國立台北大學資訊工程學系專題報告

資安事件 3D 動態具象化系統

3D Security Event Animated Visualization System

專題編號: NTPUCSIE-109-009

執行期間：2020 年 09 月 至 2021 年 05 月

1. 摘要

在資訊爆炸的時代，隨著各種駭客攻擊手法愈發成熟且多樣化，越來越多機構開始重視資訊安全之重要性，其中一種防護方式是用安全資訊和事件管理系統來監測異常行為，能夠讓機構擁有一定的資安防護能力。然而，以往的安全資訊和事件管理系統，僅能將資安事件以文字的方式逐行紀錄，資安人員須輔佐相關資訊或程式才能了解事件全貌，在成堆中的資訊中尋找想要的資料，對資安人員是一大考驗。

此專題目標在於，設計出一個以視覺化呈現資安事件的系統，用 FortiSIEM API（下稱 FortiSIEM）擷取必要資料後，透過後端伺服器運算傳至前端，再利用 Three.js 套件，將結果渲染至網頁中，讓使用者可以透過電腦或手機等裝置，以 3D 動畫呈現所選時段內事件如何在各個建築物中互相穿梭、攻擊，以及在系統介面旁顯示事件所記錄的國家資訊，讓使用者迅速了解資安事件的發生場域，而進一步對該異常事件迅速進行處置。

關鍵詞: SIEM、Visualization、Security Event

2. 簡介

目前絕大部分的 SIEM (Security Information and Event Management)系統只能匯出各個 csv 表格報表，或者用圓餅圖、長條圖方式分析各種資安事件，然而市面上並沒有客製化機構地圖的 SIEM，因為需要買方跟賣方配合，需要買方提供 IP 位置跟物理位置的對照表，甚至於需要提供建築物設計圖跟樓層平面圖，以讓系統廠商建模，曠日費時，不符合經濟效益，因此，如何將異常資安事件，以 3D 動畫讓事件以拋物線線條的方式，在建築物之間來回穿梭，再輔以簡易的相關文字資訊，以讓使用者快速理解理解異常事件的流向，即時做出正確決策，是本研究欲達成的目的。

3. 專題進行方式

3.1 系統架構

本系統可以分成兩個部分，第一個部分是使用者介面，第二個部分是後端伺服器，最後一個是資料庫：

3.1.1 前端

前端的部分，除了使用 HTML、CSS 與 Javascipt 基本三元素之外，我們使用 Jquery、SCSS 以加速網頁介面開發，並使用 Bootstrap 4 來做響應式

網頁設計(Responsive Web Design，RWD)，讓系統可以在手機上呈現，網頁的核心部分則用 three.js 這個套件來渲染出整個 3D 畫面，當前端接收到後端伺服器傳來的資料後，便在 client 端做處理，以建構出建築物及事件之拋物線，並 highlight 相對應之建築物。

3.1.2 後端

因為 Python 的便利性，後端伺服器由 Python 中的 Flask 套件來當作資料庫與前端的橋樑。Flask 為 python 所撰寫的微 web 框架，可實踐如路由（Routing）、模板(Templete)、權限(Authorization)等後台的各種簡易操作，並可加入其他元件與 python 共同運作，使其能事件其他功能。於本系統中，Flask 可以接收並解析前端傳過來的指令，再向資料庫調閱資料，並等候資料庫回傳結果，再將該結果經由一系列的資料處理之後回傳給前端渲染。

3.1.3 資料庫

因為 mysql 為 open source，且軟體所需空間小、速度快，網路上也能找到相當多的教學資源，其功能也足夠處理專題，因此選擇 mysql 來當作資料庫，資料來源為 FortiNet 的 SIEM API，使用該公司官網所提供說明修改範例程式，以獲得想要的資料。

3.2 程式實作

3.2.1 前端

前端介面可以分為左控制面板、中間的動畫畫面面板，及右方的控制面板，左方的控制面板可以看到事件的類型及時間，於左下方的面板有個下拉式選單，可以觀看攻擊來源、攻擊者的實際位置（機構內的建築物或某個國家），並可以選擇遭受事件某個建築物，與隸屬於該建築物的 IP 位置，以觀看更詳細的結果。

中間面板為 3D 動畫的渲染位置，將 three.js 的基本物件封裝成我們想要的物件後，等待查詢之後的後端回傳 JSON 資料，再用前端的 Javascript 解析 JSON，抓取裡面每項的參數，統計各類型事件的個數，依照數量及嚴重等級去設定拋物線的顏色及深淺，（紅色代表嚴重異常事件，黃色代表中等異常事件、灰色則代表輕度異常事件，顏色越深代表事件數量越高）並將顏色深淺參數丟給之前封裝好的物件，渲染出事件拋物線；面板中可以透過持續按壓左鍵拖曳滑鼠遊標以改變觀看視角，也可透過滾動滑鼠滾輪以改變動畫視距。

右側面板有「事件列表」、「IP 列表」、「搜尋」三個按鈕，首先透過「搜尋」輸入時間地點等資訊後按下搜尋，前端將會透過 POST 方式，將時間地點等參數傳至後端，並等待後端的資料以 JSON 的資料格式回傳回來，待資料回傳後，即可切換其餘兩個介面——IP 列表介面與事件列表介面，IP 列表介面將顯示所選時段內，風險最高的十組 IP 位置，在條目中右方顯示 IP 本身的風險分數，由高到低排序；在點擊其中一組 IP 位置後，於下方「相關 IP 列表」顯示所有跟該 IP 有關聯的前幾名，點擊之後將會依「所有 IP 的事件」、「該 IP 的關聯事件」輪流播放。

3.2.2 後端

後端以 Python 的 Flask 套件為核心，讓伺服器發揮橋樑的作用，使得前端跟資料庫可以溝通，因此後端的實作分成兩個部分：第一部分為 Flask 如何跟前端溝通，另一個部分為 Flask 如何跟資料庫溝通。

對前端會需要輔以 ssl、httplib2、json 等套件，使伺服器可以將從資料庫回傳的表格資料，經過 ip 對實際位置的轉換之後，將建築物名稱、樓層結果、國家、經緯度資訊、時間、事件相關資訊等結果封裝成 JSON，並丟到前端做運算；而 Flask 與資料庫間溝通則是用 Python 的 mysql.connector 套件登入資料庫並下達 SQL 指令，等待資料庫將表格回傳給後端。


```
▼Array(400)
  ▼[0 … 99]
    ▼0:
        buildingFloor: null
        buildingName: null
        country: "Taiwan"
        destIpAddr: "172.          "
        destbuildingFloor: "1"
        destbuildingName: "GW1040"
        destbuildingTitle: "校外"
        deviceTime: "Wed, 26 May 2021 14:28:35 GMT"
        eventName: "PAN-OS-TRAFFIC-start-allow"
        eventSeverity: 1
        eventSeverityCat: "LOW"
        eventType: "Permitted traffic flow started"
        latitude: "23.7122"
        longitude: "121.56542"
        srcIpAddr: "120.          "
        srcbuildingFloor: 5
        srcbuildingName: "SS"
        srcbuildingTitle: "社會科學學院"
```


（圖一、封裝過後的 JSON 格式）

3.2.3 資料庫

依照 FortiSIEM 的範例程式來修改範例程式，程式碼分成程式本體與掛載的 XML 檔，比如藉由限制獲取資料的 eventseverity 參數大小來獲取特定範圍的資料。

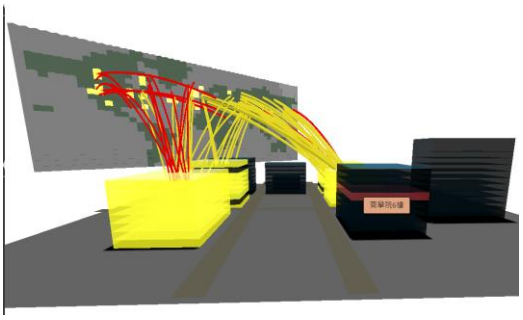
此外，為了加速 SQL 的查詢效率，我們將記憶體的緩衝區變大，使得資料能在記憶體中被處理，並嘗試多種索引（index），使資料可以被快速的從資料庫內取出。


```
<?xml version="1.0" encoding="UTF-8"?>
<Reports>
    <Report baseline="" rsSync="">
        <CustomerScope groupByEachCustomer="false">
        </CustomerScope>
        <SelectClause>
            <AttrList>eventType,deviceTime,durationMSec,eventSeverity,eventSeverityCat,srcIpAddr
        </SelectClause>
        <PatternClause window="500">
            <SubPattern name="Filter_Auth_1">
                <SingleEvtConstr>
                    eventSeverity>=5
                </SingleEvtConstr>
            </SubPattern>
        </PatternClause>
        <SyncOrgs/>
    </Report>
</Reports>
```


（圖二、XML 檔內容）

4. 主要成果與評估

4.1 網頁結果



（圖三、3D 動畫結果）

（圖四、5/25 IP 列表結果）

4.2 五月份之事件統計

由於整體系統於三四月才建制完成，故四月之後的資料比較齊全，因此故抓的五月份來當作統計區間。

4.2.1 五月份各類型事件統計

嚴重程度	事件名稱	count
1	Permitted traffic flow ended	2036519
1	Traffic denied by policy	892082
1	Permitted traffic flow started	188766
9	PAN-OS-THREAT-spyware-85886-deny	37503
5	PAN-IDP-56693	22376
5	Portmapper Info. Retrieving Attempt	17039
9	PAN-OS-THREAT-spyware-86358-deny	4345
7	FTP: login brute force attempt	2641

由圖可知，前三名的事件均為最低嚴重程度的事件，且這三件事件僅跟學校流量政策有所相關，基本上沒什麼風險，接著是高事件的 spyware、中風險的 IDP（Intrusion Detection System）偵測到的事件等等。

4.2.2 五月份各大樓事件數量統計

building	count
電機資訊學院	558128
商學院	537239
行政大樓	437408
法學院	404243
公共事務學院	363360
人文學院	320440
社會科學院	306454
圖書館	145431
田徑場	140304

由圖可知，事件數量跟 IP 使用數量有正相關的趨勢，前三名為電機資訊學院、商學院跟行政大樓，電機資訊學院因為實驗室跟裝置較多、商學院因為系所密度最高(8 層樓供 5 個系使用)、而行政大樓則是因為學校各行政處室運作，因此常有事件產生，總體來說，由於大部分為低中風險事件，如前述跟 Traffic flow 有關的低風險事件，所以能大致判別各大樓的網路使用狀況。

5. 結語與展望

以機構內部為主的 3D 資安事件視覺化系統，由於所需要的資料（機構內部建築物資訊、機構各單位之 IP 位置資訊）過於敏感且瑣碎，且資安公司大多也無法客製化這樣的需求，因此，本專題透過與學校資訊中心的公開資料，以及資訊中心系統組的 FortiSIEM 系統，雖然學校的 IP 位置表因各單位屢次搬遷而參差不齊，但我們嘗試整合上列的現有資訊，轉化成 3D 動畫，並將結果渲染在網頁上。

最終的系統可以實作三大功能，分別是：尋找高風險之 IP、展示資安事件分佈與顯示事件地理位置。首先，找出最高威脅資安事件，和相對應之的 IP，以找出問題源頭，再者，在擁有最高威脅的 IP 中，再找出相對應最高次數 IP，以呈現問題分布範圍，最後，顯示事件視覺化地圖，包含機構內部地圖與世界地圖，以呈現地理關係。

此外，我們成功的將校內主要建物的立體圖建構出來，並將多個異常事件轉化成拋物線呈現在畫面上，使用者可以由拋物線的密集程度，知道異常事件的來源分佈與接收分佈為何，而進一步調整機構內的網路安全政策。

若專題能延續下去，希望能再持續擴充系統的功能、整理後臺的程式碼跟資料庫的底層架構，並著手改善

在運算能力不足的行動裝置上的 3D 影像延遲。

6. 銘謝

由衷感謝教授用心給予指導，並在適時的情況下提供方向或想法，使我們能夠以穩定的速度持續推進專題。

感謝資中系統組，願意不願其煩的解鎖因程式寫爛所帶來的異常流量，進而被封鎖的後端伺服器的 IP。

最後感謝實驗室的學長姐，在我們於較不熟悉的前端開放遇到瓶頸時，給予我們建議與協助。

7. 參考文獻

[1] 童鵬哲、謝宏昀 著，以 ELK 進行視覺化分析資安設備日誌

[2] 連猴子都能懂的 Git 入門指南（ https://backlog.com/git-tutorial/tw/intro/intro2_1.html ）

[3] three.js (https://threejs.org/)

[4] 六角學院 (https://courses.hexschool. com/)