結合 ModSecurity Core Rule 以提供 Snort 偵測網頁攻擊能力

沈宗享　　楊中皇
高雄師範大學　　高雄師範大學
資訊教育所研究生　　資訊教育所教授
zongshen@mail2000.com.tw　chyang@nknucc.edu.tw

摘要

在現今 Web 2.0 世代中，網路安全面對嚴峻的挑戰－網頁攻擊，企業所常見的防火牆與入侵偵測系統在面對這個新威脅時顯得力有未逮，由於網頁攻擊手法靈活，常使用多層次或多類型的編碼，而入侵偵測系統所使用的特徵比對方式，在處理這類手法，常無法正確解析出惡意的 HTTP Request，而開放原始碼的網頁應用程式防火牆 (Web Application Firewall) ModSecurity，其使用規則式分析引擎，能處理網頁攻擊手法中的 SQL Injection、Cross Site Scripting、Insecure Direct Object Reference 及 Cross Site Request Forgery，因此，本文提出運用 ModSecurity Core Rule 的架構，改寫 Snort 的預先處理器使其能支援該規則庫，藉以提供 Snort 在偵測網頁攻擊行為的能力。

關鍵詞：入侵偵測系統、網頁攻擊、網頁應用程式防火牆、正規表示式。

Abstract

The open source Intrusion Detection System "Snort" lacks the ability to detect Web Attack. Tradition tools uses Regular Expression to detect web attack but effect is limited. Because Hacker usually using multi-level or multi-type encoding attack to evade Intrusion Detection Systems. The open source Web Application Firewall "ModSecurity" could use Core Rule to detect SQL Injection, Cross Site Scripting, Insecure Direct Object Reference and Cross Site Request Forgery attack. The purpose of this paper is to use the ModSecurity Core Rule structure to provide web attack ability for Snort by implementing the web attack detection engine.

Keywords: Intrusion Detection System, Web attack, Web Application Firewall, Regular Expression.

1. 前言

現今的網路已與生活密不可分，無論是網路購物、金融交易，以及時下風行的線上遊戲產業，更是與現代人息息相關，然而，網路的方便性，因而也造成了某些人士的覬覦，像是時有所聞的線上遊戲虛擬寶物被盜、花旗銀行顧客資料外洩以及國內某銀行曾發生網路銀行被轉走存款的情事發生。因此，各類型的安全防護設施，應茲而生，像是部署於網路外層的 Honeypot （蜜罐），中層的 IDS （入侵偵測系統），以及內層的防火牆等等。

當網路系統遭受攻擊時，就需依賴各類型的網路設施、軟體來防禦，例如：使用入侵偵測系統來做網路行為的監控，並藉以了解攻擊者的入侵手法，讓系統管理者，得依據入侵偵測系統的報告，來修補系統漏洞，強化系統安全，不再遭受相同的攻擊。

而在現今 Web 2.0 的世代，各家網路服務公司，紛紛提供各項網頁服務，例如：網頁、相簿空間及 Blog 等服務，人人皆可在網路上暢談個人的意見，然而，亦造成了新型態的攻擊行為——網頁攻擊。

依據 Gartner 的資料指出，現行的網路攻擊行為，已從以往利用作業系統、應用程式的漏洞，或是通訊協定的弱點，轉向更容易進行的網頁攻擊，這些惡意的網頁活動占所有攻擊行為的 75%，然而，在現行的入侵偵測系統並無法有效偵測靈活的網頁攻擊行為，例如今年八月所散播的偽 CNN 惡意郵件，其結合了釣魚網站、FLASH 惡意程式及 BOT NET 等，而傳統的特徵式入侵偵測系統，使用正規表示式比對特徵值，在面對各類型靈活運用的網頁攻擊手法，入侵偵測系統顯得力不從心。開放原始碼的網頁應用程式防火牆 (Web Application Firewall) ModSecurity，其使用規則式分析引擎，相較於過去比對特徵值方式，ModSecurity 的規則式分析法更具彈性。近期的網頁攻擊手法靈活，常使用多層次或多類型的編碼迴避偵測，而入侵偵測系統所使用的特徵值比對方式，在處理這類攻擊手法，常無法正確解析出惡意的 HTTP Request，利用 ModSecurity 的

Core Rule 能有效偵測網頁攻擊手法中的 SQL Injection、Cross Site Scripting、Insecure Direct Object Reference 及 Cross Site Request Forgery，因此，本文提出運用 ModSecurity Core Rule 架構，改寫 Snort 的預先處理器使其能運用 Core Rule，藉以偵測上述的惡意網頁行為。

2. 文獻探討

2.1 IDS 的歷史

入侵偵測系統的起源於 1980 年代的美國政府與軍方單位 [7]，為了監控網路上違反安全行為的活動而設計，之後於 1990 年代中期才於市場上風行。

2.2 IDS 的設計哲學

入侵系統依其設計方式，可分為三種 [7,13]：

(1) 網路式入侵偵測系統 (Network-based intrusion detection system：NIDS)：網路式入侵偵測系統採取將網路上傳遞的封包，截取下來，再與內建的專家系統去做模式比對，因此，需要強大的運算功能，市面上的商用入侵偵測系統，幾乎皆是以硬體方式銷售。網路式入侵偵測系統的優點在於：
   - 容易部署：NIDS 採用被動式側聽網路上的活動，並加以分析。
   - 成本較低：在一個大型的環境中，僅需在數個較為敏感的地方，部署感應器 (Sensor)，即可監控大範圍的網路。
   - 偵測範圍：透過監控網路封包，並加以分析，即可偵測到異常的活動，相較於主機式，其偵測範圍更大。

(2) 主機式入侵偵測系統 (Host-based intrusion detection system：HIDS)：主機式入侵偵測系統被用來監控較為重要的主機系統，其會監控主機上的使用者及系統活動與攻擊行為，更進階的系統亦提供了政策稽核管理、存取控制、資料鑑識等功能。主機式網路式入侵偵測系統的優點在於：更詳細的日誌：其原理為監控該部主機的系統日誌，因此，其日誌分析原因較網路式更為詳細、監控效率更高。

(3) 網路節點式偵測系統 (Network-node intrusion diction system)：又稱為分散式偵測系統 (Distributed Intrusion Detection)，其運作方式與網路式相同，其不同點在於偵測系統會將監控的日誌，交由後端的統一管理主機分析，其適合更大型的網路環境。

而入侵偵測統依其偵測方式，又可細分三種 [7, 9]：

(1) 特徵型入侵偵測 (Signature-Based Intrusion Detection System)：特徵值是指由專家分析先前攻擊的訊息中所建構的，其原理是利用特徵值來與封包做比對，其優點為：能偵測已知的攻擊、低誤警率 (low false alarms) 及效率較高 （相對於異常型）；其缺點為：無法偵測未知的攻擊，需時常更新特徵資料庫。

(2) 異常型入侵偵測 (Anomaly-Based Intrusion Detection System) ：異常型是依內建的正常通訊模式來做判斷，而違反該模式者，則判定為異常，其優點為：能偵測未知的攻擊手法；而缺點為：誤報率較高、效率較低。

(3) 混合型入侵偵測 (Hybrid Intrusion Detection System) ：為特徵型及異常型的特色融合而成，是目前的趨勢。

2.3 網頁攻擊

依據 Gartner 所作的研究報告指出，在他們所檢測三百多個網站中，發現有 97% 的網站有安全上的弱點，且有 75% 的攻擊目前是在應用程式層級，因此，在 Web 2.0 世代中，所遭遇到的重大挑戰——網頁攻擊，成為一項大議題。

在開放 Web 軟體安全計劃組織 (OWASP) 所提出的 10 大網頁攻擊手法中 [15]，與程式撰寫相關的安全弱點如下：

(1) Cross Site Scripting [8]：利用系統未對所輸入的內容做檢查的弱點，使得駭客得以植入可執行的惡意程式，其後果可能造成被植入木馬程式或導向釣魚網站。

(2) Injection Flaw：常見有 SQL Injection 在提供給使用者輸入的表單內，未予以檢查，使得駭客得以輸入原先無權限的執行的程式或資料庫，甚至在某些系統環境下，可獲取系統管理者權限，為網頁掛馬常使用的手法。

(3) Malicious File Execution：Web 系統設計上的缺陷，使得系統可執行遠端的惡意程

式，大多發生於 PHP 上。

(4) Insecure Direct Object Reference：由於系統的讀取檔案功能設計漏洞，使得駭客得以讀取系統內任意路徑下的檔案，其威脅在於可能因此外洩系統重要的系統檔案，如：密碼檔，造成密碼被破解，進而取得系統權限。

(5) Cross Site Request Forgery：為 Cross Site Scripting 的延伸攻擊手法，網站內植入惡意程式碼後，讓合法的使用者在不自覺的情況下，自動執行該惡意程式，藉以利用合法使用者的身份，獲得權限。

為何網頁攻擊如此難以偵測 [8]？當弱點存在於網頁應用程式中，駭客所進行的攻擊，其所使用的皆是正常的 Http Request（儘管其內含惡意程式），能滲透入防火牆、避開入侵偵測系統，而不會發出警訊 (Alert)；且現在攻擊方式常會經過多樣、甚至多層的編碼，以避開入侵偵測系統，因此有相關偵測方法有：使用統計法 [6,10]、決策樹法 [9]、攻擊圖法 [12]、分類法 [16] 等。

3. 系統設計與實作

3.1 Snort

Snort 為 Marty Roesch 在 1998 年時 [2,5,17]，所發展出的一套輕量級的入侵偵測系統，其特色為：運算效率良好且開放原始碼，為目前開放式平台中最熱門的入侵偵測系統。在特徵式入侵偵測系統中，其最重要的議題是需時常更新其特徵資料庫，而 Snort 的特徵資料庫更新頻繁，另外，Snort 設計了一套規則語言，若使用者熟悉該套語言，則可自行建構所需要的特徵檔。

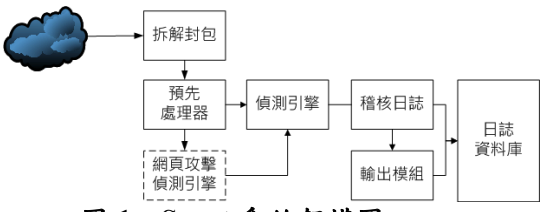

圖 1 Snort 系統架構圖

(1) 拆解封包

Snort 底層利用 PCAP 函式庫來擷取網路上傳送／接收的封包，擷取內容含蓋了擷取的時間、封包長度、連結類型（如：Ethernet, FDDI 等等），並建立一個可直接指向該封包的指標，如此，可加速 Snort 分析封包的速度。另若以 inline 模式運行時，還加上封包傳送、變更封包內容、拒絕某些特定封包及將封包丟棄的功能等防火牆功能。

(2) 預先處理器

在擷取到封包後，Snort 會將封包傳送到預先處理器，進行封包重組，以及依各協定的封包格式進行正規化，預先處理器還可作到封包流量的統計分析，以及非規則式攻擊偵測，如：阻斷式攻擊及蠕蟲攻擊等。

(3) 偵測引擎

偵測引擎為 Snort 的核心，使用者可從其官方網站下載所釋出的特徵資料庫，搭配適當的設定，可以有效偵測網路攻擊，當系統將所擷取到的封包與特徵規則比對符合者，將產生攻擊警訊送至稽核日誌。

(4) 稽核日誌

當系統判定攻擊時，即會將當時攻擊的相關資訊及所判定的攻擊行為，產生日誌及警訊，讓系統管理者得以排除攻擊行為，其有兩個機制，可對警訊做較好的輸出。事件佇列 (Event Queue) 及門檻值 (Thresholds) 設定，當一項攻擊行為發生因而觸發多條規則，則可使用事件佇列將規則定義優先順序，再依序產生警訊，如此，系統管理者得以排除較不緊急的攻擊行為；另一機制為訂定門檻值，此功能可用在當在短暫時間內發出大量的相同攻擊行為，僅發出一次警訊，例如：當阻斷式攻擊或蠕蟲發生時，會大量發出警訊，在此情況下，則可設定每 60 秒內，在相同 IP，相同的攻擊行為，僅發出一次警訊，如此可大量減少警訊的數量，以簡化系統管理者在追蹤來源的複雜度。

(5) 輸出模組

Snort 支援下列多項輸出模組，使用者可依環境，自訂輸出的目的地。

- Default Logging
- SNMP traps
- XML Logging
- Syslog
- SMB Alerting

- PCAP logging
- SnortDb
- Unified Log

3.2 ModSecurity

ModSecurity 是一開放原始碼網頁應用程式防火牆 (Web Application Firewall) [4, 14]，是由 Ivan Ristic 所開發，其使用規則式偵測引擎，較正規表示式偵測法更有彈性，在偵測各類型的 Web 攻擊，如：SQL Injection、Cross Site Scripting、Insecure Direct Object Reference 及 Cross Site Request Forgery 有良好效果。

3.3 網頁攻擊偵測引擎實作

Snort 的特徵資料是以型別及關鍵字組成，再透過 PCRE 函式庫以正規表示式來進行模式比對 (Pattern Matching) ，其特色是對特定攻擊特徵比對效率好，但對靈活多變的網頁攻擊則效果較差，因此，本文改寫 Snort 的預先處理器 Http Inspect 作為網頁攻擊偵測引擎，使其能引用 ModSecurity 內建的 Core Rule，來偵測網頁攻擊行為。本文以 Snort 預先處理器 Http Inspect (hiclient.c:1882) 為基礎，結合 ModSecurity Core Rule 提供頁攻擊的偵測能力，在 Windows XP 環境下，以 GCC 3.4.5 (MinGW 5.1) 開發，並實作下列說明之功能：

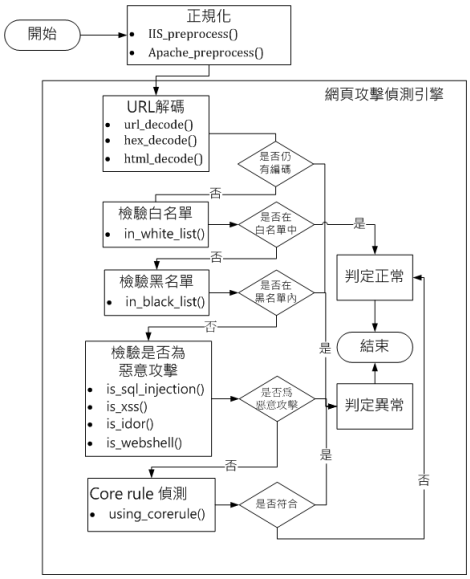

圖 2 系統流程圖

(1) 正規化

由於本文係採用網頁伺服器日誌，作為輸入資料，而 Snort 並無法直接分析該日誌，因此，採用替代方式，以離線方式，將所蒐集的網頁伺服器日誌，依 Apache/IIS 的格式，預先處理，取出完整 URL 作為資料，再輸入所實作的網頁攻擊偵測引擎加以判別。

(2) URL 解碼

由於瀏覽器在送出 HTTP Request 前，會先將部份符號及中文字加以編碼，所以在做檢驗前應先進行解碼的動作，如此可降低偵測系統誤判的機率。而另一方面，攻擊者為避免攻擊行為遭入侵系統偵測到，會利用各類型的編碼法，將攻擊網頁的指令，加上編碼，期藉以避開偵測，而本功能會將所含 URL 編碼予以解碼，若仍有經編碼過的 URL，即判斷為惡意攻擊。

(3) 檢驗白名單

各類型的網頁程式環境，皆有其特殊的模式，以及各大搜尋引擎，如：Google、Yahoo、Msn 及 Baidu 等等的搜尋機器人，可先加入白名單內，可加快檢驗的速度，以及是屬於正常的 Http Request ，但與惡意攻擊特徵相似，亦需建置於內。

(4) 檢驗黑名單

檢查所收集惡意連結作為黑名單 [1,3]，由於該些惡意連結常是以正常檔案的方式呈現，如: js、css 等文字檔或是內含 shell code 的惡意二進位檔如： gif、ppt、pdf 等等，而無法單純以 HTTP Request 做為判斷依據，需另加以分析後觀測其行為始能驗證，因此，本研究引用網路論壇所發佈的清查清單作為檢驗黑名單。以已知的惡意連結作為黑名單，若所傳送內容裡，內含黑名單內所知的連結時，即可判定是攻擊行為，其優點是誤判率低，但無法主動偵測未知惡意網站。

(5) 檢驗是否為惡意攻擊

本功能主要實作 OWASP 2007 所提出的十大網頁弱點中的最嚴重的 SQL Injection、Cross Site Scripting、Insecure Direct Object Reference 及 Cross Site Request Forgery 等攻擊的辨識能力。如：防止上傳網頁後門：當發生網頁攻擊時，其中的一大威脅即為系統遭受

上傳網頁後門 (WebShell) 攻擊，此功能以使用者行為作判斷，當從使用者所上傳的資料內含有網頁程式時 (ASP/PHP/JSP 等等)，即可判定是攻擊行為，如表 1 所示。

表 1　攻擊關鍵字表

功能	相關關鍵字
is_sql_injection()	or , --' , '--, and , exec , select , insert , update , delete , drop , where , dbo, cast(, char(, union
is_xss()	javascript , <scrip, /script>, document.write, document.cookie , url(, eval(, expression(, <object, onload, onmouseover , onerror, , windows.open, <iframe, function, .location, (',);
is_idor()	.ini , ./. , , \\.. , .. , .swf? , boot.ini , etc/ , /passwd
is_webshell()	<form, <%, <?, <php, %>, ?> , php> , action

(6) Core Rule 偵測

以 ModSecurity 的 Core Rule 作為最後一階段的攻擊偵測，本文實作了可引用 Core Rule 規則庫的偵測引擎，讓系統可運用另一攻擊資料庫來提高偵測率。

4. 實驗數據

傳統入侵偵測系統常使用 DARPA Dataset 作為入侵偵測測試資料集，但其已久未更新（至 2000 年），因此，本文以某醫學中心入口網站日誌共一百五十萬餘筆記錄作為資料來源，並以離線方式分析，流程如下：經分析六日日誌，經分類統計如表 2 所示。

表 2　攻擊分析表

日期	正常	SQL Injection	XSS	Insecure Direct Object Reference	Web Shell	Encoding	總計
DAY1	522,805	344	55	11,187	2	0	534,393
DAY2	137,803	448	95	4,901	0	0	143,247
DAY3	357,922	281	102	568	35	0	358,908
DAY4	126,123	46	34	2	0	0	126,205
DAY5	199,069	98	71	0	3	0	199,241
DAY6	145,509	16	10	1	0	0	145,536
總計	1,489,231	1,233	367	16,659	40	0	1,507,530
比率	98.79%	0.18%	0.02%	1.11%	0.0%	0%	

另以 XSS Attacks - Cross site scripting exploits and defense [8] 所提供的惡意連結共 315 筆記錄，偵測引擎所分析所得統計如表 3 所示。

表 3　已知攻擊分析表

	正常	SQL Injection	XSS	Insecure Direct Object Reference	Web Shell	Encoding	總計
總計	0	0	312	3	0	0	315
比率	0%	0%	99.05%	0.95%	0%	0%	

5. 結論

本文為強化 Snort 在偵測網頁攻擊的不足，以內建的 Http Inspect 預先處理器為基礎，實作網頁攻擊偵測引擎，除了可偵測常見 SQL Injection、Cross Site Scripting、Insecure Direct Object Reference 及 Cross Site Request Forgery 外，對於網頁後門、檢查變型的編碼，及惡意連結黑名單的偵測上亦有良好的效果。此外，本文亦實作了能支援 Core Rule 規則偵測引擎功能，提供了 Snort 另一套攻擊資料庫，可偵測多變化的攻擊手法。

參考文獻

[1] 大砲開講，http://rogerspeaking.blogspot.com/。

[2] 楊中皇，***網路安全理論與實務***，金禾資訊，2006。

[3] 資安之眼，http://www.itis.tw/。

[4] Barnett, R.C., ***Preventing Web Attacks with Apache,*** Addison Wesley, 2006

[5] Caswell, B., Beale, J. and Baker, A.R., ***Snort IDS and IPS Toolkit***, Syngress, 2007.

[6] Cheng, Y.C., Laih, C.S., Lai, G.H., Chen, C.M. and Chen, T., "Defending On-Line Web Application Security with User-Behavior Surveillance," ***Availability, Reliability and Security,*** pp. 410-415, 2008.

[7] Crothers, T., ***Implementing Intrusion Detection Systems***, Willey, 2002.

[8] Grpssman, J., Hansen, R., Petkov, P.D., Roger, A. and Fogie, S., ***XSS Attacks - Cross site scripting exploits and defense***, Syngress, 2007.

[9] Hwang, K., Cai, M., Chen, Y. and Qin, M., "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes," ***Dependable and Secure Computing***, Vol. 4, no. 1, pp. 41-55, 2007.

[10] Kiani, M., Clark, A. and Mohay, G., "Evaluation of Anomaly Based Character Distribution Models in the Detection of SQL Injection Attacks," ***Availability, Reliability and Security,*** pp. 47-55, 2008.

[11] Lee, J.H., Lee, J.H. Sohn, S.G., Ryu, J.H.

and Chung, T.M., "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System," ***Advanced Communication Technology,*** Vol. 2, pp. 1170-1175, 2008.

[12] Lei, J. and Li, Z.T., "Using Network Attack Graph to Predict the Future Attacks," ***Communications and Networking in China,*** pp. 403-407, 2007.

[13] McClure, S., Scambray, J. and Kurtz, G. Hacking exposed - network security secrets & solutions, 4th ed., McGraw Hill, 2004.

[14] ModSecurity, http://www.modsecurity.org/.

[15] OWASP, http://www.owasp.org/images/e/e8/OWASP_Top_10_2007.pdf.

[16] Seo, J., Kim, H.S. Cho, S. and Cha, S., "Web server attack categorization based on root causes and their locations," ***Information Technology: Coding and Computing,*** Vol. 91, pp. 90-96, 2004.

[17] Snort, http://www.snort.org/.