167d542e6b4f72366574261790d15619537
8859fc0740ee7a3db7138dc298964e93c720e56c48c9988b4a70b4e259051e13037#RQIrGgzWrsEc
8uC8oX3bKe6jWYshaZ3ptxKRAnGdsRpgDLBOyqwCNgNDTi+aUR/LrAGz6WiRodQxlVjBo/KaVgmPhSWG
sCpuIXeK4dC9TkhA1Z63rqmIZ7UMj+2gZeCf5GOLUQ8xHmDQkMrOGQwl/FCJdN2CwSlqP/jKIjsdDvo=
'
```

場景 3 試結果

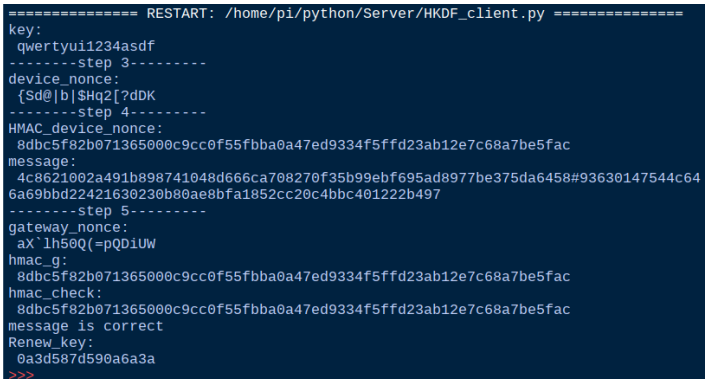
Server:

加密金鑰更新機制 Step 1、Step 2、Step 5

```
=============== RESTART: /home/pi/python/Server/HKDF_server.py ===============
<socket.socket fd=7, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 1240), raddr=('127.0.0.1', 36770)>
 ('127.0.0.1', 36770)
key:
 qwertyui1234asdf
--------step 1---------
gateway_nonce:
 aX`lh50Q(=pQDiUW
--------step 2---------
HMAC_gateway_nonce:
 8dbc5f82b071365000c9cc0f55fbba0a47ed9334f5ffd23ab12e7c68a7be5fac
message:
 cf2fa88c1ce89dc91f816f41ea37f493bd3851b24ca952c21f65814f5a9b8dc6#8dbc5f82b07136
5000c9cc0f55fbba0a47ed9334f5ffd23ab12e7c68a7be5fac
--------step 5---------
device_nonce:
 {Sd@|b|$Hq2[?dDK
hmac_c:
 93630147544c646a69bbd22421630230b80ae8bfa1852cc20c4bbc401222b497
hmac_check:
 93630147544c646a69bbd22421630230b80ae8bfa1852cc20c4bbc401222b497
message is correct
Renew_key:
 0a3d587d590a6a3a
```

Client:

加密金鑰更新機制 Step 3、Step 4、Step 5


========== RESTART: /home/pi/python/Server/HKDF_client.py ==========
key:
qwertyui1234asdf
--------step 3---------
device_nonce:
{Sd@|b|$Hq2[?dDK
--------step 4---------
HMAC_device_nonce:
8dbc5f82b071365000c9cc0f55fbba0a47ed9334f5ffd23ab12e7c68a7be5fac
message:
4c8621002a491b898741048d666ca708270f35b99ebf695ad8977be375da6458#93630147544c64
6a69bbd22421630230b80ae8bfa1852cc20c4bbc401222b497
--------step 5---------
gateway_nonce:
aX`lh50Q(=pQDiUW
hmac_g:
8dbc5f82b071365000c9cc0f55fbba0a47ed9334f5ffd23ab12e7c68a7be5fac
hmac_check:
8dbc5f82b071365000c9cc0f55fbba0a47ed9334f5ffd23ab12e7c68a7be5fac
message is correct
Renew_key:
0a3d587d590a6a3a
>>>

子計畫三

表18

Test Case #	Results(PASS/FAIL)	Comment
MTA-AT-001	**PASS**	如圖所示，可正確接收 Log 資料。
MTA-AT-002	**PASS**	如圖所示，可正確顯示正規化後之資料。
MTA-AT-003	**PASS**	如圖所示，可顯示攻擊者來源國家，並可於 10 秒內回應。

概览 | 索引 | 数据浏览 | 基本查询 [+] | 复合查询 [+]

搜索 logstash-app_log (3538 个文档) 的文档，查询条件：

must | match_all | + | -

搜索 返回格式: Table 显示数量: 250 显示查询语句

查询 5 个分片中用的 5 个，3538 命中，耗时 0.066 秒

_index	_type	_id	_score ▲	event_type	event_time	level	ip	content
logstash-app_log	apache_access_log	AWKoe2M4Ew34PTXO82Ym	1	Scanning(Scanning_Tools)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2NNEw34PTXO82Yn	1	Scanning(Scanning_Tools)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2QAEw34PTXO82Yy	1	Scanning(Scanning_Tools)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2STEw34PTXO82Y_	1	Scanning(Scanning_Tools)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2UDEw34PTXO82ZC	1	Scanning(Scanning_Tools)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2UPEw34PTXO82ZD	1	Scanning(Scanning_Tools)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2YMEw34PTXO82ZE	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2aEEw34PTXO82ZP	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2aPEw34PTXO82ZQ	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:01	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2bUEw34PTXO82ZW	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:02	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2boEw34PTXO82ZY	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:02	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2cTEw34PTXO82Zc	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:02	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2cqEw34PTXO82Ze	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:02	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2cyEw34PTXO82Zf	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:02	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2c7Ew34PTXO82Zg	1	Gaining_access(SQL_Injection)	2018-04-09 11:38:02	6	10.0.2.2	10.0.2.2 - - [
logstash-app_log	apache_access_log	AWKoe2dLEw34PTXO82Zi	1	Compromised(SQL_Injection)	2018-04-09 11:38:02	9	10.0.2.2	10.0.2.2 - - [

圖17　　MTA-AT-001

圖18　　MTA-AT-002

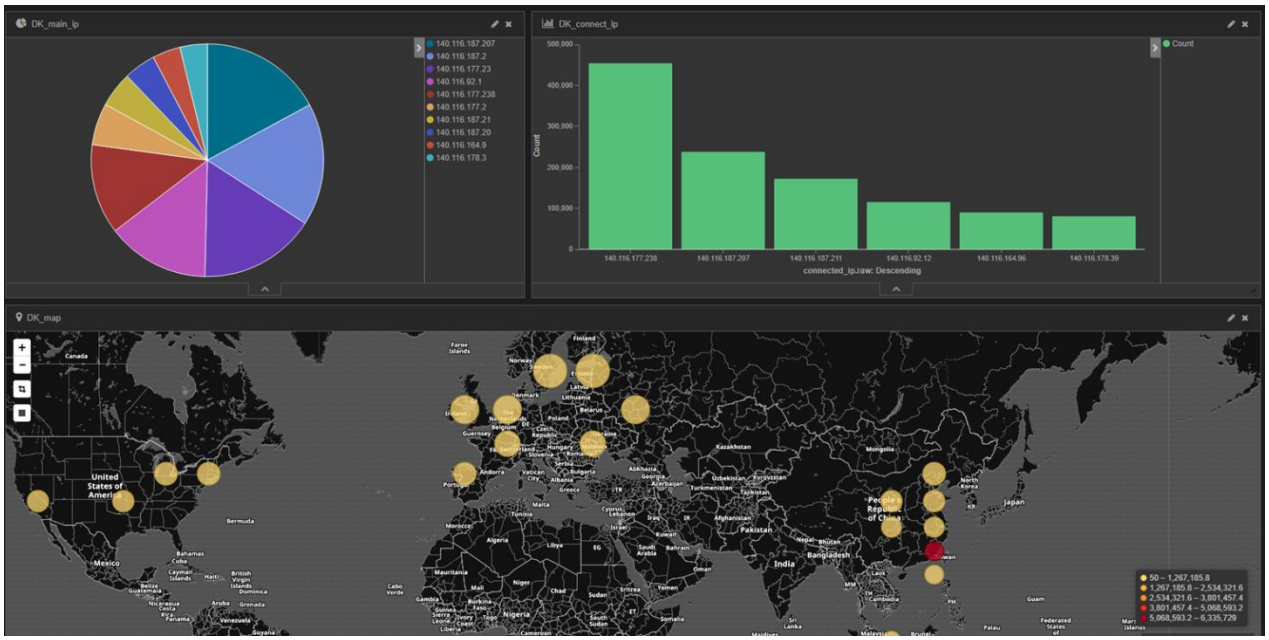


圖19　　MTA-AT-003

附錄 A 追朔表

總計畫之追溯表

表19　總計畫 子系統 vs. 測試案例追溯表

Sub System / Test Cases	PEI	MTA	MTA
HiSAMS-AT-001	V	V	V
HiSAMS-AT-002	V	V	V

子計畫一追溯表

表20　子計畫一 子系統 vs. 測試案例追溯表

Sub System / Test Cases	PDM	ADM	RTVM
PEI-AT-001	V		
PEI-AT-002		V	
PEI-AT-003			V

表21　子計畫一 系統需求 vs. 測試案例追溯表

Sub System / Test Cases	PEI-TR-001	PEI-TR-002	PEI-TR-003
PEI-AT-001	V		
PEI-AT-002		V	
PEI-AT-003			V

子計畫二追溯表

表22　　子計畫二 子系統 vs. 測試案例 追溯表

Sub System / Test Cases	LBA	KGRIS
LAD-AT-001	V	
LAD-AT-002		V

表23　　子計畫二 系統需求 vs. 測試案例 追溯表

Sub System / Test Cases	LAD-TR-001	LAD-TR-002	LAD-TR-003	LAD-TR-004	LAD-TR-005	LAD-TR-006
LAD-AT-001	V	V	V	V		
LAD-AT-002					V	V

子計畫三追溯表

表24　　子計畫三 子系統 vs. 測試案例追溯表

Sub System / Test Cases	Input Module	Processing Module	Report Module
MTA-TR-001	V		
MTA-TR-002	V		
MTA-TR-003		V	
MTA-TR-004			V

表25　　子計畫三 系統需求 vs. 測試案例追溯表

Test Req / Test Cases	MTA-TR-001	MTA-TR-001	MTA-TR-001	MTA-TR-001
MTA-AT-001	V	V		
MTA-AT-002			V	
MTA-AT-003				V