利用流量抽取機制強化真實流量資料庫

陳健宏 何承遠 林盈達
國立交通大學資訊工程學系
E-mail:chchen.cs99g@nctu.edu.tw,cyho@csie.nctu.edu.tw,ydlin@cs.nctu.edu.tw
September 30, 2011

摘要

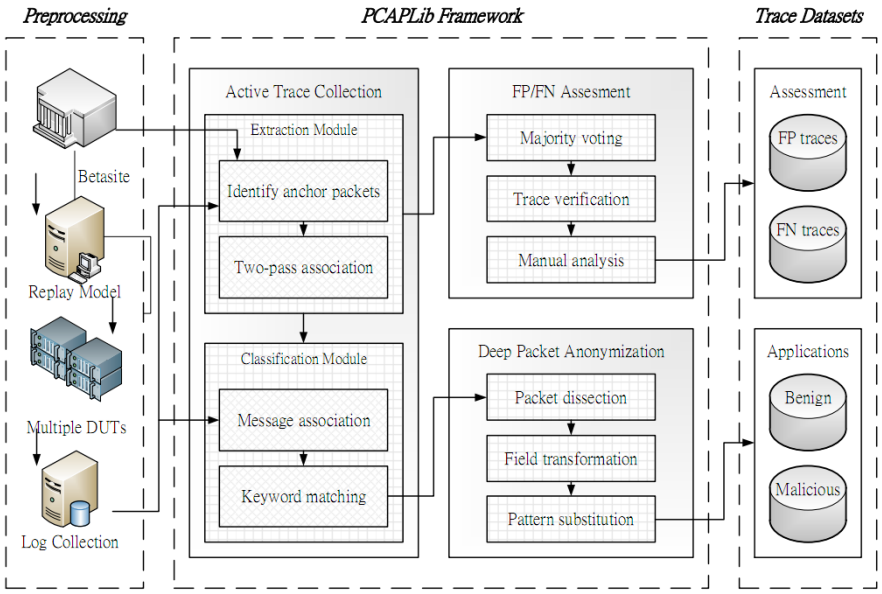
真實流量資料庫(PCAP Lib)的機制能夠對於尚未處理過的真實流量進行萃取、分類與匿名，去收集到各種不同的真實流量樣本，然而目前的萃取機制只能夠基於網路攻擊偵測設備所觸發的警報來辨識需要抽取的連線，使得資料庫中缺乏病毒與垃圾信件的流量樣本，並且無法只針對各種應用層協定的流量進行萃取。本篇研究透過結合 ClamAV 與 Snort 的病毒偵測能力，SpamAssassin 與 ClamSMTP 的垃圾與病毒信件判斷能力，以及 L7-filter 的應用層協定辨識能力設計了一套自動的流量抽取機制，這套機制能夠從未處理流量中辨識出惡意程式、垃圾與病毒信件及不同的應用層協定，並且將這些被辨識出的流量抽取出來。而在本機制的實驗結果中，共抽出了 163 條包含惡意程式的流量，與 4 條的垃圾信件流量，另外也抽出了 14 種不同應用層協定的流量，成功的補強了 PCAP Lib 原本萃取機制中所欠缺的部分，而增新流量之後的資料庫樣本數可參考第八頁之表三。

關鍵詞:真實流量萃取、ClamAV、Snort、SpamAssassin、ClamSMTP、L7-filter

1. 簡介

為了能夠讓做測試時所用的網路流量更貼近於真實環境的情況，使用真實流量來做測試是非常需要的，一個完整分類的真實流量資料庫，可以幫助研究人員更快速的選擇出所需要的流量類別，因此 PCAP Lib[1]建立了一套真實流量萃取、匿名與分析的機制，如圖一所示，PCAP Lib 包含了三大部分，第一，Active Trace Collection 從真實環境所錄製的流量中，經由具有網路流量偵測能力的設備進行判斷，依照判斷的結果萃取出可疑的流量，並進一步對於這些網路流量分類出十種不同的類型，同時也區分為良好或惡意兩大類的流量。第二，Deep Packet Anonymization 是為了避免個人隱私洩漏，對於萃取出來的流量進行匿名化處理，PCAP Lib 除了對於一般 TCP/IP 標頭匿名之外，還對於更高層的內容進行深度匿名，並在最後將抽取的結果收集於資料庫之中，提供了一個方便取得真實流量的環境。而第三個部份的 FP/FN Assessment 是針對於網路測試中，被測試的網路設備並非百分之百準確，因此 PCAP Lib 同時設計了一套透過多台設備進行多數決的方式，來分析設備誤擋與漏擋的流程。

然而目前PCAP Lib的抽取機制是基於入侵偵測系統設備所反應的警告來辨識需要抽取的連線，雖然能夠辨識出包含惡意行為的流量，但這些設備沒有偵測病毒和垃圾信件的能力，所以在目前的資料庫中所蒐集到的流量缺乏病毒與垃圾信件的真實流量樣本，真實的病毒與垃圾信件流量為測試病毒與垃圾信件辨識能力時所需要的，同時現存的機制也無法基於特定的通訊協定流量來進行抽取，故筆者藉由現存的病毒偵測軟體、垃圾信件偵測軟體、通訊協定流量辨識軟體設計了三種不同的抽取機制來強化目前的 PCAP Lib，希望能藉由這三種抽取機制讓現有資料庫中的樣本更加豐富。



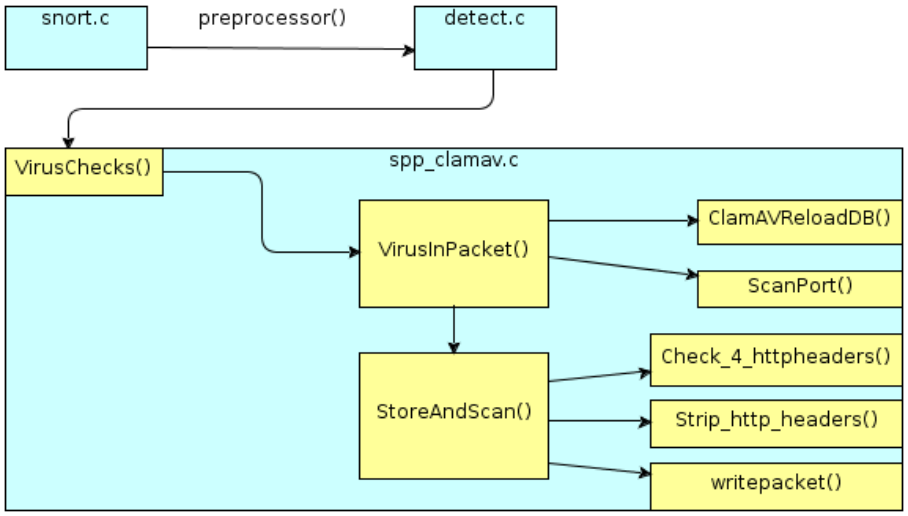
圖一 PCAP Lib 萃取、匿名與分析流程[1]

2. 偵測與辨識軟體簡介

惡意程式流量偵測:Snort 與 ClamAV[2]

為了找到所要收集的惡意流量，必須要先能夠辨識出它們。要對流量進行掃描，這邊使用一套有名的網路攻擊偵測系統 – Snort[5]，來對未處理的流量進行檢查，Snort 使用大量事先定義好的攻擊特徵去比對所有經過的流量，當流量內容符合了某一項規則時，就會觸發警告來提醒使用者可能有攻擊流量經過。不過因為這邊我們需要檢查的是包含有惡意程式的流量，只使用 Snort 的話，只有辦法偵測攻擊，而沒有辦法偵測出惡意程式，所以同時搭配了另一套 Open Source 的 AntiVirus – ClamAV[6]，來進行惡意程式辨識，並使用 Snort inline[7] 所設計的 ClamAV Preprocessor 連結 Snort 的流量處理能力和 ClamAV 的惡意程式偵測能力，進行惡意程式流量偵測。

以圖二來進行說明，每當一個封包經過時，ClamAV Preprocessor 首先會檢查封包是否為使用者所設定需要檢查的封包，並以 Port number 來作為判斷，不過因為我們需要對所有流量進行檢測，故這邊已設定為對所有 Port 的連線都進行檢查，接著如果封包內容包含有 HTTP 的標存在，也會將標頭拿掉來增加掃描時的準確度，最後會對於每一條連線建立一個相對應的暫存檔案，並將處理過後的封包內容寫入對應的檔案，再呼叫 ClamAV daemon 去對檔案做掃描，如果發現檔案有符合病毒碼時，則將該檔案所對應的連線判斷為包含有惡意程式的連線。



圖二 ClamAV preprocessor 封包處理流程[8]

垃圾信件與病毒信件偵測: SpamAssassin 與 ClamSMTP

為了辨識垃圾與病毒信件，這邊使用了兩套過濾信件的軟體，首先使用 SpamAssassin[3][9]來辨識出信件中可能是垃圾信件的特徵，例如不正確的信件標頭，或是信件內容有垃圾信件常出現的關鍵字，並且對於每一種特徵都會有一個相對應的分數，當這些分數相加的結果大於管理者所設定的門檻值時，就會將這封信件判斷為垃圾信件。而除了垃圾信件之外，也同時使用 ClamSMTP[10]來辨識病毒信件，ClamSMTP 可以直接透過 SMTP 的通訊協定來把需要檢查的信件內容送進程式，能夠對信件做解壓縮、解密等等的預先處理，再利用 ClamAV 的 Daemon 來掃描其信件內容，當發現該信件包含有病毒時，則會直接透過 SMTP 回傳警告訊息來通知使用者。

但由於 SpamAssassin 與 ClamSMTP 都必須搭配信件伺服器使用，只能夠對於伺服器所收到或送出的信件進行掃描，兩者皆無法直接對於流量進行信件掃描，故下面的部分將會再介紹如何從流量中將信件抽取出來，再送給這兩套軟體掃描的機制。

應用層協定流量辨識:L7-filter

過去在辨識封包是屬於甚麼應用層協定的流量時，會依照 OSI 第四層的 Port 號碼為依據來做分類，但由於現今的應用層協定會使用 Dynamic-port 來進行連線，只依照 Port 分類協定的方法已經不夠準確，必須直接從第七層的內容來進行辨識才能夠真正的辨識出應用層協定，這邊我們使用 L7-filter classifier[4][11]進行協定的辨識，L7-filter 會依照每個協定出現的不同特徵，直接對於應用層的封包內容進行字串比對搜尋，目前 L7-filter 已提供了一百二十種不同的特徵，由於是直接搜尋應用層的內容，故能夠解決協定跑在不是預設 Port 時，無法被辨識的問題。但由於 L7-filter 是 Linux 的防火牆 IPTables 的一個搭配套件，它的程式雖然提供了過濾流量的功能，卻無法直接套用於分類流量上，所以筆者這邊並不直接使用 L7-filter 的程式，而只利用它所定義好的應用層協定特徵，來自行對於未處理流量的每一個封包內容做正規表示法的比對，藉此找出所要抽取的目標流量。

3. 流量抽取機制

透過上節所介紹的三種不同的辨識與偵測方法，本節將介紹筆者所設計的抽取架構，藉由這些軟體對流量內容的辨識，將辨識的結果依照 5-tuple(來源 IP 位址、目標 IP 位址、來源 Port 號碼、目標 Port 號碼、以及 Protocol ID)來將封包分類成每一條不同的連線，並同時透過 PCAP Library 原有的抽取方式，將相同 5-tuple 的封包抽取成各別的連線。以下將分別介紹三種萃取流程。

病毒流量抽取流程

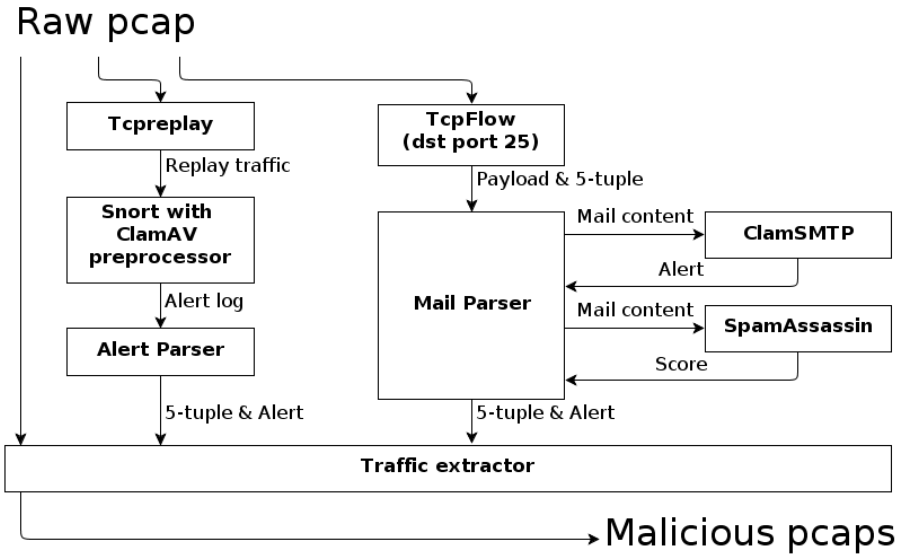
利用圖三的左半部來說明本流程，首先利用 TCPreplay 這套流量重播工具來將未處理流量重播至 Snort 上進行掃描，由於我們已經事先利用 ClamAV preprocessor 來連結 Snort 和 ClamAV 的功能，故可以辨識出重播流量中所包含的病毒流量，待重播完成之後，接著將重播過程中所觸發的警告記錄檔送給筆者所寫的 Alert Parser，Alert Parser 會處理 Snort 的記錄檔，依照相同的 5-tuple 來將這些警告所記錄的訊息整理成一條條的連線，之後再將這每一條連線的 5-tuple 以及觸發的警告一起送給 Traffic extractor 來進行抽取。

垃圾與病毒信件流量抽取流程

由於 SpamAssassin 和 ClamSMTP 並不具有直接對於網路流量進行掃描的能力，而大部分信件會經由跑在 Port 號碼 25 的 SMTP 協定來傳送，為了能夠得到信件本身，其流程如圖三右半部所示，這邊的抽取流程首先會使用 TcpFlow 這套工具

將所有目標 Port 號碼為 25 的連線過濾出來，TcpFlow 同時會將其應用層協定的封包內容組合起來，之後將其組合的內容以及該連線的 5-tuple 一起送給筆者所寫的 Mail Parser。

Mail Parser 會處理這些收到的內容，並依照每一條的連線做下面的處理，首先將信件本文的部分從這些內容中給抽取出來，藉由 SMTP 協定定義，從”DATA”指令到單獨輸入一個”.”為止為本文的規則，可以很快的辨識出信件本文的所在位置。並在抽取出信件本文之後，先透過 SMTP 協定將信件本文傳送給 ClamSMTP 來做辨識，如果該信件中包含病毒的話，ClamSMTP 會直接回傳警告給 Mail Parser，若不包含病毒，則回傳一般 SMTP 的訊息，完成之後，再將信件本文寫入到檔案之中，利用 SpamAssassin 的 test mode 來直接對該檔案進行掃描，其掃描結果所得到的分數會寫入到一個檔案之中，再由 Mail Parser 去讀回，接著再去處理下一條的連線內容。當所有的連線都得到了 ClamSMTP 的掃描結果以及 SpamAssassin 的評比分數之後，最後就將這每一條連線的 5-tuple 以及觸發的警告與分數一起送給 Traffic extractor 來進行抽取。

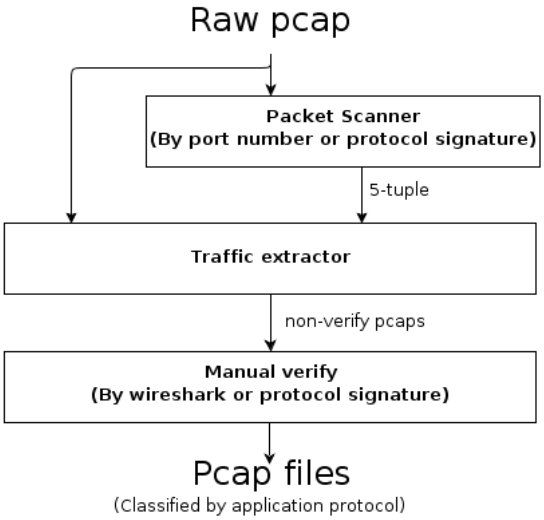


圖三 惡意程式、垃圾與廣告信件流量抽取流程

不同應用層協定流量抽取流程

這邊的抽取流程如圖四所示，首先用筆者所寫的 Packet Scanner 來對於未處理流量中的每個封包進行分析，設定要抽取協定的特徵以及該協定預設的 Port 號碼，掃描過程中會檢視封包的來源或目標 Port 號碼是否符合我們所設定的號碼，同時利用 L7-filter 所定義的特徵來和封包內容做字串比對，當兩者有其中一項符合時，Packet Scanner 會記錄該封包的 5-tuple，並在所有封包檢視

完成之後，將其結果送給 Traffic extractor 來進行抽取。最後抽取完成後，因為目前的準確度還不是百分之百，故需要再由人工透過 Wireshark[12]檢查其連線內容，同時再使用一次 L7-filter 定義的特徵來對所有抽取出來的流量做比對，以達成更高的準確度。



圖四 不同應用層協定流量抽取流程

4. 實驗結果

為了驗證上一節所介紹的三個抽取流程的實用性，這邊選擇了一份從校園網路中錄製半小時的流量來分別對於惡意程式流量抽取、垃圾與病毒信件流量抽取、不同應用層協定流量抽取進行實驗。

實驗流量錄製時間: 2011/07/08 19:10~19:40 實驗流量大小: 15.6GB	
抽取惡意程式流量結果	163 條流量（5 個種類）
抽取垃圾信件流量結果	4 條流量
抽取病毒信件流量結果	0 條流量

表一 惡意程式、垃圾與病毒信件抽取結果

如表一所示，在這半小時的流量之中，本機制成功的抽取出 163 條的惡意程式流量，其觸發了五種不同種類的警告；在垃圾信件的部分，這個機制成功抽取出 4 條被 SpamAssassin 標示為高於 5 分的垃圾信件流量；然而在病毒信件的部分，則沒有偵測出任何一條連線包含有病毒信件，故這邊沒有抽取出任何一條的流量，但在筆者自行所產生包含有病毒信件的流量下，本機制可以成功抽取出該病毒信件流量，故這邊認為可能是選擇的實驗流量中並沒有包含有病毒信件所導

致。

在抽取不同應用層協定流量的部分，可參考表二所列之結果，本機制最後抽出了 14 種不同的應用層協定的流量，其中抽取數量較少的幾種協定經過人工全部驗證後，已確認並沒有問題，而抽取數量多的協定則隨機挑選數比流量來做驗證，其驗證結果也都為正確。但也遇到了兩個意外的情況，在抽取 edonkey 和 IRC 的流量時，依照 edonkey 的特徵抽取出來的結果，其準確度太低，筆者從結果中挑選出數筆流量來進行驗證，皆未發現正確的 edonkey 流量，故不將抽取的結果列上，而 IRC 協定的部分雖然也抽取出來一些流量，但經過檢驗之後發現其流量的內容皆只包含 SYN 封包而已，並未抽取出真正有 IRC 協定溝通的流量，因此也不將 IRC 列為成功抽取的協定之一。

協定名稱	抽取連線數目	協定名稱	抽取連線數目
Bittorrent	6048	SMB	138
DNS	9386	SMTP	4
FTP	96	Soulseek	1
HTTP	53053	SSH	3
MSNMessenger	41	Telnet	1463
POP3	3	VNC	1
PPlive	119	YahooMessenger	1

表二 不同通訊協定抽取結果

5. 結論

本篇文章提供了一個新的流量萃取機制來加強目前 PCAP Lib 不足之處，利用 Snort 與 ClamAV 的搭配辨識出包含有惡意程式的連線，對於垃圾與病毒信件分別使用 SpamAssassin 和 ClamSMTP 來進行信件的分析，最後透過 L7-filter 所定義的應用層協定特徵辨識出不同應用層的協定，為 PCAP Lib 提供了更多樣化的流量。然而目前的機制仍然有不少限制，為了偵測惡意程式而重播流量到 Snort 時，若重播的速度過快則會導致 Snort 偵測的準確度降低，漏判率提高，所以目前為了保持 Snort 偵測的準確度，使用了很低的速度來進行重播的工作，但卻使得處理的時間大幅加長，將來可以考慮如何去解決 Snort 偵測效率與準確度的問題。在垃圾信件的部分，目前僅處理 SMTP 協定的信件偵測，未來也可以再針對 POP3 以及 IMAP 等傳送信件的協定去做擴充。而在辨識應用層協定的部分，依靠的L7-filter所定義的特徵雖然有上百種，但真正品質較好，誤判率較低的特徵，僅有十到二十種特徵，要如何再去加強這個部分的辨識能力，仍然值得探討。最後的抽取結果經過筆者從中挑選適當大小與數量的流量出來，加入到現有的真實流量資料庫之中，加入之後的資料庫樣本數量可以參考表三。

流量種類	通訊協定	流量數量	流量種類	通訊協定	流量數量
Chat	AIM	1	Encryption	FTPs	1
	Googletalk	1		HTTPs	1
	ICQ	4		SSL	11
	IRC	7	Streaming	Octoshape	1
	MSN	44		Orb	1
	Skype	1		PPLive	96
	Yahoo	5		QuickTime	1
Email	IMAP	5		Slingbox	1
	POP3	8	File Sharing	Azureus	1
	SMTP	15		BearShare	1
File Transfer	FTP	124		Bittorrent	102
	SMB	122		eDonkey	1
	TFTP	1		Gnutella	1
Network	DNS	119		Groove	1
	NetBIOS	22		LimeWire	1
	SNMP	3		P2P	4
	Socks	1		Pando	1
	STUN	1		Pruna	1
Remote Access	VNC	3		Sharelike	1
	Telnet	106		SoftEther	1
	SSH	7		Soulseek	2
	RDP	4		TeamViewer	1
VoIP	SIP	4		Winny	1
Web	HTTP	227		Xunlei	1
流量種類	流量數量		流量種類	流量數量	
Benign	794		Attack	101	
Virus	165		Spam	9	

表三 強化過後的真實流量資料庫之通訊協定流量樣本數

參考文獻

[1] 王聲浩，林盈達; 「萃取、分類及匿名封包流量與誤檔漏檔之個案研究」，2010 年。

[2] 王聲浩、陳一瑋、林盈達; 「攻擊、病毒與廣告信的辨識機制與套件」，2008 年。

[3] 李志祥、曹世強、林盈達； 「剖析三大代理伺服器-快取、防火牆及內容過濾」； 網路通訊；151 期，2004 年 2 月。

[4] 張朝江、林盈達; 「沒有固定 port 應用程式的偵測與過濾: L7-filter classifier」, 2006 年。
[5] Snort
http://www.snort.org/
[6] ClamAV.
http://www.clamav.net/
[7] Snort-inline.
http://sourceforge.net/projects/snort-inline/
[8] Snort Inline by Pete Savage.
http://linuxgazette.net/117/savage.html
[9] The Apache SpamAssassin Project.
http://spamassassin.apache.org/
[10]ClamSMTP: An SMTP Virus Filter.
http://thewalter.net/stef/software/clamsmtp/
[11]Application Layer Packet Classifier for Linux.
http://l7-filter.sourceforge.net/
[12]Wireshark · Go deep.
http://www.wireshark.org/