應用隱藏式馬可夫模型預測網路攻擊

陳奕明
中央大學資訊管理研究所
cym@mgt.ncu.edu.tw

官炳宏
中央大學資訊管理研究所
92423021@cc.ncu.edu.tw

孫文駿
中央大學資訊管理研究所
93443016@cc.ncu.edu.tw

摘要

網路攻擊預測能在警訊開始發生時，告知管理者後續最可能的攻擊目標，讓管理者有較充分的時間針對重點做防禦。這在資訊安全營運中心（Security Operation Center，SOC）人力資源有限，而又必需管理眾多網路設備，且面對不斷發生的網路攻擊時特別重要。但是要做到預測攻擊，必需先克服下面兩個問題：(1)如何從低階的攻擊警訊，辨識攻擊者的高階意圖？(2)一個攻擊警訊可能對應多個高階意圖，如何判斷哪一個最有可能是攻擊者的真正意圖？過去的研究，大多集中在關聯低階警訊（如 snort 所產生的警訊）以形成高階警訊（如 slammer 網蟲攻擊），很少提供預測攻擊的功能。為解決上述兩個問題，本論文提出一種結合隱藏式馬可夫模型（Hidden Markov Model, HMM）與彩色派翠網（Colored Petri Net, CPN）模型的技術。我們以彩色派翠網來表達攻擊者在多步驟攻擊後面的攻擊意圖，然後以隱藏式馬可夫模型來預測最有可能的攻擊意圖。論文中除介紹我們的方法外，也將介紹我們的系統雛形，實驗方法和實驗結果。我們的實驗結果顯示，隨著新的攻擊警訊不斷出現，隱藏式馬可夫模型可以更新攻擊者的攻擊意圖，讓網路管理者可以針對最新的攻擊威脅預作處理。

關鍵字：預測攻擊、隱藏式馬可夫模型、資訊安全營運中心、警訊關聯

一、前言

所謂預測攻擊(attack prediction)，是指在攻擊者發動網路攻擊的初期，網路安全管理者在僅收到少數入侵警訊時，便能預測攻擊者的下一步動作，甚至攻擊目標會是什麼？能做到預測攻擊的好處是這樣能讓管理者更專注於真正的安全威脅，而不會浪費時間心力於較不重要的警訊或較不可能受到攻擊的目標上。這個好處對於目前日益普及的資訊安全營運中心（Security Operation Center，SOC）來說顯得特別重要，因為這些中心往往人力有限而要管理的網路設備又非常多。但是要做到預測攻擊的困難在於：(1)如何從低階的攻擊警訊（例如 snort 發出一個 TELNET access 警訊），辨識攻擊者的高階意圖（例如攻擊者想要以 sadmind 漏洞進行 DDoS 攻擊）？(2)一個攻擊警訊可能對應多個高階意圖，如何判斷哪一個最有可能是攻擊者的真正意圖？例如 snort 發出的 ICMP Echo reply 警訊，代表攻擊者可能在做 IP 掃瞄，但是 IP 掃瞄完畢後是要進行 sadmind 的 DDoS 攻擊，還是要進行 FTP Bounce 攻擊呢？過去雖然有許多關聯攻擊警訊的研究被提出來，例如[1][11][15]，但大多仍集中在關聯低階警訊（如 snort 所產生的警訊）以形成高階警訊（如 slammer 網蟲攻擊），很少提供預測攻擊的功能。即使近年來較新的研究，如[2][10][17]等提出以前提(pre-condition)以及結果(posit-condition)的條件比對等方法來關聯警訊以重建攻擊情境(attack scenario)，但是因為相同警訊可能對應多個攻擊情境，如前面所述的"ICMP Echo reply 警訊"可能代表" sadmind 的 DDoS 攻擊"或" FTP Bounce 攻擊"，過去的研究即使能找出所有這些可能發生的攻擊情境，但是卻無法指出哪一個攻擊情境最可能發生，因此也就沒有完全達到預測攻擊的功能。

為了解決上述問題，本論文提出一種結合隱藏式馬可夫模型（Hidden Markov Model, HMM）與彩色派翠網（Colored Petri Net, CPN）模型的技術。我們以 CPN 來表達攻擊者在多步驟攻擊後面的攻擊意圖，然後以 HMM 來預測最有可能的攻擊意圖。會採用 HMM 的原因是因為該模型本來就是應用於根據系統的表象特徵（或稱為觀察值）去推測系統真實狀態（隱藏於系統之中，使用者無法由外界直接觀察得到）[8][20]，適用於攻擊預測上面。我們以圖 1 來說明 HMM 與攻擊預測的概念。

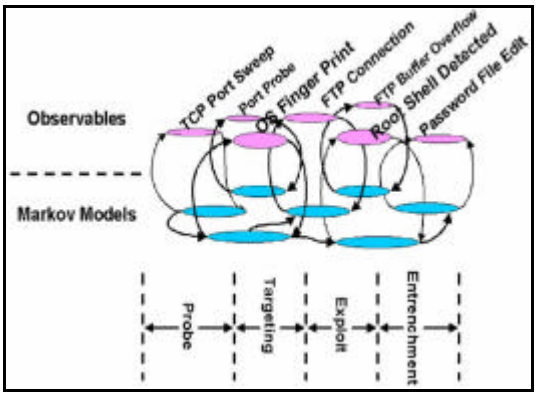

圖 1. 多步驟攻擊與狀態轉換圖之關係[6]

圖 1 中顯示我們將多步驟攻擊者的攻擊情境視為有限狀態機(Finite-State Machine)，

而每一個攻擊步驟視為有限狀態機中的一個狀態。攻擊者每完成一個攻擊步驟，就視為一次狀態轉換。入侵偵測系統所觀察到的低階警訊，稱為 HMM 觀察值，出現在圖 1 所示的 Observables 層；攻擊者真正的狀態，是隱藏的資訊無法從外界直接觀察，出現在圖中的 Markov Models 層。雖然我們僅能觀察到 Observables 層的資訊，但是 HMM 提供一套完整的演算法，包括 Viterbi 演算法和 Forward 演算法。前者能推論出觀察到的低階警訊所對應的最佳攻擊狀態序列，也就是攻擊者的攻擊情境，後者則能計算攻擊狀態序列和完成攻擊的發生機率。只要結合這兩種演算法，便能幫助我們做出攻擊預測。

Qin 和 Lee[21]曾提出一種機率模型，也能夠由警訊中辨識使用者的攻擊意圖以及預測攻擊。但是他們的方法需要事先建立因果網路(causal network)。但是由攻擊警訊對應到因果網路需要資訊安全專家的人工協助才有辦法完成，不像我們的系統，只要事先建立好各種 CPN 攻擊樣版（Qin 和 Lee 的系統也需要事先建立攻擊計畫資料庫），系統會自動將警訊套用到 CPN 攻擊樣版，然後啟動 HMM 計算模組進行攻擊預測。

我們根據所提出的技術完成一個雛形系統並以 DARPA2000 的警訊資料庫[9]進行測試。我們的結果顯示以 HMM 結合 CPN 的技術可以達到和 Ning 等人[4]一樣好的辨識高階攻擊情境的能力，而我們的系統還可以在多種可能攻擊情境中預測哪一個是最可能發生的情境，同時還可以提供量化的數據給管理者參考。

本文共分為五節。第二節將簡介 HMM 的原理並說明其如何與 CPN 共同建構我們的系統理論基礎；第三節說明我們的雛形系統、實驗過程與實驗結果；第四節討論兩個利用 HMM 於入侵偵測系統的相關研究，同時將他們的研究結果和我們的結果相比較。最後我們在第五節做一簡短結論並說明未來的研究方向。

二、CPN 與 HMM 的結合

我們之前曾成功地利用 CPN 關聯多步驟攻擊[2][3]，但是和 Ning 等人的相關研究[10][17]遇到的限制一樣，當遇到一個警訊可以對應到多個攻擊情境時，我們的 CPN 警訊關聯系統無法提供一個量化的數據告訴管理者哪一個攻擊情境最可能發生。因此在本節中，我們首先說明 CPN 用於攻擊預測上的限制，然後說明如何以 HMM 來補足 CPN 的不足。

2.1 採用 CPN 關聯多步驟攻擊的方法

CPN是一個圖形化的塑模工具，1980年代由 Kurt Jensen [16]所提出，和Petri Nets不同的地方是增加了Color Set（彩色集）的觀念，彩色集相當於資料型別（Data Types），使傳統的Petri Nets具有更直觀，豐富的表達方式。在我們的警訊關聯系統中，我們以CPN 中的Places 來表示攻擊行為的事前必要條件以及造成的結果；以Transitions來表示攻擊特徵或行動；Token 代表系統狀態。

我們以圖2為例，簡要說明CPN的使用。圖中Place用橢圓形或圓形表示，Transition則是用長方形表示之，Arc用箭頭表示，Arc上標示了Arc Expression，該Expression用來描述了一個Transition要發生所需要的條件；每個Place都有一組Token，而每個Token都擁有一個宣告過型別的值，當Token中的值符合Arc Expression的條件時，Transition就被啟動了。由圖2的攻擊樣版，我們可以看出，若被攻擊之主機存在portmapper及mountd服務，而且攻擊命令包含有rpcinfo 之命令字串，代表這是rpcinfo攻擊。若警訊關聯系統將圖2中各種警訊的IP位址比對為同一台主機且接收到的命令字串含有rpcinfo，則我們可以判斷rpcinfo 攻擊已經發生了。詳細的CPN攻擊模型建立方法請參考[3]。

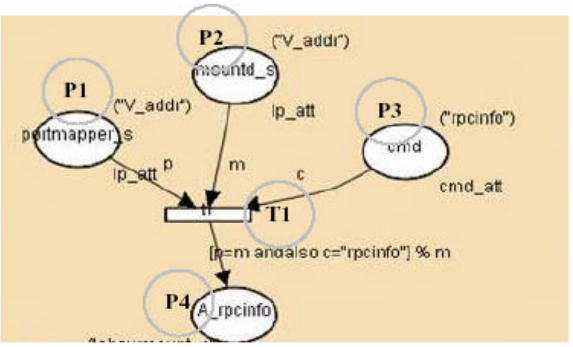

圖 2. rpcinfo 攻擊之 CPN 示意圖

CPN雖然能幫助我們順利表達多步驟攻擊情境，但是仍然有下列幾點不足之處：

(1) 無法找出警訊背後所真正隱含的攻擊意圖

這是因為同一個警訊有可能會對應到不同的攻擊狀態所致，比如在Sadmind Exploit的多步驟攻擊情境中，“TELNET access”警訊有可能是屬於第三個攻擊步驟“SadmindExploit”，也可能是屬於第五個攻擊步驟“DDoS Attack”，(參見圖3之CPN 樣版圖)。因此，若發生了“TELNET access”的警訊，CPN並沒有辦法分辨出是屬於哪一個攻擊步驟。

(2) 無法充份表達攻擊步驟之間的循環關係比如攻擊者在進行遠端的緩衝區溢位攻擊中，會不斷的嘗試入侵，企圖找出正確的堆疊指標，所以會進行循環的攻擊。而 CPN無法表達出這樣的關係。

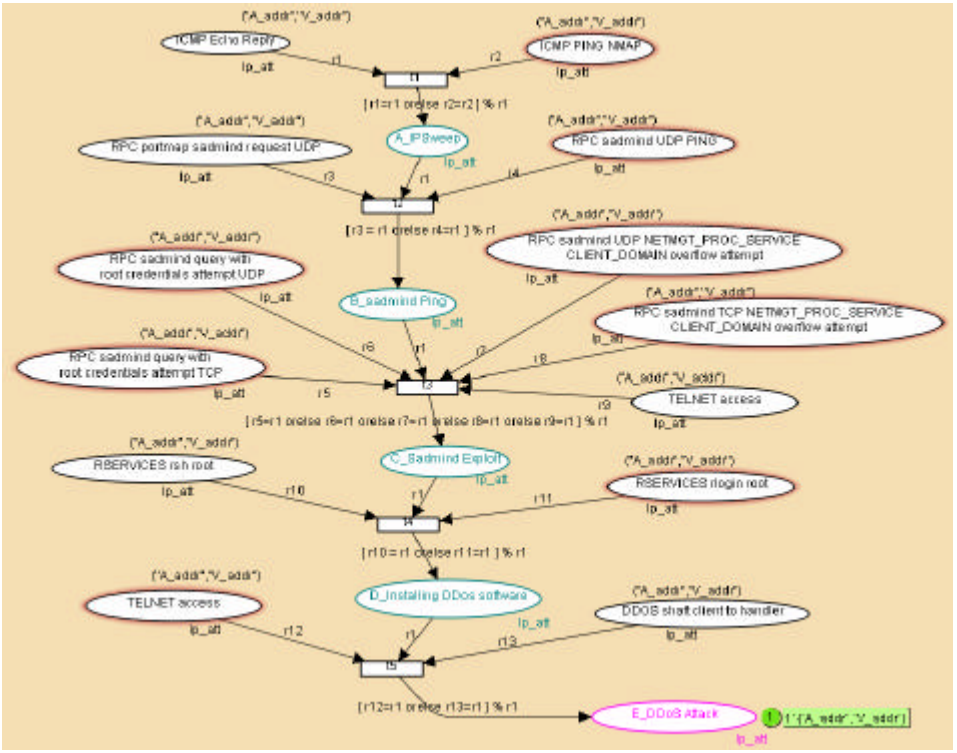

圖 3. 以 Sadmind 漏洞進行 DDoS 攻擊之 CPN Template 圖

(3) **無量化的概念**

無量化數值輔助讓資安人員評估攻擊發生的可能性，無法從攻擊初期少量之警訊預測攻擊者的真正意圖。

因為用CPN**來關聯警訊有上述限制，所以經研究後，我們決定加入**HMM**技術來予以解決。**

2.2 隱藏式馬可夫模型（HMM）簡介

Hidden Markov Models (HMM) [8] **最先由** Baum **和** Petrie **於** 1966 **年發展出來，已被廣泛應用於語音及符號辨識、訊號處理、**DNA **識別、分子生物學等領域，近年來也有諸多學者將** HMM **應用於多步驟網路攻擊偵測上**[5] [6] [7]。

一個 HMM **具有下列五項元素及特徵：**

(1) **有限個數的狀態（**states**），以集合** S **來表示，且** $S = \{ S_1, S_2, \ldots, S_N \}$。**在時間** t **的狀態為** q_t。**雖然狀態是被隱藏的資訊，但是實際的？用中，這些狀態代表實體上的意義，比如將** HMM **應用在多步驟攻擊之偵測上，則** HMM **裡的每一個不同的狀態就是表示攻擊者發動的各個攻擊步驟**[5]。**一般來說，狀態間是處於相互連結的關係，也就是任何的狀態都可以由其它的狀態轉換而來。**

(2) **在不同的狀態之下，狀態機會表現出不同的觀察值（**observation symbols**），以集合** O **表示，**$O = \{ O_1, O_2, \ldots, O_T \}$。**觀察值相當於狀態機的產出值，以多步驟攻擊為例，某一狀態所可能會產生的觀察值就可以被對應成為某一攻擊步驟所可能會觸發之警訊**[5]。

(3)**狀態之間的轉換存在機率關係，稱之為狀態轉換機率（**state transition probability**），定義為** a_{ij}，**且** $a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i]$, $1 \leqq i, j \leqq N$。**若有** N **個狀態，則能產生一個矩陣大小為** N x N **的狀態轉換矩陣（**state transition matrix**），以集合** A **來表示。**

(4) **觀察值會根據其所屬狀態的機率分配來產生，稱之為符號產生機率（**observation generation probability**），定義為** $b_j(k)$，**且** $b_j(k) = P[O_k \text{ at } t \mid q_t = S_j]$, $1 \leqq j \leqq N$, $1 \leqq k \leqq T$。**狀態及觀察值的相對關係，能產生一個符號產生機率矩陣，以集合** B **表示。**

(5) **在** t = 1 **時刻下，每個狀態不同的初始機率（**initial probability**），被定義為** p_i **且** $p_i = P[q_1 = S_i]$, $1 \leqq i \leqq N$。N**個狀態，則產生一個大小為** N **的初始狀態機率向量（**initial state probability vector**），以集合** p **來表示。**

一個完整的 HMM **需要指定上述的五項元素，分別為** S, O, A, B**以及** p，**為了簡化表示，可以使用** ? = (p, A, B) **來代表一個完整的** HMM **參數集（**parameter set**）。接著透過** ? **便可以畫出此** HMM **模型相對應的狀態轉換圖。**

在實際的應用上，HMM 能解決三種基本問題：(1)求解觀察序列的產生機率 P(O|?)；(2)求解某一觀察序列所對應的最佳狀態序列 Q = $q_1q_2…q_T$；以及(3)調整 HMM 模型的參數 ? =（p, A, B）以使得 P(O|?)值為最大。

在我們的攻擊預測中，並沒用到上述第三個問題的解法，所以我們對第三個問題在此不多加討論，僅針對前兩個問題的解法以及其在攻擊預測上的應用予以說明。

Forward 演算法解決上面第一個問題，即給定一串觀察序列 $O = O_1O_2…O_T$ 以及 ? = (p, A, B)，計算在 ? 的前提之下該觀察序列發生的機率（亦即 P(O|?)）為何？應用在攻擊預測上，就是指將蒐集到的警訊序列載入所有定義好的 HMM 模型中，就能夠自動分析得知該警訊序列最可能是屬於哪一種攻擊情境。

Viterbi演算法用來解決 HMM 上面第二個問題，亦即給定一串觀察序列 $O = O_1O_2…O_T$ 以及 HMM 模型 ? = (p, A, B)，如何求得相對應的最佳狀態序列 $Q = q_1q_2…q_T$？應用在我們的攻擊預測上，就是從蒐集到的警訊序列當中，找出相對應駭客最有可能發動的攻擊情境。

我們將上述兩種演算法結合，應用在攻擊預測上的做法就是：將攻擊初期所蒐集到的警訊視為觀察序列 O，所有訓練好的多步驟攻擊 HMM 模型列為候選模型，然後載入系統計算各候選模型中產生觀察序列 O 的機率值，挑選其中最大值者，就代表該警訊序列最可能是屬於哪一種攻擊情境。換句話說，假設系統中已經存在 HMM_Module_Num 個訓練完成之 HMM 模型，且 IDS 產生了一串警訊序列 alertSeq，我們的目的是希望預測在此警訊序列中到底隱含了何種攻擊行為。我們的做法是將所有訓練完成的 HMM 模型（即 hmm_i），以及 alertSeq 逐一代入到 Viterbi_Algorithm，這樣即可找出警訊序列 alertSeq 在不同 HMM 模型中的最佳狀態，接著再將求得之最佳狀態以及 hmm_i 模型代入 Forward_Algorithm 以計算出警訊序列 alertSeq 在 hmm_i 模型中的產生機率，而其中機率值最大者表示 hmm_i 為最有可能的攻擊行為。以上預測過程以虛擬碼（Pseudo Code）所示如下：

```
for i=1 to HMM_Module_Num
{
  predictState = Viterbi_Algorithm ( hmm_i , alertSeq ) ;
  probability = Forward_Algorithm ( hmm_i , predictState ) ;
}
Most_likely_attack= maximum (probability )
```

隨著警訊陸續的被 IDS 觸發，警訊序列 alertSeq 也不斷的增長，因此每一次代入預測演算法中進行預測，所求得的機率值愈加準確。相對之最有可能的攻擊機率值逐漸提高，當機率值達到資安人員設定之條件值(thread hole)，即會發出某種攻擊將臨的警訊。

從上面的說明，我們可知 HMM 具有以下兩個主要的特性可以幫助我們達到 CPN 無法完成的功能。第一是 HMM 具有嚴謹的數學理論支持；第二是 HMM 能夠加以訓練，經由不斷的訓練，可以使得 HMM 的模型參數能夠愈加的收斂也愈加的趨近於攻擊者的攻擊特性，如此便能有效的提升預測攻擊的準確性。

三、雛形系統與實驗結果

本節介紹結合 CPN 和 HMM 技術的預測攻擊系統雛形以及實驗的過程和結果。首先我們說明如何由 CPN 樣版建立 HMM 模型(即上一節的 hmm_i)，同時提出調適型 HMM(Adaptive HMM, AHMM)的觀念以修正直接採用 HMM 會產生的問題。

3.1 多步驟攻擊轉換為狀態轉換圖

在開始預測攻擊前我們要需要專家協助，將每一個多步驟攻擊情境轉換成 HMM 狀態轉換圖，這動作分為三個步驟，我們以 Sadmind Exploit for a DDoS Attack[10]為例逐一說明。

(1) 建立 CPN 樣版(Template)圖形

第一個步驟是將已知的多步驟攻擊情境利用CPN工具(CPN Tool) [16]轉換為CPN 樣版圖形（參見圖4）。專家必須先蒐集完整的多步驟攻擊情境，然後再參考IDS的規則，找出有哪些警訊可能達成這個攻擊情境裡面的攻擊步驟。再利用CPN工具將攻擊步驟、相關警訊、以及攻擊的必要條件等資訊繪製成CPN樣版圖型。

(2) 將 CPN 樣版轉換為 HMM 的狀態轉換圖

我們將CPN 樣版圖上每一個攻擊步驟視為是HMM裡面的狀態（states）；而警訊則當作是HMM中的觀察值[5]；第一個攻擊步驟為初始狀態；攻擊發起步驟的選擇為該狀態的初始機率值（p）。攻擊步驟與攻擊步驟之間的轉換的發生機率，就是HMM的狀態轉換機率（A）。攻擊者藉由攻擊手法完成某攻擊步驟，會觸發相對應警訊，不同警訊的發生機率，相當於HMM的觀察值產生機率（B）。整個CPN 樣版與HMM之間的對應關係如圖4的例子所示，例如在該圖中共有5個狀態以及13個觀察值，而p_A代表攻擊者可能會從狀態IPSweep開始入侵的初始機率，a_{AB}為攻擊狀態會由IPSweep移到SadmindPing的狀態轉換機率，而至於b_A(A1)則表示當攻擊者進入到了

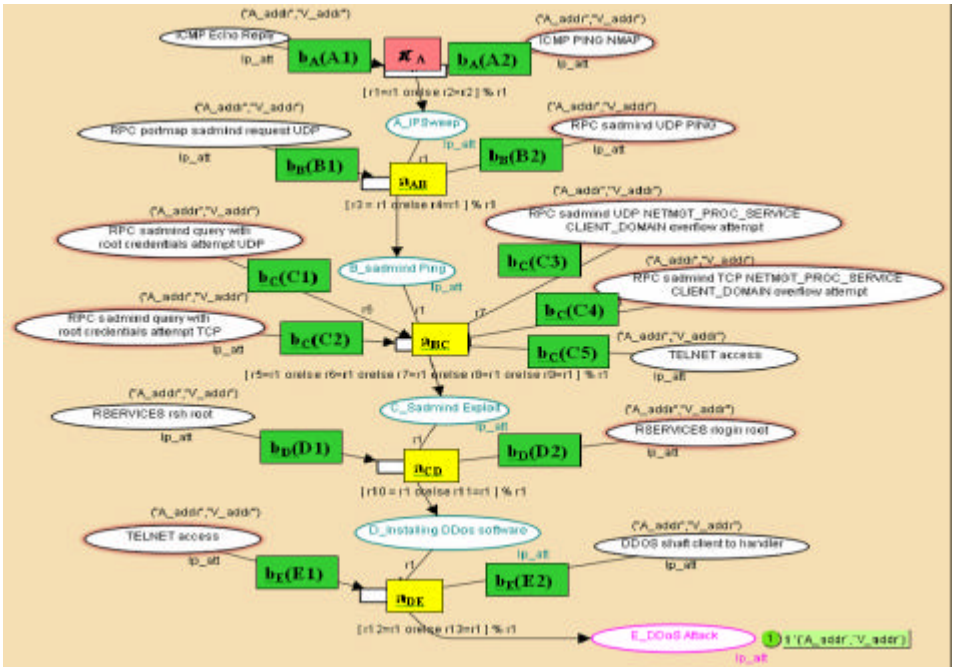

圖 4. CPN 樣版與 HMM 之對應關係圖

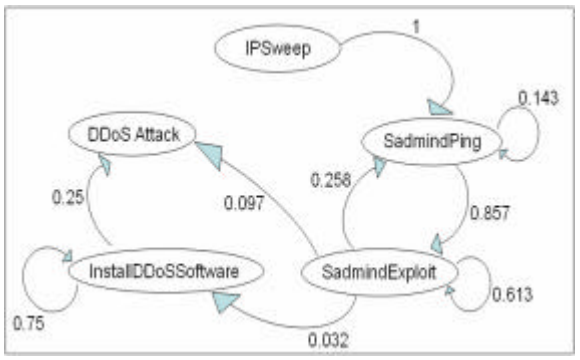

圖 5 Sadmind 攻擊之 HMM 狀態轉換圖

IPSweep 的狀態，可能會觸發 ICMP Echo Reply 警訊的符號產生機率。

(3) 訓練 HMM 的狀態轉換圖

我們將事前收集到的警訊資料視為訓練資料 (training data) 對狀態轉換圖進行訓練的動作，我們得到三種參數值（p, A, B），訓練之後的結果如圖5所示。

3.2 調適性隱藏式馬可夫模型概念

傳統的 HMM 針對攻擊警訊的處理並沒有考慮到 AND 與 OR 的關係，如此一來會喪失掉警訊之間隱含的意涵，導致計算失誤。以下將就圖 6 以及圖 7 來分別說明為何傳統的 HMM 並沒有考慮到警訊之間 AND 與 OR 的關係。

圖 6 是一個 OR 關係的 CPN 樣版圖。由該圖我們可以看出若要達到 IPSweep 的攻擊步驟，可以有圖上所示的五種不同攻擊手法，換句話說，要進入 IPSweep 的狀態必須要滿足的必要條件是 [r1=r1 orelse r2=r2 orelse r3=r3 orelse r4=r4 orelse r5=r5] % r1。從這個 guard function 的表示式可以看的出來，這五種攻擊手法之間是屬於 OR 的關係且根據 Snort 規則集（rule sets）的定義，其彼此相互獨立，因此若愈多 place 被觸發，則表示會進入 IPSweep 攻擊狀態的可能性愈大，因此想法上我們應該要將這些被觸發的 place 所對應的觀察值產生機率相加起來。圖 6 中，若左邊有 token 值的兩個 place 被觸發，則用傳統 HMM 求警訊序列發生機率的計算方式為

$$P_{HMM} = p_i \, b_i(O_1) \, a_{ii} \, b_i(O_2) \qquad (1)$$

其中 IPSweep 狀態用變數 i 代表，警訊 “ICMP Echo Reply”以及“ICMP PING NMAP”則分別用變數 O_1 與 O_2 來表示。從此計算式可以看出，若系統再觸發了一個新警訊，則只是再將一個狀態轉換機率及一個觀察值產生機率乘上去而已，並沒有考慮到警訊之間是屬於 OR 的關係，如此一來，代表會進入 IPSweep 狀態的警訊序列發生機率反而是愈來愈低；因此，我們所提出的 AHMM 會考慮 CPN 中 guard function 的語意，若該 function 中含有 orelse 關鍵字，代表警訊間是 OR 關係，如此計算式改為

$$P_{AHMM} = p_i \, [\, b_i(O_1) + b_i(O_2) \,] \qquad (2)$$

(2)算出來的機率值會大於(1)算出的值，而且當愈多的 place 被觸發，則 P_{AHMM} 值愈大，這樣才算是符合攻擊的真實情況。

圖 7 是一個 AND 關係的 CPN 樣版圖，從 guard function 上的條件式“andalso”可以看出，若要達到 Direct DNS attack 的攻擊狀態，則勢必圖上的兩個 place 都要被觸發，因此我們應該要將這些 place 所對應的觀察值產生機率相乘，以避免若只有一個 place 被觸發而已，還是能進入到 Direct DNS attack 攻擊狀態的情形發生。圖 7

中，若只有右上角的 place 被觸發，則傳統 HMM 求警訊序列發生機率的計算方式為

$$P_{HMM} = p_i\, b_i(O_1) \;?\; 0 \qquad (3)$$

其中 Direct DNS attack 狀態用變數 i 來代表，警訊“DOS Winnuke attack”則用變數 O_1 來表示。由式子(3)可看出來求得的機率值會是一個非零的數值，但是在入侵偵測上，這是一個錯誤的答案，因為如同上面所述的，必須要兩個 place 都被觸發的情形下才能進入到 Direct DNS attack 的攻擊狀態；因此同樣地，我們所提出的 AHMM 會考慮 CPN 中 guard function 的語意，若該 function 中含有 andelse 關鍵字，代表警訊間是 AND 關係，如此計算式改為

$$P_{AHMM} = p_i\, [\, b_i(O_1) \times b_i(O_2)\,] \qquad (4)$$

式子(4)的結果為 0,這是因為當只有一個 place 被觸發，表示攻擊沒有成功，所以計算的機率值也應該為零。

由上面的說明，我們可以看出如果採用本文所提出的 AHMM 取代直接引用 HMM，將能夠降低攻擊誤判的情形。

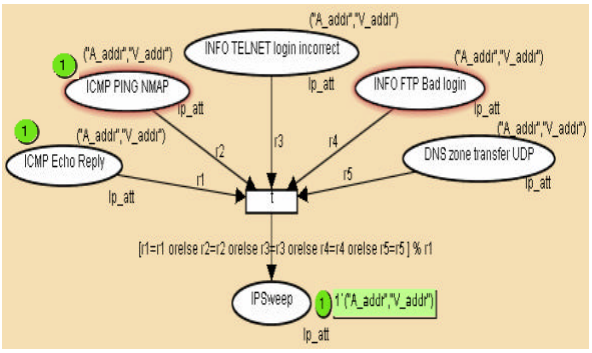

圖 6. 警訊關係為 OR 之攻擊步驟

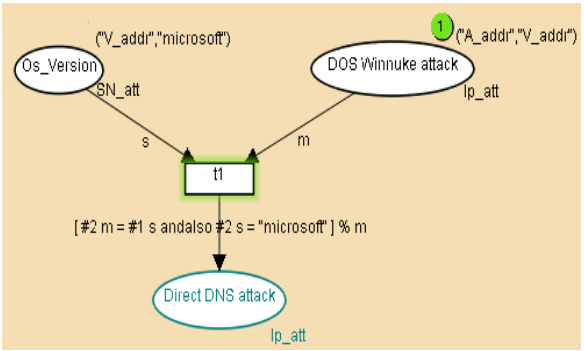

圖 7. 警訊關係為 AND 之攻擊步驟

3.3 雛型系統設計與實做

本論文提出一系統架構圖，如圖 8 所示。一開始我們利用對攻擊流程的瞭解或是從外部得來的攻擊樣式，利用 Attack Pattern Addition Module 將攻擊案例用 CPN 予以建模，再增加到 CPN 樣版的資料庫當中；透過已建立完成的 CPN 樣版以及訓練資料，一起載入 HMM Generator Module 中便能得出經過訓練後的 HMM 模型；另外，將各 IDS 所產生的警訊標準化後，經由 Alert Filter 的處理，便能產生多組的 Examples，而 Example 的定義為從一來源位置到另一目的位置間所有依時間排序的警訊。最後將已訓練過的 HMM 模型與 Examples，甚至是一些弱點輔助資訊一起輸入 Adaptive HMM Module 當中，便能產生數值型態的攻擊報告。

以下將針對系統架構中的各個元件進行較詳細之說明。

1. IDSs

   各種入侵偵測系統的裝置，整合由各種 IDS 所產生出來初期的警訊供其它模組使用。

2. Network Status

   弱點輔助資訊，例如關於主機的弱點稽核檔等等。

3. Internet Attacks

   網路上的攻擊，或可由各處蒐集來的多步驟攻擊案例。

4. CPN 樣版 s Database

   在此資料庫中，所存放的為 Attack Pattern Addition Module 轉換各多步驟攻擊案例為 CPN 模型的 CPN 樣版，由於此資料庫中所存放的皆為以 CPN Tools 所描繪之圖形，因此皆以一個個的 XML 檔案型式存在。

5. Numerical Attack Report

   此為對警訊序列（Examples）進行分析之後所產生的多步驟攻擊報告。

6. Attack Pattern Addition Module

   在網路上所蒐集到以及我們所瞭解的多步驟攻擊，是一堆凌亂的資料，經由此模組，可將這些攻擊資料轉換為 CPN 樣版 s。而此模組的轉換工作亦可經由人工來完成。

7. Attack Pattern Modification Module

   此模組的主要目的是用來修正之前所建立之 CPN 樣版 s。之前所建立的 CPN 樣版 s，可能經由事後額外的知識經驗，而需要做些微修改，則可透過此模組進行攻擊樣式的修改。同樣亦可由人工來進行修改。

8. HMM Generator Module

   此模組有二個主要的功能，第一個是將資料庫當中的 CPN 樣版轉換為 HMM 模型，而第二個功能則是載入警訊資料以便將 HMM 模型加以訓練。另外，還能將訓練過後的 HMM 模型加以儲存以進行日後的再訓練動作。

9. Alert Filter

當我們獲得一堆尚未處理過之警訊資料，將之放入 Alert Filter 當中，便能產生一組一組不同 IP 組合的 Examples 以便 Adaptive HMM Module 分析。

10. Adaptive HMM Module

在此模組中主要是要進行警訊序列的分析工作，企圖找出隱藏在警訊堆當中的多步驟攻擊行為。而要載入的資料有，已訓練過的 HMM 模型與經由 Alter Filter 所產生的 Example，甚至是一些弱點輔助資訊。經由此模組內的一些相關演算法之分析運算，即能產生攻擊報告。

本系統的開發環境，作業系統為 Windows XP，使用的是 JAVA 語言且版本為 1.5.0，另外，還使用了 CPN Tools 來當成繪製 CPN 樣版的工具。在 HMM 的 API 方面是使用 GPL 授權模式的 Jahmm v0.3.3[22]。在 IDS 方面是使用著名的 Snort 入侵偵測系統，而弱點輔助資訊則是取自 Nessus 這套弱點掃描工具。

以下我們首先針對 HMM Generator Module 來進行說明與實作。圖 9 為實作 HMM Generator Module 之畫面，第一個步驟是先選擇我們想要轉換為 HMM 模型的 CPN 檔案，程式便會自動地對該 CPN 檔案進行過濾資訊以及判斷的動作，所有在 CPN 檔案上的攻擊步驟以及相對應可能會被觸發的 place 都會被轉成狀態以及觀察值，並將轉換後的結果呈現於 HMM Module 的方框內。接著步驟二允許使用者選擇適當的訓練資料來對上一個步驟轉換後的 HMM 模型進行訓練，選擇了警訊檔之後，程式會將該警訊檔的內容呈現於畫面下方的 Raw Alert 方框內，當使用者按下 Training 按鈕後便開始進行訓練的動作，並將結果包含初始機率、狀態轉換機率以及觀察值產生機率等 HMM 參數機率值顯示在 Lambda 的方框中。產生了訓練結果之後，還能透過第三個步驟將訓練過之 HMM 模型以物件的方式儲存起來。這樣的好處是，當我們又獲得了另外的訓練資料時，則可以選擇步驟四將要進行再訓練的 HMM 模型給載入程式當中，接著選擇訓練資料檔，按下 Training 按鈕之後即可進行再訓練。

接著我們再針對第二個重要模組 Adaptive HMM Module 進行說明與實作，如圖 10 所示。在這模組當中，首先第一個步驟使用者必須選擇欲進行分析的警訊資料檔，選好之後，程式便會將此警訊資料檔分成一個一個由不同來源位置以及不同目的位置所組成的 IP 組合，再將分類後的結果呈現於 IP Combinations 的方框內。接著，可以選擇想要進行分析的 IP 組合，再透過 Copy 按鈕將 IP 組合給複製到右邊的 Queue 方框內。而第二個步驟則是選取經由 HMM Generator Module 產生的 HMM 模型，程式會將該選擇的 HMM 模型之相關內容，包含如狀態、觀察值、初始機率、狀態轉換機率以及觀察值產生機率等等資訊分別顯示在 HMM Module 以及 Lambda 的方框中。最後第三個步驟則是按下 Analysis 按鈕以便進行 IP 組合內之警訊序列的分析動作。

3.4. 實驗結果與討論

在開始進行預測攻擊的實驗之前，我們必須先行建立並訓練好多步驟攻擊的 HMM 模型。在本論文的實驗中，我們建立了兩個 HMM 模型：

HMM 1 - Sadmind Exploit for a DDoS Attack

此攻擊由五個步驟所組成，探測、非法入侵、安裝 DDoS 軟體以及對遠距伺服器發動 DDoS 攻擊。利用 Solaris 平台上的 Sadmind 程式所具有的緩衝區溢位弱點，攻擊者正確找出一個正在執行的 Sadmind 程序中的堆疊指標並覆蓋，就能成功的入侵此台主機，遠端取得管理者權限以執行任意程式碼。此攻擊樣版參見圖 3.

HMM 2 - FTP Bounce Attack

攻擊者利用“目標主機”所信賴“FTP 主機”上之使用者的遠端介殼程式服務（remote shell service）。命令其開始下載包含 rsh 訊息的檔案到客戶端。由於 rsh 協定裡的認證弱點而執行它。讓攻擊者一個擁有 root 權限的介殼程式（shell）。此攻擊樣版參見圖 11。

以上兩個模型，我們首先利用專家知識[17]建立攻擊情境和攻擊步驟，而後我們使用美國麻省理工學院林肯實驗室所提供的 DARPA 2000 當成警訊資料的來源[9]，且因為在 LLDOS 2.0.2 當中的攻擊是以比較隱密的方式進行攻擊，所以有些攻擊步驟的警訊並沒有被我們所採用的 Snort 入侵偵測系統所觸發，因此我們是用 LLDOS 2.0.2 另外加上 LLDOS 1.0 當中一組成功的攻擊案例當成訓練資料以求得 HMM ? 當中的（p, A, B），表 1、表 2 以及表 3 所示為 HMM 模型 1 經訓練所產生的結果。表 4、表 5 以及表 6 所示為 HMM 模型 2 經訓練所產生的結果。

若一開始 IDS 產生警訊“ICMP Echo Reply”，則藉由 Adaptive HMM Module 此一模組裡包含的 Viterbi 演算法以及 Forward 演算法的計算，便可以求得下列的二組警訊發生機率值：

HMM 1：$P_{Sadmind} = 0.25 \times 1 = 0.25$

HMM 2：$P_{FtpBounce} = 0.667 \times 0.25 = 0.167$

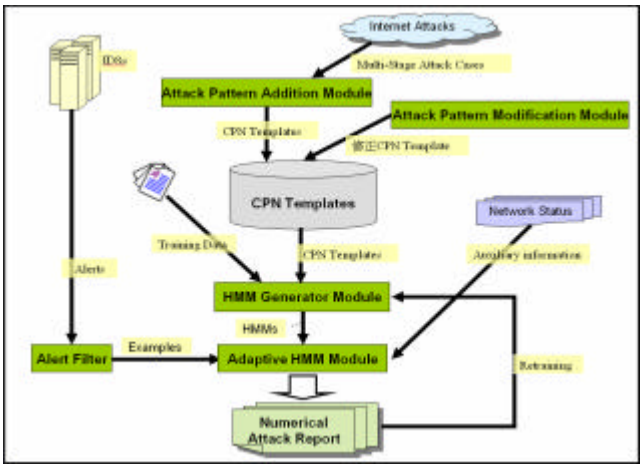

圖 8.系統架構圖

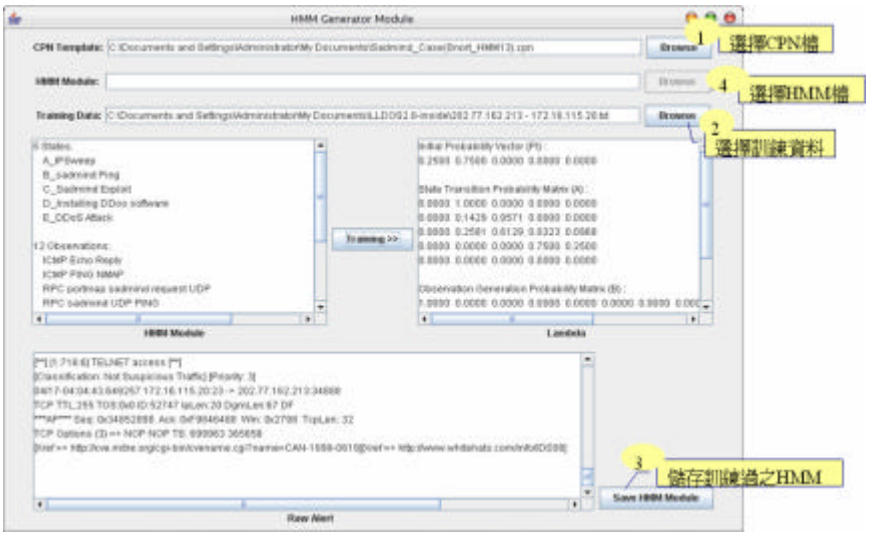

圖 9. HMM Generator Module 之實作畫面圖

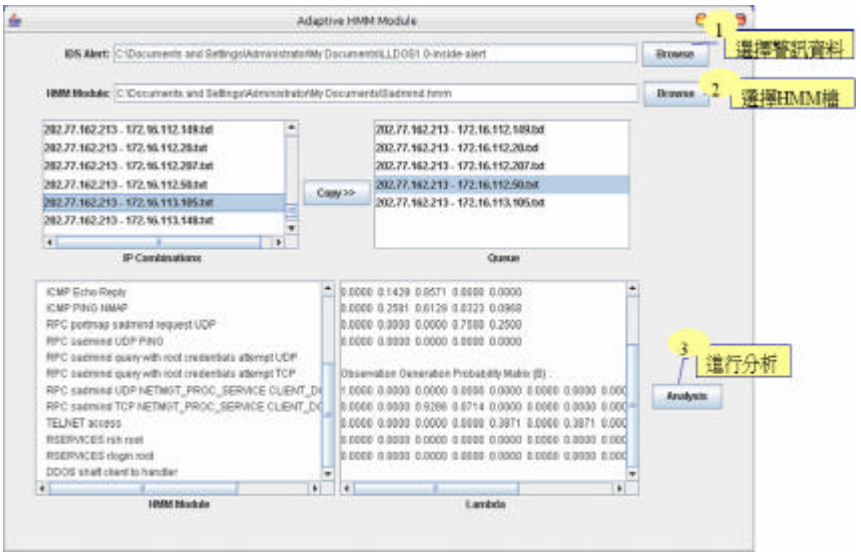

圖 10. Adaptive HMM Module 之實作畫面圖

由上面 2 個機率值的比較可知，$P_{Sadmind}$ 的機率值 0.25 是最高的，因此可以預測警訊"ICMP Echo Reply" 最有可能是屬於 Sadmind Exploit 的攻擊行為。當 IDS 又產生了一個新的警訊"RPC portmap sadmind request UDP"，則警訊序列裡就有了兩個警訊，同樣可以代入 Adaptive HMM Module 裡求得出下列的五組機率值：

HMM 1：$P_{Sadmind} = 0.25 \times 1 \times 1 \times 0.929 = 0.23$

HMM 2：$P_{FtpBounce} = 0.667 \times 0.25 \times 0 = 0$

由上面兩個機率值的可以看出，"RPC portmap sadmind request UDP"最有可能是屬於 Sadmind Exploit 的攻擊行為。在 $P_{FtpBounce}$ 的計算中，因為警訊"RPC portmap sadmind request UDP"並不包含在 FTP Bounce Attack 的觀察值當中，因此即使前一個警訊"ICMP Echo Reply"是有可能屬於這個攻擊的，然而整個警訊序列在這個攻擊當中卻是不可能發生，因此算出來的機率值為零。

為了避免現階段 HMM 模型的不足，而導致無法從現有的實驗中充份看出本論文應用在預測攻擊上的能力，因此我們將做一個假設，期望能先暫時排除因為 HMM 模型之不足所帶來的效用不明顯的問題。做法如下：首先將 HMM 模型一（Sadmind Exploit for a DDoS Attack）中的第二個攻擊步驟變成 PortScan 的攻擊同時也將原本的觀察值"RPC portmap sadmind

表 1. HMM 1 初始機率 p

State A	State B	State C	State D	State E
0.25	0.75	0	0	0

表 2. HMM 1 狀態轉換機率 A

	State A	State B	State C	State D	State E
State A	0	1	0	0	0
State B	0	0.143	0.857	0	0
State C	0	0.258	0.613	0.032	0.097
State D	0	0	0	0.75	0.25
State E	0	0	0	0	0

表 3. HMM 1 符號產生機率 B

	Alert1	Alert2	Alert3	Alert4	Alert5	Alert6	Alert7	Alert8	Alert9	Alert10	Alert11	Alert12
State A	1	0	0	0	0	0	0	0	0	0	0	0
State B	0	0	0.929	0.071	0	0	0	0	0	0	0	0
State C	0	0	0	0	0.387	0	0.387	0	0.226	0	0	0
State D	0	0	0	0	0	0	0	0	0	1	0	0
State E	0	0	0	0	0	0	0	0	1	0	0	0

表 4. HMM2 初始機率 p

State A	State B	State C	State D	State E
0.667	0.333	0	0	0

表 5. HMM2 狀態轉換機率 A

	State A	State B	State C	State D	State E
State A	0.50	0.50	0	0	0
State B	0	0.824	0.177	0	0
State C	0	0	0.625	0.375	0
State D	0	0	0	0.75	0.25
State E	0	0	0	0	1.00

表 6. HMM2 符號產生機率 B

	Alert1	Alert2	Alert3	Alert4	Alert5	Alert6	Alert7	Alert8	Alert9	Alert10
State A	0.25	0.75	0	0	0	0	0	0	0	0
State B	0	0	0.118	0.882	0	0	0	0	0	0
State C	0	0	0	0	0.625	0.375	0	0	0	0
State D	0	0	0	0	0	0	0	1	0	0
State E	0	0	0	0	0	0	0	0	0.833	0.167

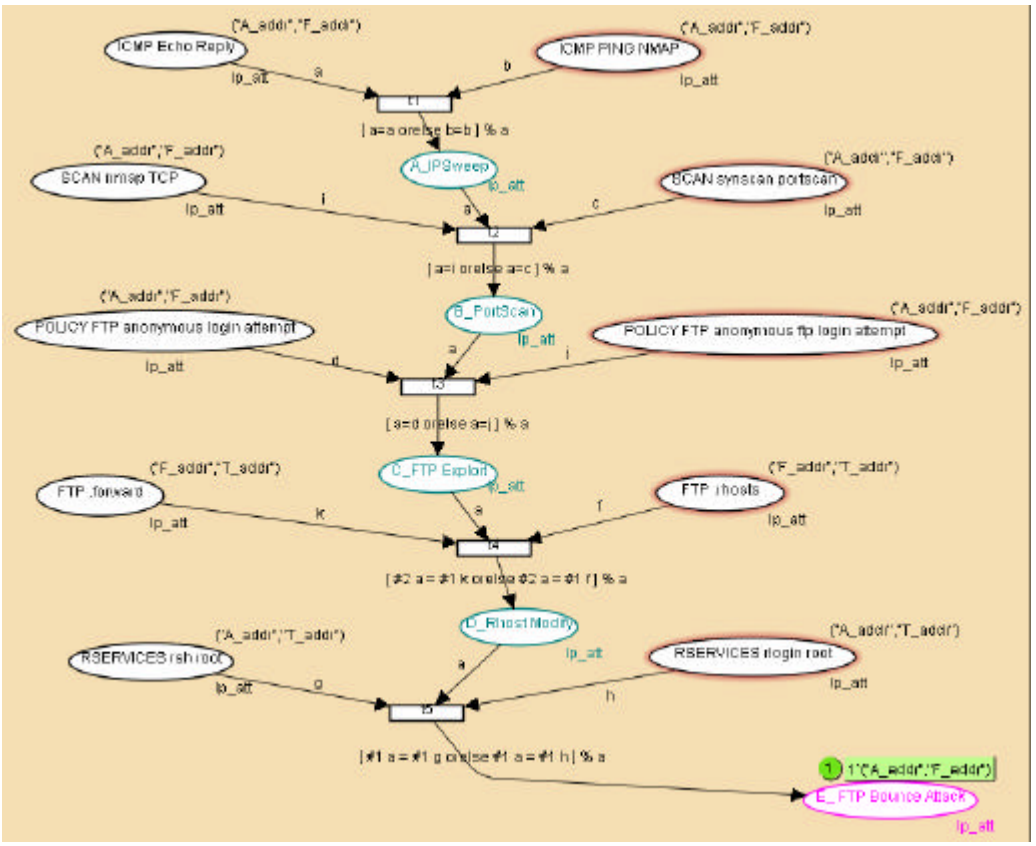

圖 11. FTP Bounce Attack 之 CPN Template 圖

request UDP" **改為** "SCAN nmap TCP" **以及** "RPC sadmind UDP PING" **改為** "SCAN synscan portscan"**，此外，我們也假設狀態轉換機率及符號產生機率均不變。如此一來，在** Sadmind Exploit for a DDoS Attack**以及** FTP Bounce Attack **這兩個** HMM **模型中的第一個攻擊步驟就會是原本就相同的** IPSweep**，第二個攻擊步驟則變成是相同的** PortScan**。**

我們依據上面的調整再做一次實驗，第一個警訊仍然是"ICMP Echo Reply"**，但是當** IDS**產生了第二個新的警訊** "SCAN synscan portscan"**，同時代入** Adaptive HMM Module **求出新的機率值：**

HMM 1：$P_{Sadmind}$ = 0.25 x 1 x 1 x 0.071 = 0.018

HMM 2：$P_{FtpBounce}$ = 0.667 x 0.25 x 0.5 x 0.882 = 0.074

由上面的計算結果可以看出，$P_{FtpBounce}$**的機率值** 0.074**比** $P_{Sadmind}$**的機率值** 0.018**來的高，這時候我們可以看出最可能發生的攻擊從原本的** Sadmind Exploit **攻擊轉變為是** FTP Bounce Attack**。探究其中最主要的原因在於** HMM 1 **中以** SCAN synscan portscan **手法達成** PortScan **的機率僅為** 0.071**，代表的意義即為攻擊者不太可能使用這個攻擊手法達成其階段性目的，故會使得整個警訊序列發生的機率大幅下降，這也就是為何經由** Adaptive HMM **的判斷之後認為攻擊者比較有可能的攻擊行為是屬於** HMM 2**，**FTP Bounce Attack**的原因。**

換句話說，也就是當只發生了 "ICMP Echo Reply" **此一警訊時，最有可能的攻擊是** Sadmind Exploit**，而當接著** IDS **接著又觸發了下一個警訊** "SCAN synscan portscan" **時，則經由** Adaptive HMM **的分析預測最可能攻擊者發動的攻擊反而從原先的** Sadmind Exploit **攻擊轉變為是** FTP Bounce Attack**。因此可以看出，隨著警訊序列內容不斷的變化，**Adaptive HMM **的確是有足夠的能力來預測出該警訊序列最可能是屬於何種類攻擊。而且，當有愈來愈多的多步驟攻擊手法被轉換為** HMM **模型並且適當的加以訓練，**Adaptive HMM **的這項預測攻擊的能力就可以被發揮的更加淋漓盡致，幫助資訊安全人員真正做到從初期少數被觸發的警訊當中，就能預測出這些警訊背後所真正隱含的攻擊意圖。**

四、相關研究

本節我們簡單介紹將 HMM **應用在** IDS **上的兩個相關研究，並與我們的結果做定性比較。**

Ourston **等學者** [6] [18]**提出將隱藏式馬可夫模型應用於多步驟網路攻擊偵測上的概念，其具體做法是將** HMM **當成分類器的角色，整個**

分類的過程如圖 12 的系統架構圖所示，首先會對不同入侵偵測系統所收集到的警訊進行處理，處理的過程可以分為兩個步驟，第一個步驟是將警訊資料庫內重複的警訊移除，由 Data Pre-Filtering 模組來負責；另外一個步驟則是將已過濾完後的警訊透過 Connection Records 模組組合成一個一個的 Example，而 Example 的定義為從某一來源位置到另一目的位置間 24 小時內且依時間排序的警訊資料。接著便可以將 Examples 載入 HMM 模組當中進行分析運算即可求解出 Examples 所代表的警訊序列最有可能屬於何種已事先定義好的攻擊。

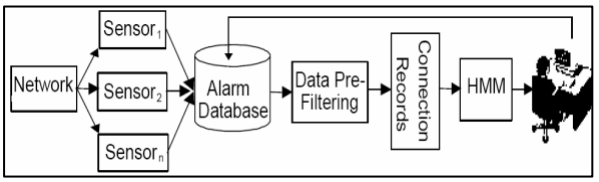

圖 12. Ourston 學者之系統架構圖[5]

在分類結果方面，Ourston 等學者直接與決策樹以及類神經網路這兩種古典的機器學習方法進行比較，結果證明了 HMM 比決策樹（C4.5）分類效果更好且遠勝於類神經網路（NN）。然而，有一點不足之處是，經由 HMM 的分類器，雖能分辨警訊序列屬於何種類攻擊，但並無法讓人得知該警訊序列是否已成功達成攻擊，或僅是屬於無效的攻擊行為。

Dong Yu 等學者[7]基於彩色派翠網（CPN）的理論提出一種新的隱藏式彩色派翠網（Hidden Colored Petri Net，HCPN）的架構，主要是利用代理人（agents）存取系統資源（resources）這樣的概念取代傳統 CPN 上 Place 屬於彩色集（color sets）的關係，並且加入了類似於 HMM 的想法，正式的 HCPN 定義共包含了 11 個元件，即 HCPN = (S, Q , D , A , O , G , E , $?_0$, ?, G, ?)，分別如下所示：

1、 S（color set），代表代理人（agent）的非空且有限集合。
2、 Q（place set），代表資源（resources）的有限集合。
3、 D（transition set），代表代理人可能發動的行為（actions）的有限集合。
4、 A（arc set），代表 A_1 及 A_2 聯集的有限集合，其中 A_1 表 place 到 transition；而 A_2 表 transition 到 place。
5、 O（observation set），代表觀察值的集合。可以是警訊或原始稽核檔。
6、 G（guard function set），代表與 A_1 有關的必要條件的集合。
7、 E（effect function set），代表與 A_2 有關的事後結果的集合。
8、 $?_0$（initial marking distribution），表示代理人與資源之間的初始機率分配。
9、 ?（transition probability），表示下一個攻擊者行為會被進行的機率。
10、 G（observation probability），表示攻擊行為產生某觀察值的機率。
11、 ?（tolerance），用來決定兩個狀態是否為難以辨別的。

由上述的 HCPN 定義中我們不難發現，其想法與隱藏式馬可夫模型（HMM）的基本概念是極為相同的，因為代理人對不同的資源擁有不同的初始機率、攻擊行為之間的轉換存在有轉換機率以及同一個攻擊行為對不同的觀察警訊存在有不同的產生機率。

圖 13 為將一個 local-to-root（L2R）攻擊[11][12]轉換成 HCPN 的表示圖。在這個多步驟攻擊中共包含四個攻擊行為，分別是 copy, chmod, touch 以及 mail，而每一個攻擊行為均會導致攻擊者存取到某一資源。而圖中有另外一個非攻擊行為的 transition，稱之為 normal，用來表示一些不是入侵的行為。圖中的六個 places 被用來代表相關的資源，其中一個比較特殊的是 q1，它所代表的資源可被所有的代理人存取。另外還有箭頭被用來表示攻擊行為的必要條件及事後結果。在圖下方虛線所連結到的所有警訊 Alert 1到 Alert N表示攻擊行為所可能會觸發的可觀察警訊（observable symbol），而與警訊個別相關的四個攻擊行為及一個正常的行為則是屬於隱藏的狀態（hidden state）。因此目的同樣如隱藏式馬可夫模型一樣希望能從可觀察到的警訊推出隱藏起來攻擊者所發動的攻擊行為。

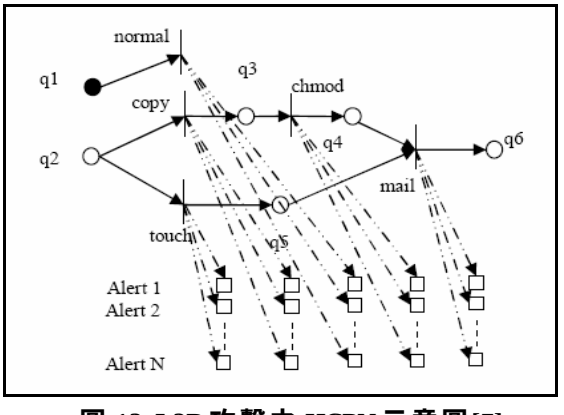

圖 13. L2R 攻擊之 HCPN 示意圖[7]

以 HCPN 架構為基礎的這套警訊關聯系統在輸出的結果上呈現給管理人員的資訊是已被入侵的資源（compromised resources），如圖 13 上被入侵的 q3到 q6，而不是原始的警訊資料，

表 7. HMM Classifier 與 Hidden CPN 之比較表

	HMM Classifier	Hidden CPN	Adaptive HMM
能否判斷攻擊是否成功？	無法判定	可判定	可判定
攻擊案例之建構是否可以模組化？	可以	不可以	可以
是否需要事先建立攻擊警訊間關係？	需要	需要	需要
攻擊之推論方式	整體觀點	個別觀點	整體觀點

如圖 13上的 Alert 1到 Alert N。好處是可以有效地減少回報的數量，幫助管理人員能即時做出因應對策。另外此系統除了呈現出被入侵的資源之外，也會利用 Dong Yu 等學者提出來的推論及學習演算法（Inference and Learning Algorithms, [7]）相對的計算出該資源被入侵的機率，以提供資訊安全人員另一項可判斷的依據。

回顧 Ourstone 與 Dong Yu 等人的研究後，我們發現 HMM 的分類器雖能分辨警訊序列屬於何種攻擊，但缺點是無法得知該警訊序列是否已完成攻擊，或僅屬無效的攻擊行為。Hidden CPN 技術可以分辨攻擊是否成功，還可以提供攻擊狀態或資源已被入侵的機率。但是 Hidden CPN 建構難度高，且缺乏結構性、相關性。而調適性隱藏式馬可夫模型可以分辨攻擊是否成功，並觀察整個攻擊事件全貌。對於不同的攻擊行為以機率分析發生的可能性，因此最適用於攻擊預測。三個系統的定性比較整理如表 7所示。

五、結論

由於攻擊手法的演進，資安人員難以由傳統的低階資安警訊瞭解攻擊者意圖及入侵行為全貌。然而，一個有效的多步驟攻擊，本身具有一定的程序，如果能夠由攻擊發起之初的蛛絲馬跡，預測攻擊者的目的地，將可以為資安管理人員爭取到寶貴的反制和預防的時間。但是攻擊者真正的意圖是隱藏的資訊，無法從表面的警訊資料直接看出。

本論文將多步驟攻擊情境視為有限狀態機(Finite-State Machine)，再利用彩色派翠網（CPN）和隱藏式馬可夫模型（HMM）結合建立攻擊行為狀態轉換的模型。再利用隱藏式馬可夫模型的機率理論，由 IDS 所觀察到的低階警訊推論攻擊者真正的狀態，進而預測攻擊者發動該多步驟攻擊行為的可能性。

隱藏式馬可夫模型是一個能夠加以訓練的模型，經由不斷的訓練過程，可以使得 HMM 的模型參數能夠愈加的收斂也愈加的趨近於攻擊者的攻擊特性，提升預測攻擊的準確性。

另外，本論文提出調適性隱藏式馬可夫模型（Adaptive HMM, AHMM）此一全新概念，針對目前 HMM 在多步驟攻擊上警訊處理的不足之處提出改進方案，考慮到攻擊步驟之間 AND 與 OR 邏輯關係，更接近真實的攻擊情境，增加預測機率的準確性。

最後本論文開發一套系統，並以 DARPA 2000 的資料進行實驗，證明能做到找出多步驟攻擊行為、預測攻擊、降低誤判率及漏判率等目的。也証明以調適性隱藏式馬可夫模型的預測能力。

關於本論文後續可能的研究方向，我們分為以下二點進行探討：

(1) 蒐集更多種類的多步驟攻擊情境及訓練資料

由於目前在本論文的系統當中只有五個 HMM 模型，尚無法充份發揮預測攻擊的能力，故若能再蒐集到更多的攻擊案例，則預測攻擊的效果將能更加的明顯。另外在訓練資料上若能取得愈多，則愈能使 HMM 的參數機率值更趨近於收斂值，也就能更符合攻擊的真實情況。

(2) 將本系統與 Generic 樣版的概念結合

由於目前的方式是將所蒐集到的各攻擊案例，先轉換為 CPN 樣版圖型，再轉換為 HMM 模型，缺點是只要遇到一個攻擊案例即需轉成一種 HMM 模型，較沒有模組化的概念，因此期望往後的研究可以朝向針對小的、基本的攻擊建立 Generic 樣版這樣的模型，而欲建立較複雜的多步驟攻擊時，即可利用基本攻擊模型組合成較大的多步驟攻擊模型。

六、參考文獻

[1] 翁興國，“資訊安全營運中心之事件關聯處理的根本問題分析”，2004 網際網路安全工程研討會論文集, 台北, 2004, pp.57-84

[2] 劉美君、陳奕明，“一種利用彩色派翠網關聯警訊以重建多步驟攻擊的方法”，第十四屆全國資訊安全會議論文集，台北，2004。

[3] 劉美君，一種利用彩色派翠網關聯警訊以重建多步驟攻擊的方法，國立中央大學資訊管理學系碩士論文，6月 2004。

[4] 官炳宏、陳奕明，“結合隱藏式馬可夫模型與彩色派翠網以關聯多步驟攻擊警訊之方法”，2005全國資訊安全會議論文集, 高雄, 2005.

[5] Sandeep Kummar, Eugene H. Spafford, “A Pattern Matching Model For Misuse Intrusion Detection”, In Proceedings of the 17th National Computer Security Conference, October 1994.

[6] Dirk Ourston, Sara Matzner, William Stump, Bryan Hopkins, “Applications of Hidden Markov Models to Detecting Multi-stage Network Attacks”, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS’03), 2003.

[7] Dong Yu, Deborah Frincke, “A Novel Framework for Alert Correlation and Understanding”, Proceedings of Applied Cryptography and Network Security, Second International Conference (ACNS 2004), Lecture Notes in Computer Science, Vol. 3089, 452-466, June 8-11 2004.

[8] Lawrence R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, Proceedings of the IEEE, VOL.77, No.2, February 1989.

[9] MIT Lincoln Lab, "2000 DARPA intrusion detection scenario specific datasets", http://www.ll.mit.edu/IST/ideval/data/2000/2000data_index.html

[10] Peng Ning,Yun Cui, “An Intrusion Alert Correlator Based on Prerequisites of Intrusions”, Technical Report, TR- 2002-01, North Carolina State University, Department of Compu- ter Science, 2002.

[11] Yuan Ho, Deborah Frincke, Donald Tobin, “Planning, Petri Nets, and Intrusion Detection”, In Proceedings of the 21st National Information Systems Security Conference (NISSC’98), 1998.

[12] Koral Ilgun, Richard A. Kemmerer and Phillip A. Porras, “State Transition Analysis: A Rule-Based Intrusion Detection Approach”, In Proceedings of IEEE Transactions on Software Engineering, 21(3), 1995.

[13] CERT/CC, “Overview Incident and Vulnerability Trends”, May 2003.

[14] CERT/CC, “CERT/CC Statistics 1988-2005”, 2005.

[15] Kristopher Daley, Ryan Larson, Jerald Dawkins, “A Structural Framework for Modeling Multi-Stage Network Attacks”, Proceedings of International Conference on Parallel Processing Workshop, 2002.

[16] Kurt Jensen, “Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Vol 1：Basic Concepts”, Monographs in Theoretical ComputerScience, Spring-Verlag, 1992.

[17] Peng Ning, Yun Cui, Douglas S. Reeves, “Constructing Attack Scenarios through Correlation of Intrusion Alerts”, In Proceedings of the 9th ACM Conference on Computer & Communications Security, pages 245--254, Washington D.C., November 2002.

[18] Dirk Ourston, Sara Matzner, William Stump, Bryan Hopkins, “Coordinated Internet attacks: responding to attack complexity”, Journal of Computer Security 12 (2004) 165-190, 2004

[19] Guy Helmer, Johnny Wong, Mark Slagell, Vasant Honavar, Les Miller, “Software Fault Tree and Colored Petri Net Based Specification, Design and Implementation of Agent-Based Intrusion Detection Systems”, ACM Transactions on Computer Security, Iowa State University, Department of Computer Science, 2001.

[20] Zoubin Ghahramani, “An Introduction to Hidden Markov Models and Bayesian Networks”, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 15, No. 1, 2001.

[21] Xinzhou Qin, Wenke Lee, “Attack Plan Recognition and Prediction Using Causal Networks.” ACSAC 2004: 370-379

[22] Jahmm Website, "A Java implementation of Hidden Markov Model related algorithm", http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/