國立台北大學資訊工程學系

專題報告 資安事件圖表分析系統

專題組員:陳致維、張揚鑫、任旻哲

專題編號:PRJ-NTPUCSIE-108-006

執行期間:108 年 09 月至 109 年 05 月

一、摘要

隨著科技的日新月異，網路的蓬勃發展，近年來網路攻擊事件層出不窮，從最傳統的竊取使用者帳號密碼、到 DDoS，再者更到近年頗負盛名的 Wanna Cry 勒索病毒，無非都是現代人和企業所需要防範的攻擊事件，所以現代人會用一些安全資訊和事件管理系統來做網路威脅的管理，進而提升本身的資安防護能力，但網路管理者或相關之研究人員在面對成堆的資訊，肯定是費時且費力。

本專題研究如何從大量的紀錄檔中，對其進行資料庫的整理，並利用網頁的方式，將其以圖表的方式呈現出來。

二、簡介

(一)研究背景

現今資訊安全的議題愈來愈受到重視，駭客也不停的想辦法要做攻擊，面對這些資安攻擊事件，Server 端總會留一些紀錄檔，而記錄檔裡就記載了一些攻擊事件或者警訊的訊息，面對這些龐大的記錄檔數量，用人工方式作分析，肯定會耗費一堆時間成本，弊大於利。

(二)研究目標

本專題運用 SQL Sever 來整合成堆的紀錄檔，運用 ADO.NET 及 ASP.NET 使這些資料給圖表化和統整，讓使用者能更快速地瀏覽各個資料的關聯。

(三)主要預期效益

1. 架設網站使使用者可以更方便使用。

2. 資料圖表化，並把資料統整的更清楚，讓使用者能一瞬間就發現重要細節並找出它與其他資料的關係。

三、專題進行方式

(一)實作平台

(1)前端

前端的部分我們套用了 UiKit、BootStrap、CSS 等等的框架工具，讓我們的網頁畫面更加美觀整齊。

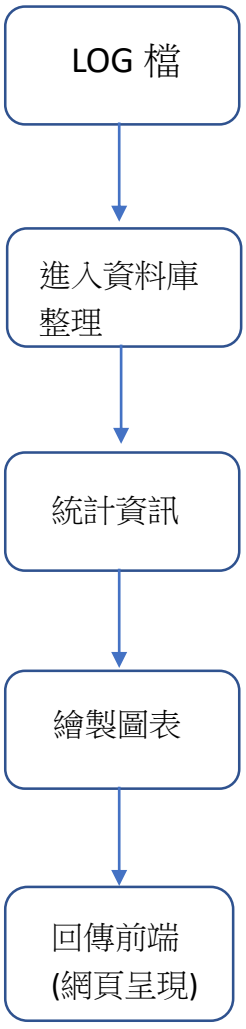
(2)後端及繪圖工具

使用 MS Chart 與 Google Chart 整合資料於圖表上，並用各種不同的圖來呈現，如長條圖、折線圖、圓餅圖等等。

(3)資料庫

使用 Microsoft SQL Server 資料庫處理收到的紀錄檔，並執行 SQL 指令來進行統計。

(二)流程



(三)資料來源

資料的來源來自於國立台北大學的資訊中心，透過 PA、SONIC WALL、FortiSIEM、POP3、IMAP4……等收到的紀錄檔來作為本專題研究之資料。

(四)繪圖工具

(1) Google chart

Google chart 是一款十分強大的資料視覺化 API，不需要安裝任何軟體，只需要透過瀏覽器，便能讓使用者看到各式各樣的圖表。 最常見的使用方式是將 JAVASCRIPT 嵌入網頁：首先會導入 Google chart 所提供的 libraries，接著建立欲視覺化的資料表(可以從 json 檔，Google sheet ，**Database**…etc. 任何你擅長資料擷取的方法匯入)，設定客製化選項，最後便能建立屬於你專屬的可視化圖表。而每一個圖表或是控制項都會有一個自己的 Container（容器）寫在前端 html，讓繪製好的圖表在那個容器呈現，所以可以搭配一些 html 來使用，十分便利。

而除了基本的圖表外，Google chart 也提供了許多和使用者互動的 Events。如我們網頁中的時間軸過濾器(Chart Range Filter)，它能夠讓使用者使用圖表下方的日期過濾器，選取到想要觀看的時間區段，篩選的結果便會立即呈現在上方的圖表控制項中，而不需要再重新建資料生成圖表，如此繁雜的動作。

詳解圖表運作方式: 導入 Google Chart libraries 後，便要來做較為繁瑣的資料擷取部分。
我們使用的方法是利用 asp 的 ADO.NET，先和進行資料庫連線，再執行 SQL QUERY ,最後利用 SqlDataAdapter 將存下的資料放入在 asp 中宣告的 datatable。
但是由於 asp 的 ADO.NET 是在後端進行資料處理，而 Google chart 是在前端(JAVASCRIPT),所以我們必須讓後端的資料能夠傳入前端去做繪圖，因此使用了 **ClientScriptManager 類別**。 這個類別可以將字串傳入前端，故我們利用剛剛存取的 datatable ,先把資料轉化成 Google chart 可以讀取的格式(類似於 json 檔), 再使用 Stringbuilder 把程式碼都寫成字串,最後傳入前端，即完成。剩下的工作便是簡單的繪圖或是根據喜好新增功能。

(2)MS CHART

MS CHART 是微軟在 ASP 內建的圖表控制項,使用前端 ASP 和後製程式碼(後端)來做溝通，後製程式碼常見的有 VB 或 C#,我們在此使用的則是 C#。

同樣地，在繪製圖表前，必須先做資料擷取。這邊也採用 **ADO.NET** 的方式，主要有分成三大物件:

1. Connection:負責建立與資料庫的連結
2. Recordset:負責存取資料庫內容
3. Command:負責下達執行 SQL 指令

順序如下: 先使用 SQLConnection 物件建立連結，創建 SQLCommand 物件，接著用 SQLDataAdapter 物件存取資料庫內容，在用另外宣告的 DataTable 物件做接收，即完成 ADO.NET 的動作。

而在我們某些圖表會使用 SQL 的參數傳遞，來和使用者互動，如我們在 FIREWALL 中的 Msg 種類圖表，便是先讓使用者用日曆選取特定日期區段，將其日期寫入 Textbox(文字輸入框)中，最後使用參數傳遞，將 Textbox 之日期傳入 SQL 指令的參數。雖然這種方法較為瑣碎，但卻能有效地避免 SQL INJECTION 攻擊。若是使用字串組合 SQL 指令的方式，便會有 SQL INJECTION 的風險存在，某些有心的攻擊者便可以透過在 Textbox 中輸入某些特別的字串，便可以任意竄改資料庫、沒帳密就能登入帳號、個人資料外流，等等十分嚴重的資安問題。

資料準備妥當後，利用 Chart 控制項的 Databind method 綁定資料表的某兩個欄位當作是 X,Y 軸，最後再加入 Chart 控制項的眾多屬性，Title、Legend(圖例)，等等客製化選項，依照個人需求去做撰寫。

(五)架設網站

利用家中桌機，架設 IIS 網頁伺服器，於防火牆開啟特定 PORT,並在路由器介面中的 Port Fowarding 設置讓外網連線。

(六)遭遇的困難

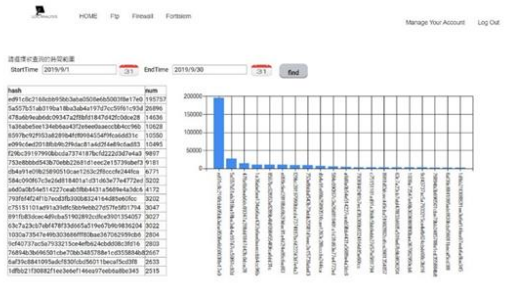
原本呈現資料在網頁上時，常常會遇到 run time error 的訊息，其中便是因為資料量太大了(千萬級別)，雖然可以將 TIMEOUT 強制增加，但這並不是根本的解決之道。因此我們使用了**資料行存放區索引**，這樣便能快速查詢我們要的 query，如此以來便可以解決 loading 過慢的問題。而資料行存放區索引可提供非常高度的資料壓縮，通常是 10 倍，因此可大幅降低資料倉儲儲存體成本，大幅提升效能。

四、主要成果與評估

(一)網站結果

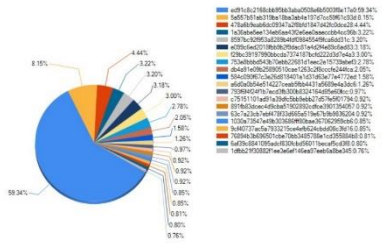
(1)FTP

POP3 及 IMAP4 透過電子郵件的登入情況獲取資訊，我們透過 FTP 的方式取得我們要的紀錄檔。



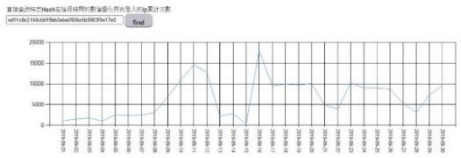
圖一

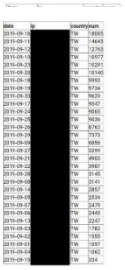
分析特定時間內 HASH 的數量排序如上



圖二

圖一分析之後的圓餅圖，可以看到各個 HASH 之間的比例





圖三

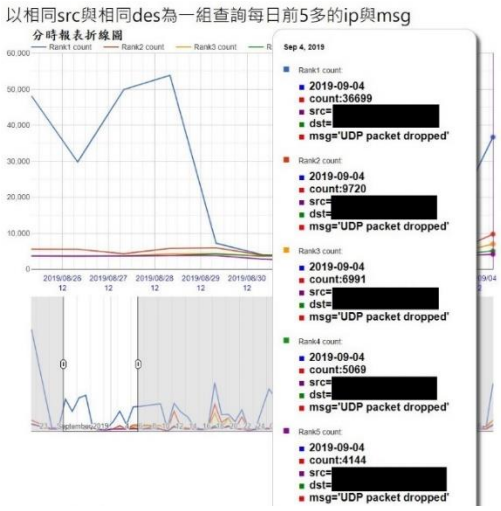
查詢特定 HASH 在特定時間內登入次數變化及登入 IP 之統計

(2)Firewall

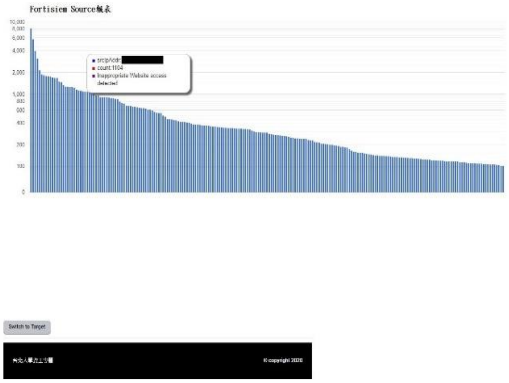
Firewall 是從 PA 及 Sonicwall 獲取資訊，我們透過 Kiwi Syslog 並且經過 Parser 過濾不必要的字元，來取得我們要的紀錄檔。

圖四

相同 source 及相同 destination 為一組做查詢每日前五多的 IP 及 msg

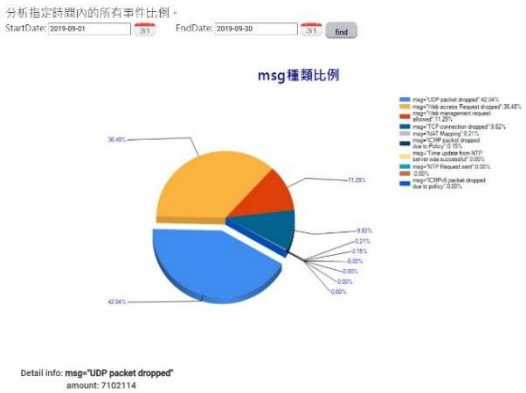


(3)FortiSIEM



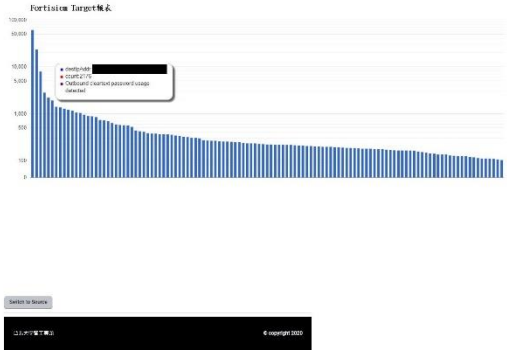
圖六

以 Source 為主，被回報的數量，主要用來防範某個特定的 Source



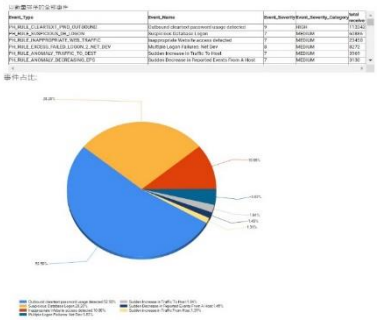
圖五

特定時間內 Firewall 收到所有事件的比例如圖五



圖七

以 Target 為主，被回報的數量，如圖七，主要用來對某個特定的 Target 做防禦的加強



圖八

回報事件的比例如圖八

五、結語

在本次的專題，我們設計出了具人性化的網頁，在人工分析紀錄檔裡的關聯的時間有大大的縮減。我們也用了很多的方式將紀錄檔與紀錄檔之間的關係，呈現給網路管理者或相關之研究人員(使用我們網站的人)看，相信他們能夠做到更好的資安事件分析，進而做到防範的工作。

六、銘謝

特別感謝指導教授在各個方面提供建議與協助。

感謝資訊中心的協助，讓我們有更多的有效資料來源提供我們做研究。

七、參考文獻

[1]
https://www.w3school.com.cn/aspnet/aspnet_intro.asp
[2]
https://developers.google.com/chart
[3]
https://getuikit.com/v2/
[4]
ASP.NET 專題實務 I：C#入門實戰
作者：周棟祥, MIS2000 Lab., 吳進魯
[5]
ASP.NET 專題實務 II：進階範例應用
作者：周棟祥, MIS2000 Lab., 吳進魯
[6]
SQL Server 2017/2016 資料庫設計與開發實務
作者：陳會安
[7]
微軟文本
https://docs.microsoft.com/zh-tw/sql/relational-databases/indexes/columnstore-indexes-overview?view=sql-server-ver15
[7]
Microsoft Windows Server 2008 架站實務
作者： 施威銘研究室著