攻擊、病毒與廣告信的辨識機制與套件

王聲浩 陳一瑋 林盈達

國立交通大學資訊工程系

E-mail:howz.cs97g@nctu.edu.tw,iwchen@nbl.org.tw,ydlin@cs.nctu.edu.tw

October 20, 2008

摘要

在網路流量愈來越大的趨勢下，如何從中獲得有用的資訊與辨識出具有危險的內容，形成一門重要的研究課題，目前主流的辨識方式，是經由特徵值(Signature-Based)的設計，搭配正規表示式(Regular Expression)透過字串比對，來達到流量封包辨識的目的。本篇內容介紹三種不同的辨識工具，Snort 以特徵規則檔分析表頭與比對內容，並使用 56%的 PCRE 正規表示式，加強特徵值比對。ClamAV 98%的特徵值以十六進制、MD5 編碼，與壓縮檔案庫方式進行比對，2%的特徵值採 POSIX 正規表示式用於辨識網路釣魚。SpamAssassin 將郵件與資料庫中的測試檔比對相似度來評分，透過所得分數加總，檢查得分是否到達垃圾郵件門檻值，測試檔中有 76%使用 Perl 正規表示式。特徵值設計的好壞決定是否產生誤擋的情況，Snort 規則檔中的 content、pcre 欄位內容，ClamAV 用來產生特徵碼所採用的惡意程式樣本，及 SpamAssassin 測試檔 Perl 正規表示式描述的技巧，直接影響誤擋發生與否。

關鍵詞: 封包辨識、內容過濾、特徵值(signature)、正規表示式(regular expression)

1. 簡介

隨著網際網路快速的發展，資訊網路安全已不再只侷限於傳統的病毒/防毒上，而 TCP/IP 網路共同協定的制定，無疑地，替網路攻擊者提供更多的攻擊途徑，許多新型態的攻擊方式與技巧層出不窮，如:病毒、蠕蟲、惡意軟體、垃圾郵件、網路釣魚，近年來使用動態連接埠(dynamic-port)的網路應用程式的出現，使傳統上透過分析 OSI Layer 3(Network layer)、Layer 4(Transport layer)封包表頭(Header)的辨識方式更顯不足，因此要精確的辨識出網路流量中的內容，勢必要透過分析 Layer 7(Application layer)的封包內容(Payload)才能達到有效辨識。

目前許多網路安全問題逐漸在第七層出現，很多的惡意程式都夾帶在應用層中。第一類網路攻擊(Attack)如: 阻斷式攻擊(Dos)、緩衝溢位(Buffer overflow)、連接埠掃描(Port scan)，其常見防範的機制是入侵偵測系統 NIDS (Network Intrusion

Detection System)，在本文以 Snort 此工具去探討辨識攻擊的機制。第二類惡意軟體(Malware)如: 病毒(Virus)、蠕蟲(Worm)、木馬(Trojan horse)，在此以防毒軟體 ClamAV 來探討辨識惡意軟體的機制。第三類垃圾郵件(Spam)如:廣告郵件、網路釣魚(Phishing)、詐騙(Scam)，主要採用郵件過濾器 SpamAssassin 去瞭解該辨識機制，上述三種工具皆是在 Linux 系統下運作的開源碼(Open Source)套件，可以經由網路下載自由使用，接下來將會介紹各辨識工具的架構與特徵值設計。

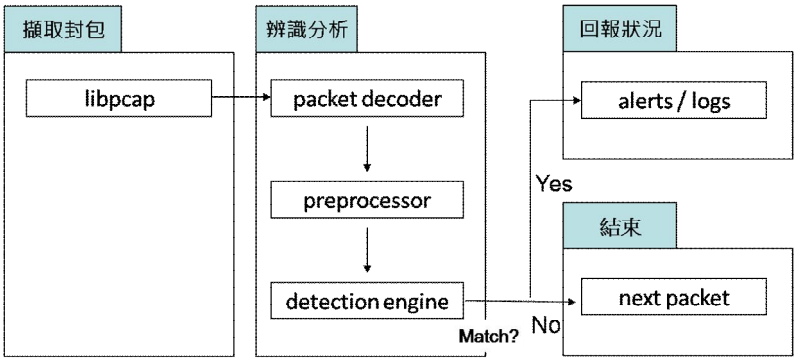
2. 系統架構

在本文中主要透過使用開源碼工具的方式，去實際瞭解辨識機制的設計與差異，因此必須對於使用的工具有一定的熟悉度，由【表一】可以得知，辨識機制上，Snort 跟 ClamAV 皆是使用字串比對(pattern matching)的方式去比對建立好的偵測規則檔與特徵碼，而 SpamAssassin 則是使用知名的 Bayesian 貝氏過濾法，透過分析郵件每一個代符(token)建立一個具有評分標準的測試檔；自動更新部分，ClamAV 內建 freshclam 達到自動更新，Snort 則需透過外掛模組 Oinkmaster 才能有自動更新的功能，而 SpamAssassin，因具有自動學習的能力，且提供分散式特徵資料庫，較不依賴自動更新，自動更新主要是讓特徵資料庫保持在最新的狀態，否則很容易受新型態的攻擊，產生漏擋(False Negative)的情形；Snort 所能偵測的通訊協定目前只提供 TCP、IP、ICMP、UDP 四種協定，SpamAssassin 主要用於掃描電子郵件，因此其支援以郵件為主的通訊協定；演算法上，Snort 與 ClamAV 運用類似的演算方式，SpamAssassin 採用基因演算法(Genetic algorithm)，尋找最適合用來檢驗垃圾郵件的代符。接下來將深入介紹各工具的系統架構與運作流程。

	Snort	Clam AV	SpamAssassin
類別	Attack	Malware	Spam
開發程式	C	C	Perl
開源碼授權	GPL	GPL	Apache
辨識機制	Multi-pattern matching	Multi-pattern matching	Bayesian filtering
辨識策略	Pattern file	Pattern file	Token file
自動更新	Option	Yes	No
支援協定	TCP IP ICMP UDP	IMAP POP3 SMTP HTTP FTP	IMAP POP3 SMTP HTTP
演算法	Aho-Corasick Wu-Manber Boyer-Moore	Aho-Corasick Boyer-Moore	Genetic algorithm

表一：三種工具特色的比較

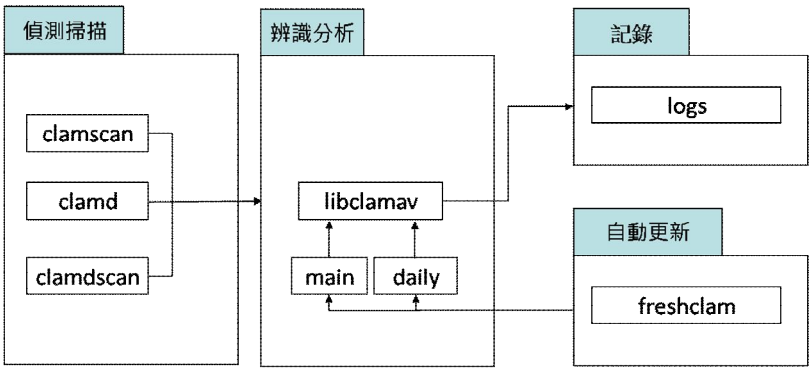
A. Snort



圖一：Snort系統架構圖

Snort [1]是一個相當著名的輕量級入侵偵測軟體，其系統架構如【圖一】，Snort 採用 Libpcap 函式庫來擷取封包，接收到封包後，傳送到 packet decoder 中將封包重組整理成適合用來分析的格式，之後以模組化的 preprocessor，加強入侵偵測與防護的能力，如 preprocessor frag2 提供 IP 碎片重組及偵測碎片攻擊；preprocessor stream4 偵測連接埠掃描及阻止阻斷式攻擊，入侵偵測引擎(detection engine)中，透過 detection plugin 的方式，將規則檔中所使用到的 plugin 皆預先載入引擎中，之後的字串比對便以其為基礎，如 msg 為記錄訊息至 log file 中；tos 為檢查 IP 表頭中 type of service 欄位等，Snort 比對偵測引擎中的入侵規則資料庫後，若有比對到則會發出 alert 並寫入 log 檔中，否則即讓其通過並對下一個封包進行比對，Snort 在 2.3.0 版後的 inline mode 加入了即時阻擋攻擊的功能，使 Snort 成為 IPS(Intrusion Prevention System)。Snort 可以在組態檔中設定要載入哪些規則檔如：dos.rules、scan.rules、backdoor.rules 等，其組態檔一般放於/etc/snort/snort.conf 下。

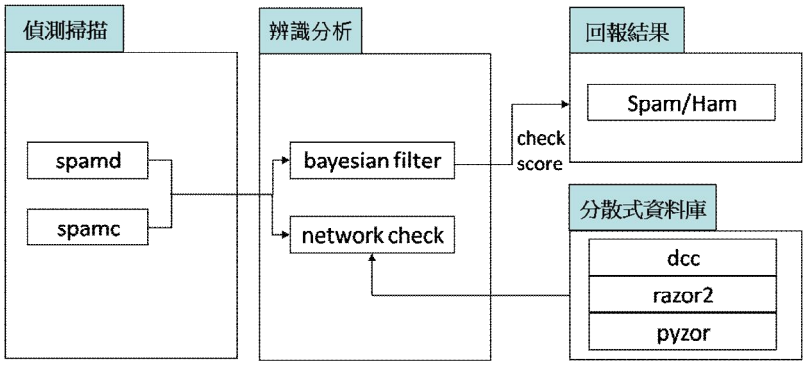
B. ClamAV



圖二：ClamAV系統架構圖

ClamAV[2]常被用來當作防毒程式，主要的功能是用於掃描出電子郵件的附件病毒檔，其系統架構如【圖二】，ClamAV 使用 Clamd 作為監聽本地連線(如 Unix socket)的防毒常駐程式，也是監聽網路連線(如 TCP socket)的防毒伺服器，以 libclamav 作為病毒偵測引擎，其中包含兩個檔案組成病毒資料庫，一個是 main.cvd 包含大多數的病毒定義，另一個 daily.cvd 則包含最新的病毒定義，透過 freshclam 可以定期自動更新病毒定義檔，保持其掃毒能力。ClamAV 提供 sigtool 工具，給使用者用於建立病毒特徵碼與查看病毒資料庫，如：sigtool –unpack-current daily.cvd 解開病毒資料庫，顯示病毒特徵碼，Sigtool –hex-dump 將惡意軟體建立成十六進制的病毒特徵，其組態檔一般放於 /etc/clamav/clamd.conf。

C. SpamAssassin



圖三：SpamAssassin系統架構圖

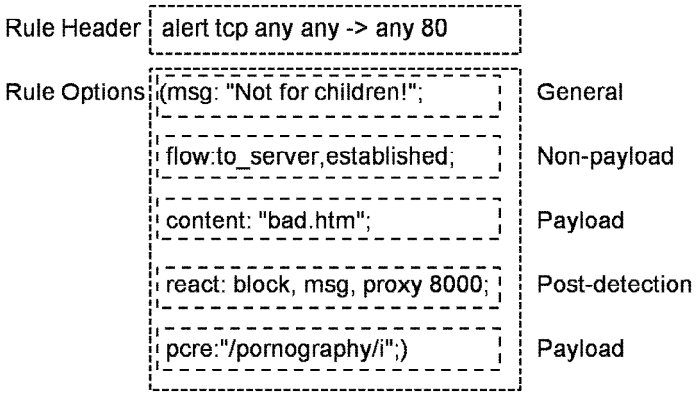
SpamAssassin[3]是利用 Perl 來對信件中的每一個代符作分析，以達到過濾垃圾郵件的目的，其系統架構如【圖三】，使用兩種用於分析評分的機制，一種為內建自動學習引擎的貝氏過濾法[4]，透過訓練的機制將非廣告的郵件(Ham)放進 hashtable_good，廣告郵件(Spam)放入 hashtable_bad，統計某代符出現時為垃圾郵件的機率有多少，進而產生評分的標準，SpamAssassin 提供三個分散式垃圾郵件特徵資料庫，分別為 Razor2、Pyzor 以及 DCC(Distributed Checksum Clearinghouse)用於網路測試模式，其判斷運作的方式是透過比對每一封信的標頭(header)、內文(body)，若符合某種特徵則加以評分，最後進行分數加總，若總分超過制定的門檻值(threshold score)，則視為垃圾郵件，SpamAssassin 有三種處理垃圾郵件的方式，可於組態檔的 report_safe <number>參數中設定，0:將資訊寫入郵件表頭；1:轉為附件 2:轉為純文字附件，組態檔一般於/etc/mail/spamassassin/local.cf。

3. 特徵值設計與比較

在瞭解過各個套件的運作情形後，要進一步知道各套件用於辨識流量封包的特徵值表示方式，在此先針對三個工具特徵值上共同的差異作比較，由【表二】可以觀察到幾個差異，在正規表示式上，由於 ClamAV 所支援的核心為 POSIX，其可用的運算符號不若 Perl 相容性來的多；而在特徵值的設計上，Snort 所使用的 Rule 規則，具有較多樣化的字串分析比對方式，使用者可自訂程度較靈活，SpamAssassin 則因其開發程式為 Perl，在特徵值的設計上大量使用正規表示式測試郵件，取代一般的簡單字串(Simple Pattern)，使其特徵值的個數相對減少許多，ClamAV 的每一個特徵碼只相對應到一個惡意軟體的特徵，造成特徵資料庫相當龐大的負擔；SpamAssassin 的貝氏過濾設計，提供一個具有自動學習能力的特徵值，可不斷改進辨識的能力，不像 Snort 與 ClamAV 則需經由人力去新增修改特徵值至資料庫中，但此方法的缺點是需要使用大量已知的垃圾郵件與非垃圾郵件去訓練過濾引擎，因此取樣的樣本會直接影響過濾的準確度；白名單與黑名單主要用來排除一些已知的誤擋(False Positive)與漏擋(False Negtive)，接下來將探討各特徵值設計上的特點。

	Snort	Clam AV	Spam-Assassin
版本	2.7.0	0.92.1	3.2.4
特徵值設計	Rule	Hexadecimal / MD5	Tests
正規表示式	Pcre	Posix	Perl
特徵值數量	3,382	415,980	689
資料庫	MySQL PostGreSQL	ClamAV Virus Databases	MySQL PostGreSQL
自動學習	No	No	Yes
白名單	No	Yes	Yes
黑名單	No	No	Yes
客製化	Yes	Not all	Yes

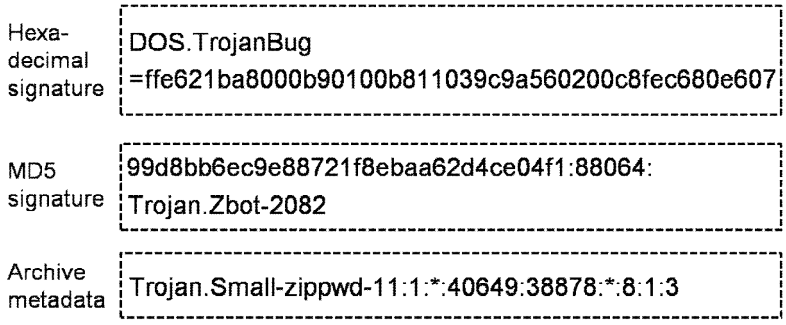
表二：三種工具特徵值的比較



圖四：Snort特徵規則檔表示式

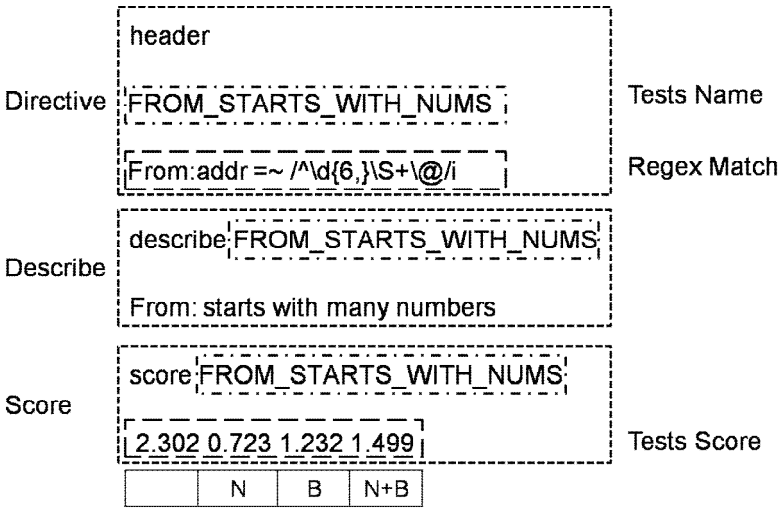
Snort 是一個基於攻擊特徵的 NIDS，因此其偵測攻擊的能力取決於攻擊特徵資料的質與量，Snort 的病毒特徵碼皆描述在.rules 檔中，【圖四】顯示基本 Snort 攻擊特徵的格式，rule header 表示監控 TCP 中任一來源 IP 和 Port 到達任一目的 IP 且 Port 為 80 的封包，方向指標可設成雙向偵測< >，前述的 4-tuple 採用變數

可在組態檔中修改，如 any 代表所有網段，HOME_NET 代表內部網路，可以設成單一 IP 或網段，EXTERNAL_NET 代表外部網路可以設定成!$HOME_NET，代表除了內部網路都屬於外部網路，在 rule options 的部分，則表示辨識封包中包含 bad.htm 與 pornography 字樣，同時建立 TCP 連線，請求 client 回應並透過 proxy 8000 傳送 msg:Not for children! 然後中斷此連線,若條件符合則執行 alert 警告動作並寫入 log 檔中，rule options 中最重要的語法為 content 與 pcre，是字串比對準確與否的關鍵，Snort 的 rule options 分成四大類，其更進階的功能請參考[5]。



圖五：ClamAV病毒特徵值表示式

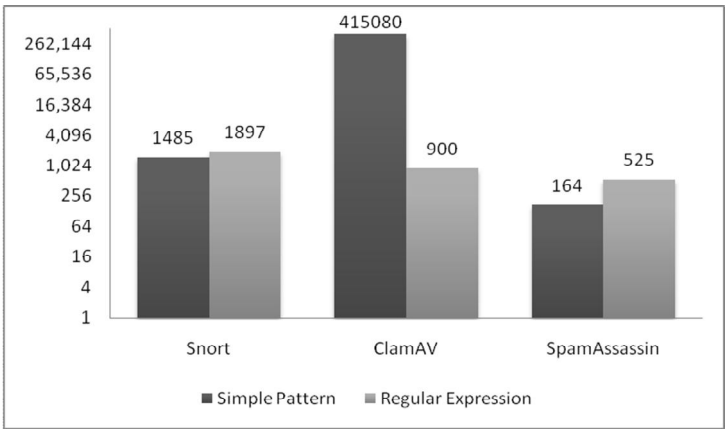
ClamAV 的病毒特徵使用三種方式[6]，如【圖五】所示，第一種為十六進制的病毒特徵，可以透過 sigtool --hex-dump 此命令建立，產生出的十六進制碼給予惡意軟體名稱，存放在.db 的檔案中，ClamAV 可透過此檔以十六進制的格式在流量中找尋惡意軟體的獨特字串，第二種為 MD5 checksum 格式，使用 sigtool --md5 產生 MD5 病毒特徵，其格式為 MD5:檔案大小(Bytes):惡意軟體名稱，須存放在.hdb 中，在使用選擇上，若惡意軟體被嵌入完全相同的檔案，此常見於電子郵件附件當中，則可用 MD5 簽章，而像是蠕蟲(worm)這種會改變或複製自己到不同檔案的惡意軟體，較適合用十六進制的特徵值辨識，第三種為透過 metadata 檔案庫進行比對，辨識出經過 Zip 或 Rar 壓縮過的惡意程式，其格式為惡意軟體名稱：是否經過壓縮：原檔案名稱：未經壓縮的檔案大小：經壓縮的檔案大小：crc32 校驗碼：壓縮方式：檔案編號：相關檔案數，其特徵碼存放於.zmd 與.rmd 中。



圖六：SpamAssassin特徵測試檔表示式

SpamAssassin 大部分的特徵測試檔分成三個部分如【圖六】，指令(directive)部分，可分為比對標頭或內文，此例是比對在@前超過六個數字為開頭的寄件者郵件位址，由於採用 perl 正規表示式的關係，因此為了可讀性，每一個特徵測試檔皆會有一個描述(describe)讓使用者知道該測試的內容，得分方式則經由測試名稱(Tests Name)到 50_scores.cf 這個檔案中，去尋找同名稱的測試分數，查詢到 2.302 0.723 1.232 1.499 四種分數，分別為經驗法則測試(heuristic test)、啟用網路測試、啟用貝氏過濾，以及兩者同時啟用，至於怎樣判別為垃圾郵件，在組態檔中有一個 require_score x.y 可自訂將其歸類為垃圾郵件的門檻值，Spamassasssin 的病毒特徵檔皆以.cf 檔案存放/usr/share/spamassassin 下。

正規表示式種類



圖七：特徵值表示中 Simple Pattern 與 Regular Expression 所佔的數量

正規表示式能提供非常有效率的方式來辨識字串，如之前所提到，一個有效率的正規表示式能夠取代多個簡單字串，然而如何去撰寫一個有效率的正規表示

式，將會是個難題，若設計不當很容易對系統效能造成極大的影響，【圖七】顯示正規表示式在每一個工具特徵值中所佔的比例，Snort 約 56%使用 PCRE 加強比對；ClamAV 除了前述的三種特徵值外，從 0.9X 版後增加可用 POSIX 正規表示式來捕捉網路釣魚的特徵值設計[7]，其所占的比例並不多約 2%；SpamAssassin 則使用約 76%大量的 Perl 正規表示式於特徵值設計上，以下是三個個別範例:

Snort : PCRE

```
alert ip any any -> any any
(pcre:"/%[0-9.]/*&?(i|n|x|s|h|e|g|a){3}/";msg:"Format string";)
```

此例是偵測 IP 封包裡是否有格式字串，可測出如: %213.2n%x%334.123$h 與 %n%n%n 之類的攻擊。

ClamAV : POSIX

```
R:.+\.welcome\.(at|ca|co\.cc|co\.jp|fr|tw)([/?].*)?:.+\.welcome\.com([/?].*)?
```

此例將此表示式，產生的組合所比對到的網址，皆視為網路釣魚網站。

SpamAssassin : Perl

```
body ASCII_FORM_ENTRY [^<][A-Za-z][A-Za-z]+.{1,15}?[\x09\x20]*_{30,}
describe ASCII_FORM_ENTRY Contains an ASCII-formatted form
```

此例測試電子郵件本文中是否含有 ASCII 格式型態。

因正規表示式較為進階，在此只解釋如何用於自訂特徵值，若讀者有興趣可以至[8]作更深入的學習。

4. 誤擋個案探討

誤擋(False Positives)是指將非惡意行為判定為惡意行為，導致使用者得不到此訊息，發生的原因可以是網路結構環境不良，軟體本身的 Bug，以及本文所探討的特徵值設計上的缺失，以下將舉三個誤擋的個案，分析其產生的原因。

Snort 會產生誤擋的情形，主要在於之前所提到 content 與 pcre 上的描述是否恰當，以如下用於偵測 IIS 5.0 所產生的漏洞，辨識 URL 中表頭包含'Translate: f'試圖取得檔案程式碼的攻擊所設計的規則檔為例：

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:
"WEB-IIS view source via translate header"; flow:to_server,established;
content:"Translate|3A| F"; nocase; reference:cve,2000-0778; sid:1042;)
```

此例中的 content 內容主要比對"Translate|3A| F"是否有出現，然而有些微軟的應用軟體中也有包含這個表頭，如 Microsoft Outlook Web Access(OWA)，因此會造成錯誤的警告。

ClamAV 其誤擋產生的原因在於特徵碼取樣是否夠嚴謹，以如下用於防止受蠕蟲感染或偽裝的 iexplore.exe 在系統中不斷複製，造成系統資源被占用的 MD5 特徵碼為例：

ecd35d17f66899882b9558f5b94c5798:93184:iexplore.exe

此例直接對 iexplore.exe 程序取其特徵碼，然而 iexplore.exe 同時也是 Microsoft Internet Explorer 的主程序，這樣的表示會造成誤將系統檔案當成病毒的問題。

SpamAssassin 的誤擋發生主要集中在 Perl 正規表示式的設計方式上，以如下測試檔用於過濾發送來源為 Outlook 使用端，且非 perl 正規表示式描述允許的信件識別碼(Message-id)郵件為例：

```
# Outlook IMO (Internet Mail Only)
header __OIMO_MUA    X-Mailer =~ /Outlook IMO/
header __OIMO_MSGID MESSAGEID =~ /^<[A-P]{26}A[ABC]\.[-\w.]+\@\S+>$/m
meta FORGED_MUA_OIMO   (__OIMO_MUA && !__OIMO_MSGID &&
!__OE_MSGID_2 && !__UNUSABLE_MSGID)
```

此例是採用刪去的方式將非範圍內的特徵值皆視為垃圾郵件，在 meta 中若 __OIMO_MSGID、__OE_MSGID_2 或__UNUSABLE_MSGID 這三個測試檔中有一個比對為 False，則此測試檔會被視為垃圾郵件，然而這三個測試檔的 perl 正規表示式描述仍不夠廣，因此會造成某些 MTA(Mail Transfer Agent)所產生的信件識別碼被視為垃圾郵件。

5. 結論與未來發展

隨著網路攻擊手法越來越多變，將辨識流量封包的層次拉到應用層，已是目前主流趨勢，然而同時也造成辨識的機制必須越趨複雜，Pattern Matching 與 Bayesian filter 提供兩個在此層辨識不同目標的機制。Snort、ClamAV、SpamAssassin 依其設定目標，分別採用入侵規則檔、數值編碼、特徵值評分設計，用以辨識處理網路攻擊、惡意軟體、垃圾郵件，正規表示式的使用，直接影響特徵資料庫所需容納的特徵值多寡，並衍生出辨識準確度上的問題，Snort 的規則檔在 PCRE 與 Content 設計上的技巧、ClamAV 即時更新病毒特徵資料庫的頻率、SpamAssassin 的門檻值和貝氏過濾供訓練的樣本皆會直接影響到辨識準確度，進而產生誤擋(False Positives) 與漏擋(Flase Negatives)的問題，如何有效的減少誤擋與漏擋，將會是另一門值得研究的課題。除了上述介紹的三個工具套件，[9-14]整理了其他相關的套件，若讀者有興趣可以自行使用。

參考文獻

[1] A look into IDS/Snort Whole thing, http://www.antionline.com/showthread.php?t=251729

[2] Clam AntiVirus User Manual, http://www.clamav.net/doc/latest/clamdoc.pdf

[3] A. Scbwartz, SpamAssassin, O,REILLY Media, Inc. 2004

[4] A.K. Seewald, “Combining Bayesian and Rule Score Learning: Automated Tuning for SpamAssassin”, Technical Report, Austrian Research Institute for Artificial Intelligence, Vienna, TR-2004-11, 2004.

[5] Snort Users Manual,http://www.snort.org/docs/snort_htmanuals/htmanual_282/

[6] Creating signatures for ClamAV, http://www.clamav.net/doc/latest/signatures.pdf

[7] Phishing signatures creation HOWTO, http://www.clamav.net/doc/latest/phishsigs_howto.pdf

[8] Regular-expressions.info, http://www.regular-expressions.info/refflavors.html

[9] Bro, http://bro-ids.org/

[10] Panoptis, http://sourceforge.net/projects/panoptis

[11] F-prot, http://www.f-prot.com/

[12] Mailscanner, http://www.mailscanner.info/

[13] Bogofilter, http://bogofilter.sourceforge.net/

[14] Spamprobe, http://spamprobe.sourceforge.net/

[15] SourceForge.net, http://sourceforge.net/

[16] 張朝江、林盈達,「沒有固定 port 應用程式的偵測與過濾: L7-filter classifier」,http://speed.cis.nctu.edu.tw/~ydlin/miscpub/hands-on_L7-filter_classifier.pdf

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

[18] Clam AV,http://www.clamav.net/

[19] A. Mitra, W. Najjar, and L. Bhuyan ,”Compiling PCRE to FPGA for accelerating SNORT IDS”, ACM/IEEE Symposium on Architectures for Networking and Communications Systems ANCS 2007.

[20] M. Roesch , “Snort—Lightweight Intrusion Detection for Networks”,Proc. 13th USENIX Conference on Systems Administration(LISA-99), Seattle, Washington, USA, November 7–12, p.229-238, 1999.

[21] M. Attig and J. Lockwood,”SIFT: Snort Intrusion Filter for TCP”, Proc. 13th Symposium on High Performance Interconnects, August 17-19, p.121–127, 2005.

[22] R. Sommer and V. Paxson ,”Enhancing Byte-Level Network Intrusion Detection Signatures with Context”, Proceedings of the 10th ACM conference on Computer and communications security,Washington D.C. ,USA, October 27-30, 2003.

[23] J. Beale, Snort 2.1 Intrusion Detection, Second Edition, Syngress Publishing, 2004