建置入侵偵測防禦系統

Ⅰ. 實驗目的

隨著網路越來越風行，網路安全的議題也日益受到重視。尤其是近來層出不窮的駭客攻擊，如 DoS、資料竊取、網站破壞等等，幾乎都造成企業很大的損失。因此近來出現許多針對這方面作防護的 IDS(Intrusion Detection System)及 IPS(Intrusion Protection System)等系統。和傳統的防火牆差別在於，這類 IDS/IPS 會針對封包內容作特徵分析及異常比對。本實驗透過 Snort 軟體的 inline 模式，掃瞄封包的內容決定作取代，丟棄等動作。來模擬 IPS 的特徵比對及防護。在進行實驗之前，需要有底下幾項的預備知識：

1.Linux 作業系統的基本操作，設定的能力；

2.瞭解網路封包的架構；

3.安裝及設定 Linux 上的網路伺服器(Apache，Proftpd 等)。

Ⅱ. 實驗設備

名稱	數量	備註
個人電腦	3	分別作爲內部 server，snort 防火牆，及外部的 client
網路卡	4	防火牆需要兩張，其他電腦各一張
Gentoo Linux	1	可免費從網路下載
Windows XP	1	
iptables	1	各 distribution 都有收錄
snort	1	各 distribution 都有收錄

Ⅲ. 背景資料

面對各類的網路駭客攻擊，針對封包內容作掃瞄的防火牆日益重要。由於攻擊的方法越來越多，防火牆也必須要隨之演進，才能保護網路的安全。像 IPS 就逐漸地引起注意。因爲網路上常見的攻擊通常都是先送出一段特定的字串，造成 buffer overflow，此類防火牆通常是對封包作分析，然後和資料庫收集的特徵(signature)，也就是用來攻擊的字串比對，決定是否通過。另外因爲可以針對 5~7 層的封包做過濾，像是 HTTP worm 之類的攻擊也能做防護。

目前市面上有許多硬體防火牆有此類功能，像是 Fortinet 的 FortiGate，還有 Netscreen IDP，McAfee i 系列等。而除了硬體防火牆之外，也有 Snort 這套軟體可以達到類似的功能。Snort 可以在 UNIX-Like 的作業系統(例：Linux，FreeBSD)及 Windows 上運作，並且可以免費取得，因此可以用很低的負擔就可以有一個基本的 IPS 防火牆。

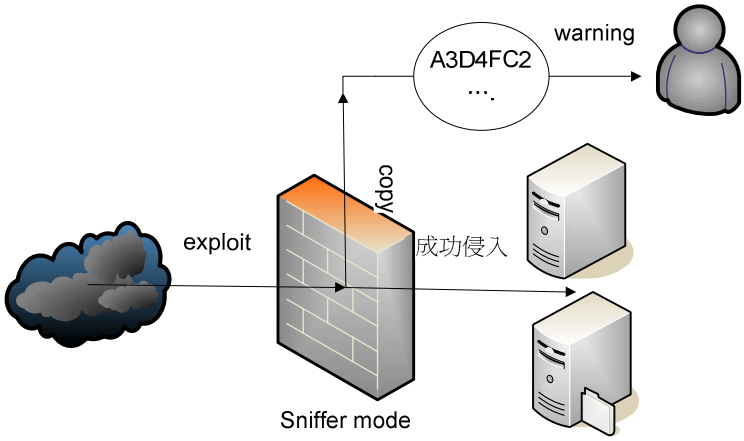
Snort 作者是 Martin Roesch，在 1998 年寫出來，目的是要做一個”輕量級”的 IDS。隨

著時間演進，功能也越來越多，也足以當作一個基本的 IPS 防火牆。在 snort 的網站 http://www.snort.org 上，除了可以下載程式外，還有討論區、文件等等。另外他們還有整理好防範攻擊的 rule，不過此服務需要付費才能即時取得，免費的下載需要等到 5 天後才能下載。目前約有 4000 條 rule。另外也可以使用自己上傳的 rule，但是這部分並沒有驗證一定可以阻擋。其 rule 分類如下：

通訊協定偵測		攻擊	其他(病毒，內容過濾)
chat.rules	p2p.rules	attack-responses.rules	bad-traffic.rules
dns.rules	pop2.rules	backdoor.rules	experimental.rules
finger.rules	pop3.rules	ddos.rules	info.rules
icmp.rules	rpc.rules	dos.rules	local.rules
icmp-info.rules	rservices.rules	exploit.rules	misc.rules
imap.rules	smtp.rules	scan.rules	other-ids.rules
multimedia.rules	sql.rules	shellcode.rules	policy.rules
mysql.rules	telnet.rules	web-*.rules	porn.rules
netbios.rules	tftp.rules		virus.rules
nntp.rules	web-*.rules		
oracle.rules	x11.rules		

Web-* 中包括 web-attacks.rules、web-cgi.rules、web-client.rules、web-coldfusion.rules、web-frontpage.rules、web-iis.rules、web-misc.rules、web-php.rules。

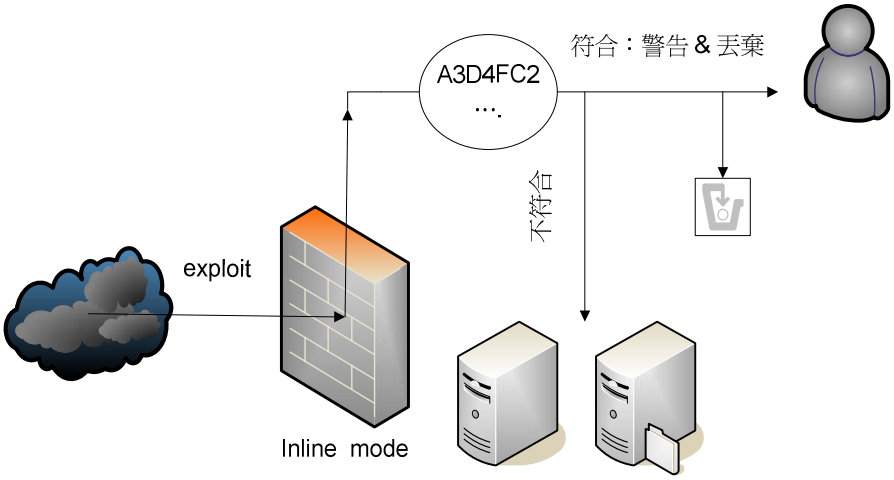
Snort 軟體主要有兩個功能，一個是 sniffer 模式，就如同 Ethereal、Sniffer 的功能。此模式在封包通過網路介面時，copy 一份做比對，如果偵測到符合 rule，就會做 log，但是不能做封鎖之類的後續防護。運作過程如圖一：



圖一、 Snort sniffer mode。

另外就是 2.3.0 RC1 版後新增的 inline 模式。主要的不同在於不會另外複製一份，而直接對封包掃瞄(也就是 inline)，決定要做替換、拒絕、通過等等，也可以設定要不要對管理者

警告，此模式一開始是從 Snort 的原始碼獨立發展的專案(http://snort-inline.sourceforge.net/)，之後才整合到原本的程式中。這種從原本的程式分出來獨立發展，是開放原始碼社群常見的發展模式，還有其他的例子像是 Xfree86 與 Xorg，還有 wine 和 winex 等等。這次實驗主要就是使用他的 inline 模式。運作過程如下圖所示：

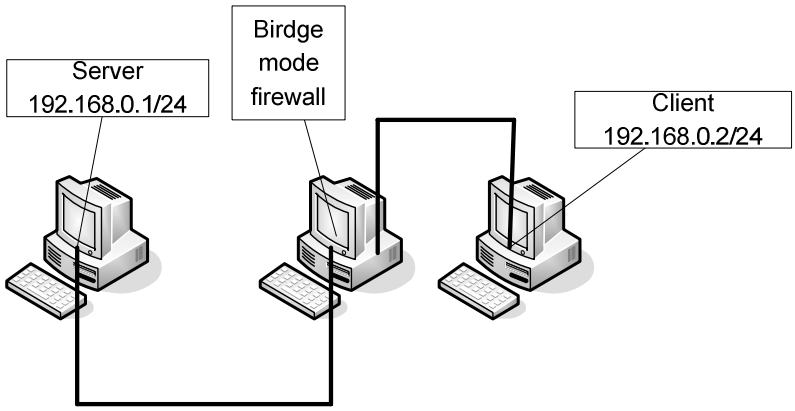


圖二、Snort inline mode。

Snort 的 inline 模式透過 iptables 軟體來運作。先由 iptables 送到由 ip_queue 模組維護的 queue 中，而 Snort 再從其中讀取封包來做比對。因此執行前需要確定 iptables 套件以及 ip_queue 模組可以正常工作。

IV. 實驗方法

本實驗首先要先使用 Linux 電腦建立 bridge，做爲內部 server 及外部 client 之間的防火牆。整個網路架構如圖三所示：



圖三、實驗環境。

而每台電腦的用途如下：

1.Server(Linux)：安裝被 client 模擬攻擊的伺服器，以及觀察被 snort 過濾後的封包。

2.firewall(Linux)：安裝 Snort，以及紀錄 snort 的 log。

3.client(Windows)：送出網路封包來驗證 snort 是否正常工作。

實驗分成三個階段，第一個階段是先熟悉 snort 的設定語法，以及操作。第二階段就是針對各種通訊協定(http,ftp,…)的封包作分析，比對。最後是就是實際拿幾種攻擊做練習。

底下介紹 Snort 一些基本的使用：

參數	說明
-v	顯示詳細的封包的 header
-d	顯示應用層封包內容
-e	連結層的資訊也顯示出來
-l	設定 log 存到那個檔案
-Q	從 iptables 取得封包
-i	指定要 listen 那個網路介面
-D	以 daemon 模式執行
-c	讀取指定的設定檔

而 inline 模式的設定，首先要用 iptables 把要過濾的封包送到 queue 中。

```
#iptables -A FORWARD -p tcp --dport 80 -j QUEUE
```

上面的例子是將對內的 80 port 封包送過去。

然後就是啓動 snort，依據自定的規則作過濾。

```
# snort -Q -i eth0 -l /var/log/snort -c /etc/snort/snort.conf
```

設定檔中的過濾條件大都是像這樣的格式：

```
<action> <protocol> <from_ip> <from_port> <direction> <dest_ip>
<dest_port> (<rules>)
```

1.Action

設定值	說明
alert	使用 rule 中設定的方法警告，並且記錄下來
log	記錄下來
pass	略過封包
drop	告訴 iptables 丟棄封包
sdrop	告訴 iptables 丟棄封包，並且不做紀錄
reject	告訴 iptables 拒絕封包
active	如同 alert，然後啓動另一條 rule
dynamic	如同 log，但只會由 active 啓動

2.protocol：目前支援 tcp，udp，icmp，ip 四種。

3.from_IP/dest_IP：可以直接設定 IP，或用 CIDR 設定一段網域。也可以用中括號設定多個 IP、CIDR，中間用逗號隔開。在前面前上 ！代表不包含這些 IP。

4.from_port/dest_port：指定一個 port，或用 ：限制一段範圍。如果冒號左邊/右邊不指定代表無下限/上限。

5.direction： -> 由左至右， <-> 雙向。

Rule 部分有以下的參數可以設定，各設定間用分號隔開。

1.msg

格式：msg: "<message text>";

說明：設定 log 及 alert 動作時顯示的訊息。

2.content

格式：content: "<content string>";

說明：設定要比對的封包，符合的話才會啓動 rule。內部可以使用 ascii 或十六進位。如果是十六進位的話需要用 | 括住。另外，在字串前加上 ! (!"<content string>") 則是做反面指定，不符合的話就會啓動 rule。

3.replace

格式：replace: "<replace string>";

說明：將前一個符合 content 的字串取代成 replace 指定的內容。

另外還有一些修飾的參數，一樣是寫在 rule 中，如下表所示：

參數/格式	說明
nocase	不管大小寫
rawdata	看封包原始內容，沒有經過解碼
depth:<number>;	只看封包前幾個 byte
offset:<number>;	從第幾個 byte 開始看
distance:<number>;	設定兩個 content 間間格多少 byte ex: content:”a”; content:”c”; distance:2 代表只要 a，c 間間格 2 bytes 就符合

另外設定檔中的字串都可以設定成變數，像是 IP 可以用一個變數來指定：

```
var source_ip [140.113.87.0/24,192.168.0.0/16]
log tcp $source_ip any -> any any (msg:”a simple test”;)
```

V. 實驗步驟

1. 設定 bridge

在 firewall 電腦上編輯 /etc/conf.d/net，加入以下內容：

```
brctl_br0=( "setfd 0" "sethello 0" "stp on" )
bridge_br0="eth0 eth1"
config_eth0=( "null" )
config_eth1=( "null" )
config_br0=( "null" )

depend_br0() {
```

```
  need net.eth0 net.eth1
}
```

eth0 和 eth1 依實際情況而定

然後確定 /etc/init.d/ 有 net.eth0，net.eth1，net.br0 的 init script。如果沒有的話從 net.lo0 建立 symbol link 到這些檔案。接著安裝 net-misc/bridge-utils 管理 bridge 的套件。

```
#emerge net-misc/bridge-utils
```

啓動 bridge，以 ping 測試 client 和 server 是否可以互通

```
#/etc/init.d/net.br0 start
(在 client 執行)
#ping 192.168.0.1
```

如果啓動時出現這樣的錯誤訊息：

add bridge failed: Package not installed

代表核心沒有編譯 bridge 的支援，開起下列選項並重新編譯核心

```
Device Drivers  --->
 Networking support  --->
  Networking options  --->
   <> 802.1d Ethernet Bridging
```

2. 啓動 iptables：

首先安裝 iptables 套件

```
#emerge iptables
```

輸入以下的指令，讓 iptables 把全部 tcp 封包送到 queue 中。

```
#iptables -A FORWARD -p tcp -j QUEUE
```

如果啓動時出現這樣的錯誤訊息：

FATAL: Module ip_tables not found.

代表核心沒有編譯 iptables 的相關支援，開啓核心中下列的選項：

```
Device Drivers  --->
 Networking support  --->
  Networking options  --->
   [] Network packet filtering (replaces ipchains)  --->
```

開啓此項及他底下的

```
 IP: Netfilter Configuration  --->
  <> IP tables support (required for filtering/masq/NAT)
```

以及這底下的所有功能。

注意此時因爲所有封包都送到 queue 中，但是沒有相對應的程式處理，所以伺服器會暫時連不上(封包都卡在 queue 中)。

3. 安裝 Snort

Gentoo 安裝 Snort 的預設是不包含 inline 模式。如果要啓動 inline 模式，在 /etc/portage/package.use 加入：

```
net-analyzer/snort inline
```

然後安裝 snort

```
#emerge snort
```

4. 設定 Snort

編輯 /etc/snort/snort.conf 設定底下的 rule：

```
log tcp any any <> 192.168.0.1 80 (msg:"HTTP conn";)
```

5. 啓動 Snort

輸入底下指令啓動 snort

```
snort -Q -i eth1 -l /var/log/snort -c /etc/snort/snort.conf
```

起動時 Snort 會顯示一些訊息，包括目前執行的模式(Inline or Sniffer)、設定檔的位置、rule 數目、log 位置等等。另外如果有加上 –v –d 等顯示封包的參數，他也會把目前的封包即時的顯示出來。將 snort 啓動時的訊息節錄前五行記錄下來**【記錄一】**。

然後在 client 使用瀏覽器連接 http://192.168.0.1 。在實驗記錄中記錄可否連線，以及 Snort 的 log 訊息(log 可在 –l 參數設定的目錄中觀察)**【記錄二】**。

6. 過濾條件試驗

除了 log 以外，Snort 還支援了 alert，pass，drop，reject，sdrop。把上面 rule 的 log 改成這些，分別記錄下結果(可否連線，log 訊息，其他狀況等等)。**【記錄三】**

7. 內容過濾

對 IPS 系統來說，最重要的就是比對封包，確定是不是要做攻擊。Snort 在比對上最基本的就是 content 參數，以及其他修飾的參數等。

首先先設定 rule：

```
log tcp any any <> 192.168.0.1 80 (msg:"admin connection";
content:"/admin/"; replace:"/Admin/")
```

然後在 Apache root 目錄下建立一個 Admin 目錄，裡面任意放置 index.htm。使用瀏覽器開啓 http://192.168.0.1/admin/，觀察 log。**【記錄四】**

8. 對 FTP 過濾

Snort 有內建對封包的解碼器，所以幾乎每種通訊協定都可以直接對他的內容比對(像 http 就可以比對 GET，POST，PUT 等指令)。底下測試 FTP 通訊協定的過濾。

首先在 server 先安裝好任一種 ftp 伺服器(ProFTPD，PureFTP，Wu-ftp…)，啓動並確

認可以正常運作。接下來設定 rule 對下面幾種虛擬的情況過濾：

1.過濾含有病毒的檔案 virus.zip，禁止使用者上下傳此檔

2.防止使用者上傳內容有機密資料的文件，文件的內容有 “Important”

將過濾的 rule 以及 log 記錄下來。**【記錄五】**

9. 實際攻擊試驗

1. Win2k IIS UNICODE 漏洞

這個是早期蠻有名的 IIS 漏洞。一般伺服器會阻擋 ../ 之類讀取 http 根目錄以上目錄的網址，但是如果把 . 轉成 Unicode 再傳過去，IIS 並不會阻擋，因此可以看光整台電腦的檔案，甚至使用 cmd.exe 執行任意指令。有許多的網頁置換就是透過這個漏洞達到的。底下的網址就是一個例子：

http://www.thisisanesample.com/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir+c:\

這可以執行 dir，在瀏覽器把 c:\ 下的檔案列出。

接下來就實驗以 Snort 阻擋此類攻擊。設定好 rule 將此類網址 deny 或是 reject，並且在 log 中記錄下來。把你設定的 rule 以及 log 記錄下來。**【記錄六】**

VI. 實驗記錄

【記錄一】

啓動訊息：

【記錄二】

連線狀況：

snort log：

【記錄三】

類別	連線狀況
alert	
pass	

drop	
reject	
sdrop	

【記錄四】

log：

【記錄五】

snort rule：

log：

【記錄六】

snort rule：

log：

Ⅶ. 問題與討論

注意：請針對問題中每一項目回答，並避免引述「太」多資料。

1. rule 中的方向有 -> 及 <-> 兩種，試解釋爲何沒有 <- 的方向(右到左)。
2. 簡短解釋 Snort 啓動時各項訊息所代表的意義。
3. 設定 rule 時，有那些設定的技巧可以讓過濾的效能提高？

4. 設定 rule 的 action 時，有 deny，reject 兩種拒絕的方法，說明這兩者的不同，並且指出他們 TCP 連線中的不同處。
5. 承上題，兩種方法你覺得在建立防火牆時用什麼比較好，爲什麼？
6. 如果你要入侵一台電腦，但那台電腦有類似此實驗的防火牆保護，那你還有什麼方法可以成功入侵？
7. 找出 Snort 在對封包內容和 content 設定比對時所用的是什麼字串比對演算法，並分析使用此演算法的優缺點，是否還有更好的方法？
8. 從 Snort 網站下載 rule，選一個 rule 來分析他阻檔的封包形式。
9. 說明 Snort 的 HTTP decoder 對封包做哪些處理。
10. 試設計一段實驗測試 rule 的 performance。

Ⅷ. 參考文獻

[1]. “IDS 偵測網路攻擊方法之改進,” http://www.cert.org.tw/document/column/show.php?key=85 .

[2]. “SnortUsers Manual 2.2.3,” http://www.snort.org/docs/snort_htmanuals/htmanual_233/ .

[3]. “Bridging Howto,” http://bridge.sourceforge.net/howto.html .

[4]. “Securing Debian Manual Appendix D - Setting up a bridge firewall,” http://www.linuxsecurity.com/docs/harden-doc/html/securing-debian-howto/ap-bridge-fw.en.html .

[5]. “Gentoo Handbook,” http://www.gentoo.org/doc/en/handbook/ .