兩類型入侵偵測攻擊圖與警報關聯配對之研究

王智弘[1]、宋孝謙[1]、邱業宸[1]、楊吉閔[1]、陳彥學[2]
國立嘉義大學資工所[1]、工業技術研究院巨資中心[2]
wangch@mail.ncyu.edu.tw[1]
killo@itri.org.tw[2]

摘要

網路攻擊與軟體漏洞(Vulnerability)近年來快速成長，修補甚至預防系統漏洞的工作愈顯重要，尤其在雲端環境中，系統管理者無法有效地逐一檢視大量虛擬機器，並找出潛在的安全漏洞；因此，自動化安全分析工具成為許多專家學者之研究方向與目標。網路攻擊的產生可能源於系統環境參數設定瑕疵，或提供網路服務之軟體本身存在系統漏洞。

攻擊圖(Attack Graph)即為掃描現有系統架構之設定後，繪製網路攻擊策略圖，協助管理者針對系統弱點進行修補，提高網路安全之可靠性。入侵偵測系統(Intrusion Detection System)為一種檢驗網路封包內容之工具，若封包內容符合攻擊特徵，將產生對應警報並通知系統管理員。

攻擊圖通常與警報關聯有著相互依存的關係。攻擊圖依產生方式概略可分成兩類，本研究擬進行探討與說明。第一類是以系統本身產生的漏洞 (Vulnerability)為繪製攻擊圖的依據。第二類是依入侵偵測系統產生的警報為主體所繪製之攻擊情境 (Attack Scenario) 。本研究在第一類型中運用現有攻擊圖產生工具繪製攻擊圖後，將入侵偵測器產生之警報與攻擊圖配對，當網路攻擊發生時，可藉由警報所對應的節點，得知可能的攻擊過程與進度，達到預測或預防多步驟網路攻擊的效果。警報與攻擊圖配對之結果能用於警報分類，做更進一步之分析應用。而在第二類的研究中則介紹警報關聯之方法概念，利用警報與警報之間的前後關係，分析出攻擊者可能的入侵攻擊策略。

關鍵詞：攻擊圖、入侵偵測、警報配對、警報關聯

壹、介紹

攻擊圖(Attack Graph)為一種利用流程圖繪製因系統漏洞或環境設定所產生之潛在攻擊路徑；由於惡意使用者可能利用這些路徑攻擊，進而危害系統安全，攻擊圖能協助系統管理者在攻擊發生之前，進行相關權限修改或防火牆設定等方式，預防系統漏洞被利用。

入侵偵測系統進行網路封包檢驗之方法，可分為「規則式」(Rule-based)與「異常行為」(Anomaly-based)兩種。檢測異常行為之入侵偵測系統需先採集環境正常運作下之系統資訊，建立「正常狀態」樣板後，才開始偵測異常動作；此方法能利用分類器快速地進行封包篩選，但需要一段時間訓練相關資料庫，且經由機器學習所建立的樣板不一

定相當準確。規則式入侵偵測系統收錄大量攻擊規則模式，一旦網路之行為吻合攻擊模式，入侵偵測器能準確抓出異常封包，因此，若發生不存在數據庫中之攻擊模式，將無法產生警報。

Snort 為規則式入侵偵測系統，因開放原始碼與自帶基本規則數據庫之特性，廣泛運用於入侵偵測相關領域上。現行已有許多套件可加強 Snort 的功能，例如搭配 SnortReport 套件便可方便地運用網頁 GUI 監控警報狀態。

本研究在第一類型方面參考 MulVAL [6]所提出的基本網路架構，在此環境下模擬網路攻擊，並使用 Snort 偵測入侵行為，將產生之警報與 MulVAL 繪製的攻擊圖即時配對，防範多步驟(Multi-Hop)網路攻擊，在攻擊步驟完成前偵測到可疑網路行為。然而目前網路技術的發展日新月異，同時也會產生一些系統中的漏洞，特別是現今雲端環境中伴隨著許多新的漏洞以及未知的惡意攻擊。因此，只利用由系統中已知的漏洞所建立的攻擊圖來處理警報關聯，是無法完整地顯示當前系統的安全狀況。因此在第二類型的研究中，我們介紹了一種使用貝式網路模型建構可調式特徵權重之警報關聯系統 [10]。該系統可以由網路流量檢測到惡意的行為，利用警報與警報間之前後關係，分析出攻擊者入侵的攻擊策略，並呈現當前網絡安全狀況給管理員。

貳、相關研究

攻擊圖為一種描述網路攻擊策略之方式，在入侵偵測及網路安全中扮演重要角色。產生攻擊圖之方式大致可分為兩種，較常見者為利用入侵偵測系統之警報進行警報關連(Alert Correlation)後，依據彼此之相關性連結而成，如 [2][9][11]所提到之方式即為此種分類，又可稱為攻擊策略 (Attack Scenario)；另一種方式為分析系統之設定與潛在之漏洞，與漏洞資料庫(如：CVE、NVD…等)比對，此類方法可能根據漏洞產生之前因後果繪製攻擊圖，或者將系統狀態窮舉配對產生 [12]。

早期 Michael L. Artz 便提出繪製網路攻擊圖的概念，發表 Topological Vulnerability Analysis (TVA) [4]工具以繪製攻擊圖，藉由 Nessus 進行系統漏洞掃描後，將產生的系統描述檔案與漏洞資料庫進行交叉比對；Michael L. Artz 接著提出的 NetSPA [2] 網路安全分析工具，補足 TVA 無法讓使用者自定義規則的功能，但上述兩種方法皆需與漏洞資料庫進行交叉比對，且系統狀態改變的不確定性導致運算時間複雜度遽增；進行這類條件式比對時，容易使判斷條件進入無限迴圈，而無法有效應用於大規模的系統架構。

Xinming Ou 等人提出邏輯式的網路安全分析工具—MulVAL [6]，選擇 Open Vulnerability and Assessment Language (OVAL) 作為分析系統資訊之工具；爾後將 MulVAL 所提出之架構實作於 [5] 中，不同於以往，MulVAL 使用 XSB 邏輯運算系統 [7] 進行 CVE 漏洞資料庫的資訊比對；防止無限迴圈產生即為 XSB 系統特色，因此 MulVAL 成功地被運用在大規模系統架構中。本研究選用 MulVAL 做為攻擊圖產生工具。

除分析研究外，攻擊圖可作為警報關聯之依據，加速相關過程進行，並運用重新關聯過後之警報更新現有攻擊圖。Seyed 等人提出一種混合式架構，利用攻擊圖進行警報

關聯後，若存在未對應之警報(未知警報)，待累積至一定數量後將移動至模組二進行警報關聯，並將其結果用於更新攻擊圖 [3]。

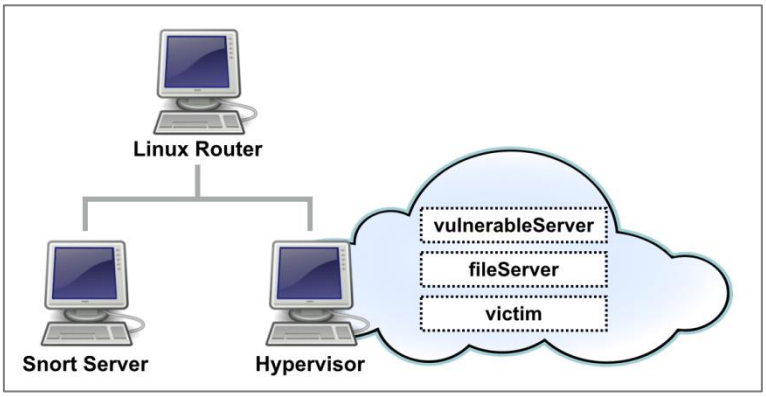
本研究第一種類型即介紹分析漏洞因果關係繪製之攻擊圖，避免窮舉法可能造成時間複雜度過高等問題，而攻擊圖之應用主要為即時配對入侵偵測系統發出之警報，提供視覺化的操作介面，並能將結果應用於警報關連中。第二類型則偏重警報關聯之利用。此類型方式主要單就系統產生的警報之間的連結性來建構攻擊者的攻擊情境與預測其攻擊意圖。相關的研究如由 Zhu 與 Ghorbani [13]提出了通過多層感知器(MLP)和支持向量機（SVM）來計算任意兩個警報之間的關聯性之概率，並進一步提出了警報關聯矩陣（Alert Correlation Matrix, ACM）的技術來用於儲存任何兩種類型警報的過去關聯經驗。Ren 等人 [8] 提出基於貝氏網絡來估計任何兩種警報類型之間的因果關係的警報關聯方法。離線的貝氏關聯特徵選擇系統來選擇兩種警報類型之間的關聯特性，並儲存此信息。在線警報關聯系統用來識別警報的因果關係，並建構即時攻擊情境。

參、研究方法

3.1 第一類型：分析系統漏洞所產生之攻擊圖

本類型之研究實驗對象架設於雲端環境中，方便還原至攻擊前之狀態；雲端中網路架構採用橋接方式，故與雲端外部機器屬於同一區域網路。網路架構使用 Linux Server 作為路由器處理封包，另外採用一台 Server 做為 Snort 封包監控專用。

雲端環境架構參考 MulVAL 之簡單網路架構，分別建置存在漏洞之伺服器(vulnerableServer)、檔案伺服器(fileServer)、以及攻擊對象(victim)，攻擊對象之網路僅限區網連接，由外部網路無法直接連線，此架構為假設 Multi-hop 網路攻擊進程，攻陷提供服務之外部伺服器後，藉由內部區域網路之信任關係，進而攻擊與外界隔絕的內部主機。圖一為實驗網路架構圖，各伺服器之代號與圖二對應。



圖一：實驗網路架構

防火牆設定區域網路為完全信任，僅開啟對外網路服務之連接埠口，檔案伺服器(fileserver) 於區網中提供 NFS 服務。檔案伺服器(fileServer)與攻擊目標 (victim) 使用

Linux Ubuntu 14.04 建置，Hypervisor 使用 XenServer 6.0.2，搭配 CloudStack 4.4.0 作為雲端平台管理介面。

存在漏洞的主機(vulnerableServer)上所建置之漏洞為 IIS 4.0 至 5.0 中存在的安全性漏洞(CVE-2000-0884)，使遠端攻擊者能夠讀取網頁根目錄資料夾外的文件，並可能執行任意的程式碼。攻擊圖產生工具 MulVAL 能根據參數的調整，生成多種不同型態之攻擊圖，主要分為 dot 檔與文字描述兩種方式；dot 檔藉由 GraphViz 套件可轉為流程圖片，考量分析方便性，本實驗採用文字描述版本之攻擊圖。圖二為 MulVAL 產生之攻擊圖文字檔，用文字描述表示目前系統之可能攻擊路徑。讀入此描述檔後，將產生動態攻擊圖，當入侵偵測警報成功配對，動態攻擊圖會以動畫方式呈現，方便管理員掌握資訊。

MulVAL 在讀入主機架構與狀態描述檔後，經過邏輯分析可以產生純文字描述攻擊圖，或者用 GraphViz 套件繪製之 dot 流程圖，由於攻擊圖配對時需要節點之相關資訊，若使用 MulVAL 直接產生之 dot 檔將增加讀取難度，檔案中參數變化複雜，格式不固定，故選擇產生文字敘述檔後，再由附加程式讀入節點資訊產生 dot 檔。

```
1  <1>|--execCode(victim,root)
2     (2) RULE 4 : Trojan horse installation
3        <3>|--accessFile(victim,write,'/share')
4           (4) RULE 16 : NFS semantics
5              <5>|--accessFile(fileServer,write,'/share')
6                 (6) RULE 17 : NFS shell
7                    [7]-hacl(vulnerableSever,fileServer,nfsProtocol,nfsPort)
8                    [8]-nfsExportInfo(fileServer,'/share',write,vulnerableSever)
9                    <9>|--execCode(vulnerableSever,iis5)
10                      (10) RULE 2 : remote exploit of a server program
11                         <11>|--netAccess(vulnerableSever,tcp,80)
12                            (12) RULE 6 : direct network access
13                               [13]-hacl(internet,vulnerableSever,tcp,80)
14                               [14]-attackerLocated(internet)
15                         [15]-networkServiceInfo(vulnerableSever,iis,tcp,80,iis5)
16                         [16]-vulExists(vulnerableSever,'CVE-2000-0884',iis,remoteExploit,privEscalation)
17          [17]-nfsMounted(victim,'/share',fileServer,'/share',read)
```

圖二：攻擊圖文字檔

上圖即為 MulVAL 所產生之純文字描述攻擊圖，其中節點之縮排程度代表不同階層之節點，每行開頭號碼代表該節點之識別號碼(ID)；若識別號碼為符號”<ID>” 表示該節點為「導致結果」(Derivation Node)，使用符號”(ID)”表示「造成原因」(Derived Fact Node)，”[ID]”表示「原始原因」(Primitive Fact Node)。

繪製攻擊圖時需建立各節點之連結關係，依據各行節點描述之縮排程度，給予不同排名(Rank)，例如：「節點<1>」之排名為 0，「節點(2)」之排名為 1；相同程度之縮排給予相同 Rank 值。給定排名後，須將各節點依照 Rank 值以反序方式排序，排名數較高之節點代表位於攻擊圖尾端。

節點代號後順位第一之字串為該節點之屬性，如第 1 行之 execCode(victim,root)代表在 victim 中以 root 身分執行任意程式碼(Arbitrary Code)，第 2 行之 RULE 對應方式由 MulVAL 而來，第 3 行表示存取 victim 主機上的檔案資料，第 7 行之 hacl 代表網路連線

狀況，第 8 行為 fileServer 主機所提供之 NFS 服務相關資訊描述，vulnerableServer 與 fileServer 在 NFS 協議下以 nfsPort 連結。

圖三為讀取攻擊圖文字敘述檔之演算法，第 7 行將所有相同排名之節點放入暫存器中，第 8 行代表若所有相同排名之節點已處理完成，則往下一排名進行，若該節點存在，將暫存器中之所有節點與該點連結；節點型別(Type)分別以 0 代表原始原因、1 代表造成原因、2 代表攻擊結果，由於「原始原因」之節點必為子節點，無法成為母節點，第 8 行之條件限制即為避免「原始原因」成為母節點；第 16 至 20 行處理跨排名「原始原因」節點之連結狀況。

```
1   Read Attack Graph
2       sortRank(); //Z->A
3       rank = rank of first node
4       missed = -1 //initialize
5       Stack<Node> temp;
6       for each Node(i){
7           if Node(i).rank>=rank then push Node(i) into temp
8           else if Node(i).type>0
9               rank--;
10              while(!temp.empty()) Node(i) link with temp.pop()
11              if missed>=0
12                  i = missed-1;
13                  missed = -1;
14              else push Node(i) into temp
15          else
16              missed = i;
17              while Node(++i)!=null
18                  if(Node(i).type>0) break;
19              i--;
```

圖三：讀取攻擊圖文字敘述檔演算法

完成攻擊圖繪製後，便可開啟 Snort 進行入侵偵測，經由 barnyard2 套件將 snort 記錄寫入資料庫後，可利用 SnortReport、BASE 等網頁 GUI 呈現，本研究採用 SnortReaport 做為介面工具。由於攻擊圖中許多節點為正常網路存取之動作，為偵測區網中的存取動作，制定以下規則於自定義規則中，方便實驗進行。

```
## dectect web access
alert tcp any any -> vulnerableServer servicePort(\
    flags: S; ack: 0; sid:930801; rev:1;)
## dectect nfs write
alert udp vulnerableServer any -> fileserver any(\
    content:"|00 00 00 00|"; offset:4; depth:4;\
    content:"|B7 73|"; offset:112; depth:2;\
    sid:930802; rev:1;)
```

偵測網路存取動作的規則中，將 Flags 值設定為 SYN 可捕捉到 TCP 連線時初始化的三個封包，利用 ACK 碼抓取第一個封包當作網路存取動作的開始，因此每次的網頁存取動作能降低至一次警報提醒。

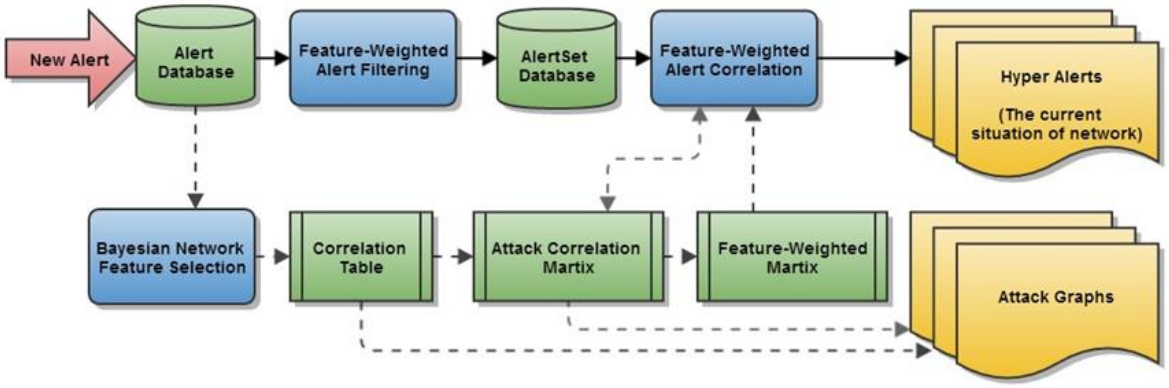
NFS 存取動作偵測方面，僅監聽由 vulnerableServer 發出的封包，由於存在漏洞的主機使用 Windows 2000 Server 作業系統，需透過 Windows Service for UNIX 掛載以 Linux 為基礎的檔案伺服器資料夾。Windows 與 NFS 溝通的套件使用 UDP 封包與檔案伺服器溝通，配合封包表頭資訊精簡的 UDP 封包，鎖定封包為 RPC request 才做處理動作(offset:4~7)，當主機端發出寫入要求時，於 offseet:112 開始會以"B7 73"標記，故在偵測 NFS 存取動作的規則上加上此限制，可排除其它資訊交換之封包，將每次的 NFS 寫入動作降低至一次警報提醒。由於 vulnerableServer 所運行的服務中，並無主動產生 NFS 寫入動作之程序，因此當 Snort 偵測到寫入動作時，在此環境下即為漏洞已被成功利用，並安裝木馬程式。

攻擊圖之所有參數將存在攻擊圖配對程式中，當 MySQL 中警報資料庫產生改變，配對程式能掃描到新增節點，並依據節點所對應之 IP 地址、Port 位置、警報 Signature ID 等屬性對應至攻擊圖上。

3.2 第二類型：依警報關聯為主體所產生之攻擊情境

利用系統環境與本身存在的漏洞所建造的攻擊圖，對於攻擊者使用未知的漏洞所作的未知攻擊，達到的警報關聯成效有限，所以本研究將介紹第二類型的警報關聯方法。

本節以使用貝式網路模型建構可調式特徵權重之警報關聯系統為例，達到警報關聯的過程包括至少三個步驟，即前處理，警報關聯和後處理。前處理的目的是降低警報的數量和刪除誤報，以提高警報關聯的效率和準確性；而警報關聯則是找出警報之間的關聯性，使我們可以從這些警報獲得一些資訊；在後處理步驟中，我們利用從警報關聯過程中所得到的資訊進行更進一步的行動，例如，找出警報的嚴重程度，給予他們優先權、或分析警報之間的因果關係，並了解入侵者的攻擊策略。我們所提出的第二種警報關聯方法的架構於圖 1 中，我們將分別描述架構中的各項組件。



圖四：使用貝式網路模型建構可調式特徵權重之警報關聯系統之架構圖

在我們的架構中，依照組件的類型分為關聯處理、關聯性記錄表及資料庫、結果圖。

關聯處理：

- Bayesian Network Feature Selection：特徵(Feature)為兩個警報間之關係，本篇所使用的特徵如表 1 所示，對於警報(a,b)，a 代表過去的警報，b 代表新的警報。透過觀察過去的警報及計算一個警報在另一警報之前一個的發生機率，找出入侵偵測器所產生的警報與警報之間的關係。對於此作法貝氏網路是個很好的選擇，我們不僅將貝氏網路用來計算發生的機率，還用來找出警報特徵與警報特徵之間的關聯強度，其關聯程度被記錄在 Correlation Table 中。

表一：使用警報的特徵與其定義

特 徵	定　義
F_1	F_1代表 a 的來源 IP 位址與 b 的來源 IP 位址之相似度。 其值為 0 到 1 之間。
F_2	F_2代表 a 的目標 IP 位址與 b 的目標 IP 位址之相似度。 其值為 0 到 1 之間。
F_3	F_3代表 a 與 b 之間目標 Port 是否相同的關係。 其值為 0 或 1。
F_4	F_4代表 a 的目標 IP 位址與 b 的來源 IP 位址相同， 並且 a 的來源 IP 位址與 b 的目標 IP 位址不相同。 其值為 0 或 1。
F_5	F_5代表 a 的目標 IP 位址與 b 的來源 IP 位址相同， 並且 a 的來源 IP 位址與 b 的目標 IP 位址相同。 其值為 0 或 1。
F_6	F_6代表 a 的警報類型與 b 的警報類型間的逆向關聯強度。 逆向關聯強度被記錄在 alert correlation matrix。 其值為 0 到 1 之間。

F_7	F_7代表 a 的警報類型與 b 的警報類型間的關聯強度。 其關聯強度被記錄在 alert correlation matrix。 其值為 0 到 1 之間。

- Feature-Weighted Alert Filtering：在此步驟中，過濾器會過濾掉重複或是已知誤報的警報。Feature Weight Matrix 提供了過濾器所需的資訊。
- Feature-Weighted Alert Correlation：對於各警報類型配對間存在著獨特的關係性。因此，我們透過警報的類型以及類型與類型之間的關係性達到關連警報的目的。

關聯性記錄表及資料庫：

- Alert Database：當入侵偵測器偵測到系統網路中可疑的封包，產生初步警報並送入警報資料庫中。
- Correlation Table：各個警報類型之間的關聯性被記錄在 Correlation Table，我們可以知道由一個警報類型發生到下一個警報類型中，警報類型間之特徵關聯程度。
- Feature Weight Matrix：當我們獲得警報類型之間的特徵關聯性，可將其資訊用在警報關聯，Feature Weight Matrix 用來選擇警報關聯的警報特徵。
- Alert Correlation Matrix：關連後的結果將會被儲存到 Alert Correlation Matrix (ACM). 根據 ACM，我們可以了解警報類型關聯後的關係性。

結果圖：

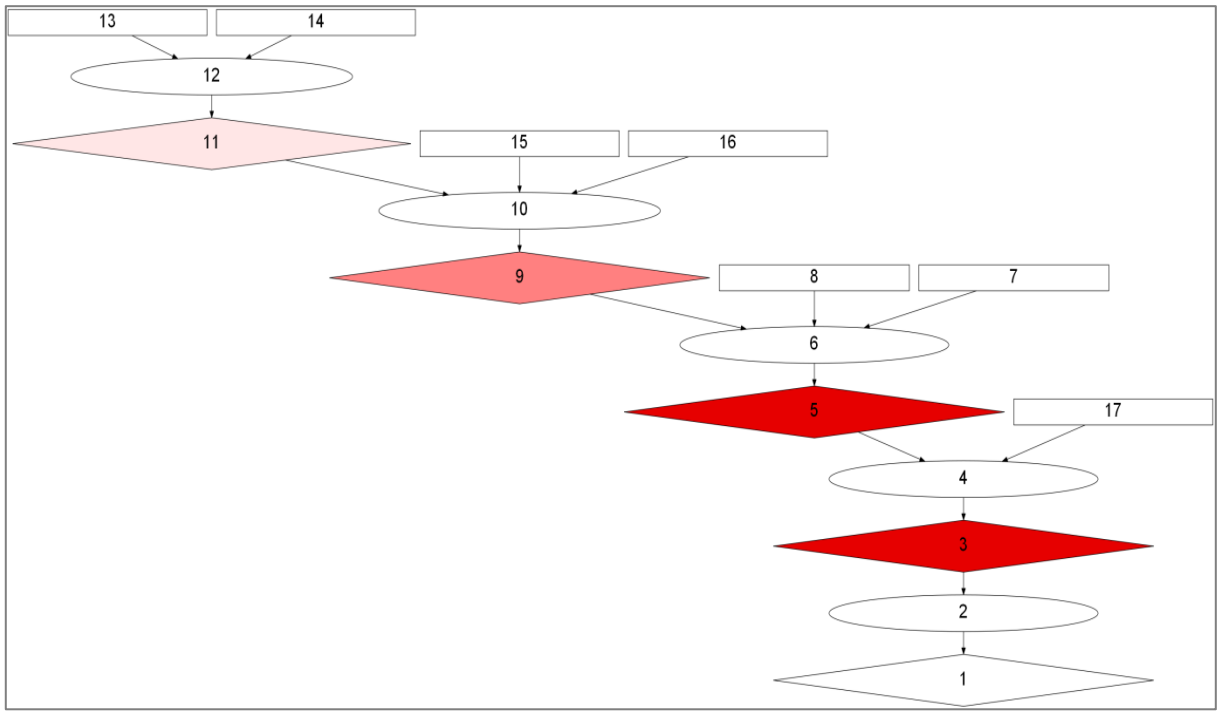
- Hyper Alerts：在執行完警報關聯程序後，我們獲得到當前的網路安全狀態。Hyper-Alert 即是由關聯相關的警報所得到的網路安全狀態。
- Attack Graph：我們由 Correlation Table 以及 Alert Correlation Matrix 建構出攻擊圖 (attack graph)，Correlation Table 構成攻擊圖的資訊表示在最近的一段時間內，入侵者的攻擊策略；ACM 構建攻擊圖的資訊表示過去的攻擊策略。

肆、實驗結果

本節將上述介紹的兩種攻擊圖產生模式以及警報之關聯情形以實驗的結果呈現，讀者可以更加了解攻擊圖之模式並進一步比較其異同之處。

4.1 第一類型：攻擊圖與警報配對

圖五為我們在 MulVAL 繪製攻擊圖上進行警報與攻擊圖配對之結果，若節點為原始原因(Primitive Fact Node)，以方形表示，造成原因(Derived Fact Node)以橢圓形表示之，若該節點為導致結果(Derived Fact Node)則以菱形表示之。警報成功配對時，攻擊圖繪製工具將改變其節點顏色為 HSV(0,1,1)並隨時間淡化，已對應過之節點不會淡化還原成白色，方便系統管理者比較警報發生時間與攻擊進程。



圖五：警報與攻擊圖配對結果

上圖各節點之內容依據 ID 號碼對應到表二，當存在漏洞之網路服務被存取時(ID:11)，利用 Snort 分析封包表頭檔之功能，將封包 flags 設為 S (SYN)，且 ACK 值為 0，限制第一個 TCP 連線三方交握之封包才紀錄存取動作，避免警報數量過多。

由於 NFS 服務可能隨機挑選 Port 位址，以 Linux 為例，需修改相關 RPC 設定，限制服務提供範圍，方便偵測進行。圖四中若攻擊者利用具權限提升或可執行任意程式碼之漏洞(ID:9)，則可藉由 NFS 服務安裝木馬程式(ID:5)，利用其分享檔案之特性達到植入木馬於 victim 主機的效果(ID:3)。

表二：攻擊圖 ID 與敘述對照表

ID	Description
1	execCode (***victim***, root)
2	RULE 4 (Trojan horse installation)
3	accessFile (***victim***, write, '/share')
4	RULE 16 (NFS semantics)
5	accessFile (***fileServer***, write, '/share')
6	RULE 17 (NFS shell)
7	hacl (***vulnerableSever***, ***fileServer***, nfsProtocol, nfsPort)
8	nfsExportInfo (***fileServer***, '/share', write, ***vulnerableSever***)
9	execCode (***vulnerableSever***, apache)
10	RULE 2 (remote exploit of a server program)
11	netAccess (***vulnerableSever***, tcp, 80)
12	RULE 6 (direct network access)
13	hac l (internet,***vulnerableSever***, tcp, 80)
14	attackerLocated (internet)
15	networkServiceInfo (***vulnerableSever***, IIS, tcp, 80, IIS5.0)
16	vulExists (***vulnerableSever***, 'CVE-2000-0884', httpd,remoteExploit, privEscalation)
17	nfsMounted (***victim***, '/share', ***fileServer***, '/share', read)

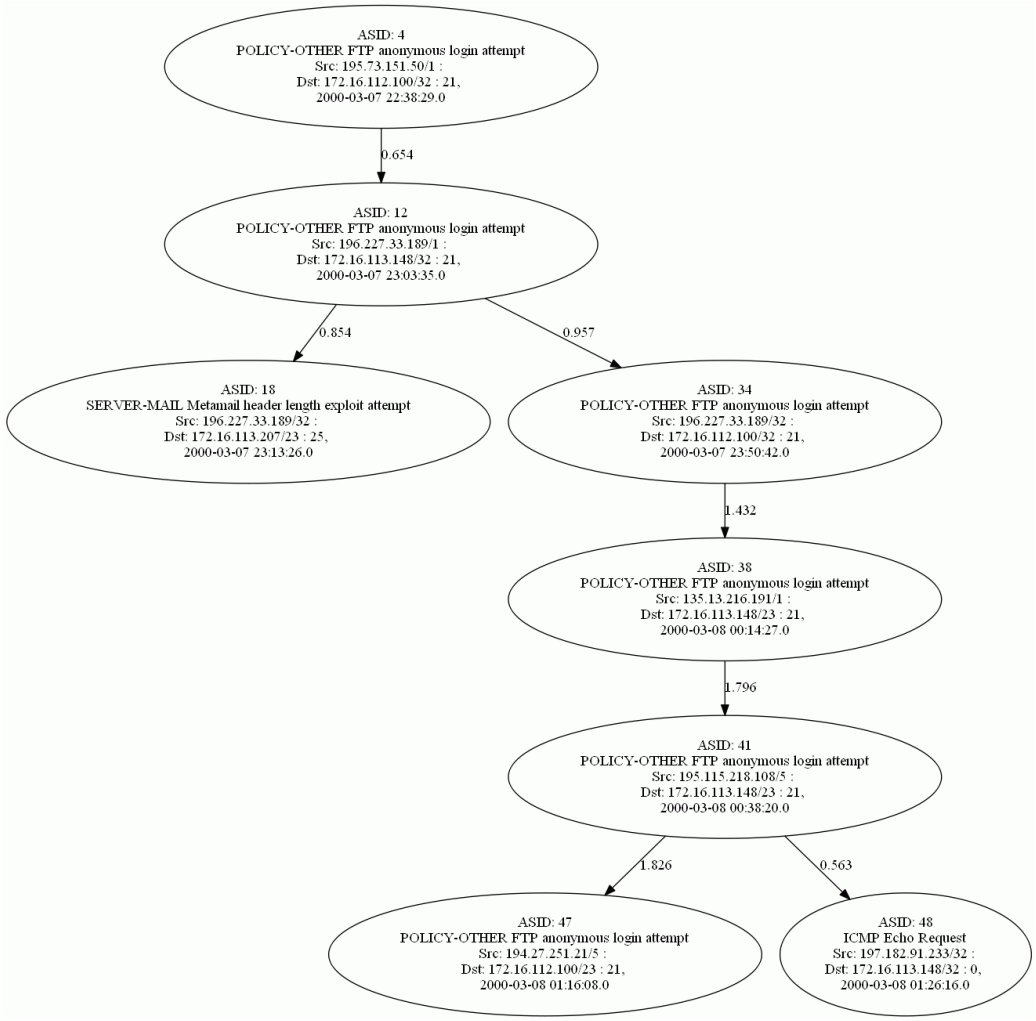
4.2 第二類型：以離線數據集建構警報關聯

在此類型中，我們使用離線型數據集 DARPA2000 資料庫 [14]測試我們的系統。DARPA 2000 是公認的入侵偵測系統評估數據集之一，並且它包含兩個多階段的攻擊場景：LLDOS1.0 和 LLDOS2.0.2。

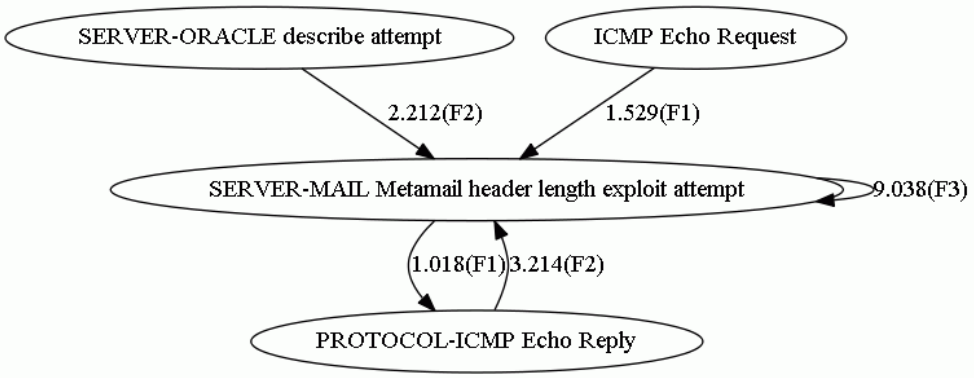
在 LLDOS1.0 攻擊情形中，這五步驟中描述了攻擊者執行 DoS 攻擊 MIT DARPA 網站的五個步驟。在 LLDOS1.0 實驗中，我們使用了 Snort 讀取 LLDOS1.0 攻擊場景的數據集並檢測惡意的數據包。

由 Snort 發出的 503 個警報，其中 256 個警報被對應於 LLDOS1.0 在經過特徵加權警報過濾後，剩餘的警報的數量是 472 個（不包括已知的誤報，DNS 響應警報），這些 472 個警報被合併成 50 個 AlertSets。該警報關聯步驟後，我們得到了多個 Hyper Alert 和攻擊圖。Hyper Alerts 為多個有向樹，一個 Hyper Alert 表示一個攻擊情境的實例（如圖六所示），圖內的點代表 AlertSet，內容記錄了來源目的 IP 位址、警報類型以及發生的時間，邊表示兩個 AlertSet 之間的關聯程度。此外攻擊圖代表的是一個攻擊情境（如

圖七所示），各個點表示警報的類型，邊表示警報類型之間的關聯度，由攻擊圖可以知道系統遭受到的攻擊情境。



圖六：由使用貝式網路模型建構可調式特徵權重之警報關聯系統實驗所得到的 LLDOS 1.0 Hyper Alerts



圖七：由使用貝式網路模型建構可調式特徵權重之警報關聯系統實驗所得到的 LLDOS 1.0 攻擊圖

伍、結論與未來工作

隨著網路日漸發達，各式網路服務供應商提供的便捷功能使我們愈來愈依賴它，甚至成為生活中不可或缺的一部分，也因此網路服務供應商擁有消費者私密的個人資訊，其中可能包含信用卡、消費紀錄...等，確保資訊安全便成為首要課題之一。

入侵偵測系統與攻擊圖即為提升網路安全的工具，將入侵偵測系統所發出之警報與攻擊圖中的攻擊策略配對，以達到快速追蹤、預測攻擊路徑等作用，進一步加強對資訊安全之防護。

本研究介紹了兩種類型的警報關聯方法，第一種類型分析系統漏洞所產生之攻擊圖；第二種類型依警報關聯為主體所產生之攻擊情境。未來我們將結合兩種類型之方法，由兩種類型的關連方式所組合成的混合式警報關聯系統，其中第一類型把警報對應到利用系統環境以及系統本身存在的漏洞所做成的攻擊圖，由於對於利用未知的系統漏洞產生的攻擊無法辨識，所以需搭配另一種類型方法來相互配合；第二類型利用過去所收集的警報統計以產生 Hyper Alert 結果，因需要一段時間來收集統計資料，第一類型相對於第二類型來說，形成攻擊圖與對應警報的速度比較快，而且對於已知攻擊的偵測較準確，運用兩種類型優點的相輔相成，達到可以準確辨識系統漏洞的攻擊以及未知的攻擊。

由於網路型入侵偵測系統無法測得主機上本地端的活動，例如檔案存取等動作便無法偵測，若能搭配附加程式偵測此類活動，將能提高攻擊圖配對之準確度。本研究中對於配對過後之警報僅做標記處理，避免重複讀取，若能將配對結果進行更多相關應用，例如警報關聯，或回饋給入侵偵測系統做即時調整，將配對失敗且無法進一步關聯之警報類型與來源 IP 加入白名單，避免入侵偵測系統產生重複警報，以降低 FP/FN 率。

參考文獻

[1] S. H. Ahmadinejad, S. Jalili and M. Abadi, "A Hybrid Model for Correlating Alerts of Known and Unknown Attack Scenarios and Updating Attack Graphs," *in International Journal of Computer and Telecommunications Networking, Volume 55, Issue 9,* 2011, pp. 2221-2240.

[2] F. Alserhani, M. Akhlaq, I. U Awan, and A. J. Cullen, "MARS: Multi-stage Attack Recognition System," *the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2010, pp. 753-759.

[3] M. Artz, "NetSPA- Network Security Planning Architecture," *Master's Thesis, Massachusetts Institute of Technology*, 2002.

[4] S. Jajodia, S. Noel and B. O'Berry, "Topological Analysis of Network Attack Vulnerability," in *Massive Computing, Volume 5,* 2005, pp. 247-266.

[5] X. Ou, W. F. Boyer, and M. A. McQueen, "A Scalable Approach to Attack Graph Generation," *the 13th ACM conference on Computer and communications security,* 2006, pp. 336-345.

[6] X. Ou, S. Govindavajhala, and A. W. Appel, "MulVAL: A Logic-based Network Security Analyzer", *the 14th USENIX Security Symposium,* 2005.

[7] P. Rao, K. F. Sagonas, T. Swift, D. S. Warren, and J. Freire, "XSB: A System for Efficiently Computing Well-founded Semantics," *the 4th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'97),* 1997, pp. 2-17.

[8] H. L. Ren, N. Stakhanova, and A. Ghorbani, "An Online Adaptive Approach to Alert Correlation," *Proc. 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Jul. 2010, pp. 153-172, doi: 10.1007.

[9] L. Wang , A. Liu and S. Jajodia, "Using Attack Graphs for Correlating, Hypothesizing, and Predicting Intrusion Alerts," *in Computer Communications, Volume 29, Issue 15, Elsevier,* 2006, pp. 2917-2933.

[10] C. H. Wang and J .M . Yang, "Adaptive Feature-Weighted Alert Correlation System using Bayesian Network Model," *Department of Computer Science and Information Engineering National Chiayi University Master Thesis*, July 2013.

[11] Z. Zali, M. R. Hashemi and H. Saidi, "Real-Time Attack Scenario Detection via Intrusion Detection Alert Correlation," *the 9th International ISC Conference on Information Security and Cryptology (ISCISC)*, 2012, pp. 95-102.

[12] S. Zhang, J. Li, X. Chen and L. Fan, "Generating Network Attack Graphs for Security Alert Correlation," *The 3rd International Conference on Communications and Networking*, 2008, pp. 230-235.

[13] B. Zhu and A. A. Ghorbani, "Alert correlation for extracting attack strategies," *International Journal of Network Security*, vol. 3, no. 3, Nov. 2006, pp. 244–258.

[14] MIT Lincoln Laboratory, 2000 Darpa Intrusion Detection Scenario Specific Data Sets, 2000.
Available:http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/