Module 13 ： 入侵偵測

學習目的

1. 網路的普及率逐年升高，網路服務的增加以及網路生活化，讓網路安全成為近年來熱門的研究主題之一。對於尚未被入侵的系統而言，防火牆是一道防線；然而，一旦系統遭到入侵，內部資源便無任何隱蔽性可言。因此入侵偵測系統(intrusion detection systems, IDS)成為系統的另外一項保障，寄望達到事前預防、增加網路安全層級。
2. 入侵偵測系統的基本工作原理是從TCP/IP網路上偵測到入侵行為的特徵模式(signatures)，建立入侵偵測資料庫，利用模式比對方式，判別是否具有攻擊的意圖，並偵測出使用何種方法來入侵主機。

3. 本課程模組中，我們將介紹各式入侵偵測系統的技術，探討入侵偵測系統主要類型-網路型入侵偵測系統(Network-based Intrusion Detection Systems)、主機型入侵偵測系統(Host-based Intrusion Detection Systems)。並且以Snort建立一個入侵偵測系統的環境，在實驗過程內收集和統計相關資料，加以分析解釋。
4. 本模組共有三個小節包括(1)入侵偵測及入侵防禦系統概念(2)入侵偵測實作(3)入侵防禦系統實作(4)專案實作，共須三個鐘點。

Module 13 ： 入侵偵測

- Module 13-1: 入侵偵測及入侵防禦系統概念(*)
- Module 13-2: 入侵偵測實作(**)
- Module 13-3: 入侵防禦系統實作(***)
- Module 13-4: 專案實作(**)

* 初級(basic):基礎性教材內容
**中級(moderate):教師依據學生的吸收情況，選擇性介紹本節的內容
***高級(advanced):適用於深入研究的內容

Module 13-1:入侵偵測及入侵防禦系統概念(*)

入侵偵測及入侵防禦系統概念

- 何謂入侵偵測系統
- 入侵偵測系統的種類
- 入侵偵測系統的限制
- 何謂入侵防禦系統
- 入侵防禦系統的種類
- 入侵偵測系統及入侵防禦系統之差異
- 集中式威脅管理簡介

何謂入侵偵測系統

- Intrusion Detection System
- 入侵偵測(Intrusion Detection)就是對電腦網路和電腦系統的關鍵結點的資訊進行收集分析，偵測其中是否有違反安全策略的事件發生或攻擊跡象，並通知系統安全管理員。
- 一般把用於入侵偵測的軟體，硬體合稱為入侵偵測系統。

IDS能做些什麼

- 監控網路(NIDS)和系統(HIDS)
- 發現入侵企圖或異常現象
- 主動告警，通知系統管理者現在網路狀況
- 將網路封包紀錄下來以為未來辨識或作為證據之用

為什麼需要IDS

- 防火牆功能不足
  - 無法阻擋合法網路連結
  - 自身可以被攻破
  - 對於某些攻擊的保護很弱
  - 不是所有的威脅均來自防火牆外部
- 入侵很容易
  - 入侵教學隨處可見
  - 各種駭客工具垂手可得

防火牆防不到的攻擊

- 緩衝區溢位攻擊(Buffer Overflows)
- 通訊埠掃瞄攻擊(Port Scans)
- 木馬程式攻擊(Trojan Horses)
- 碎片封包攻擊(IP Fragmentation)
- 蠕蟲攻擊(Worms)
- 系統與應用程式漏洞攻擊(System & Application Vulnerabilities)

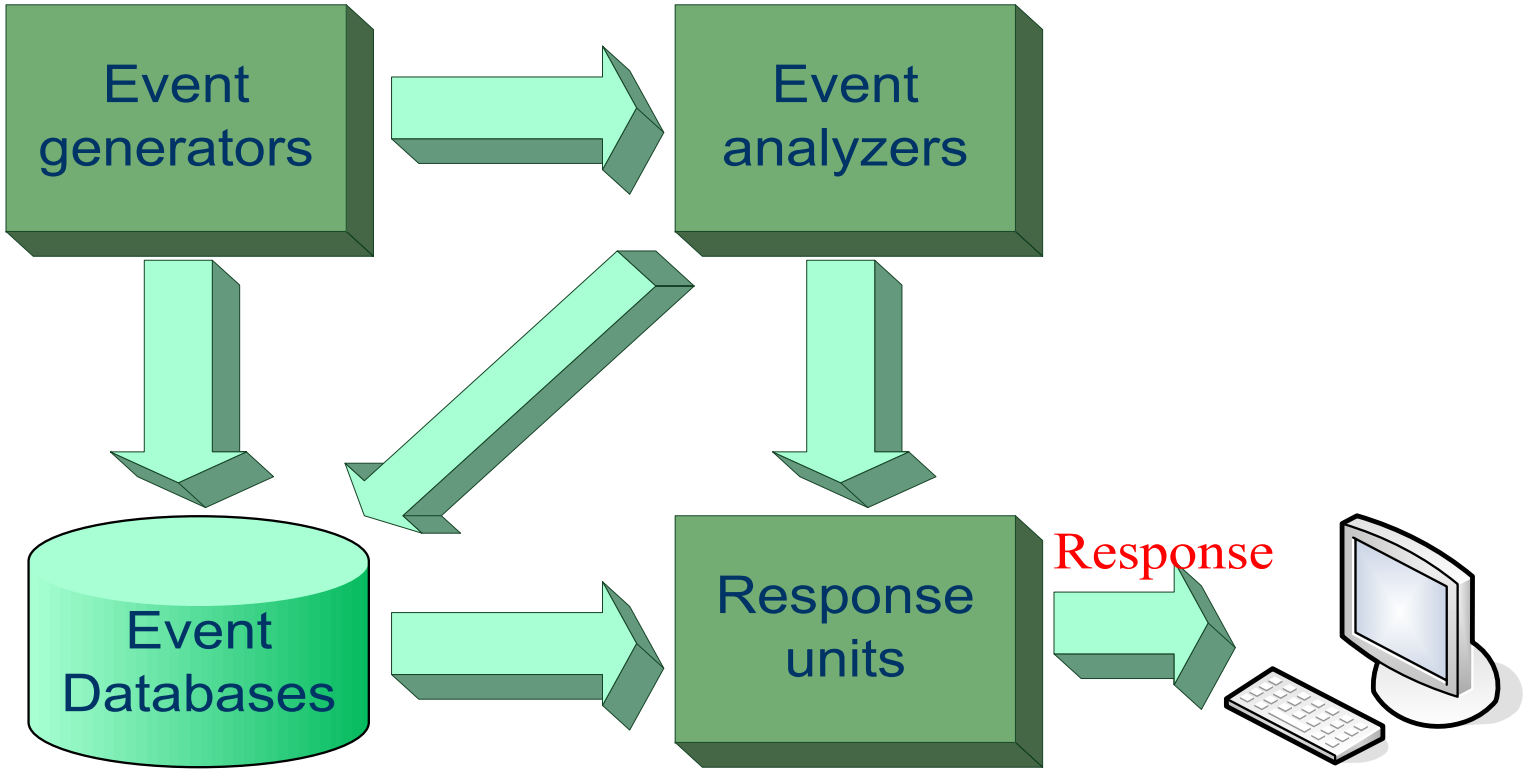
防毒軟體防不了的攻擊

- 緩衝區溢位攻擊(Buffer Overflows)
- 通訊埠掃瞄攻擊(Port Scans)
- 系統與應用程式漏洞攻擊(System & Application Vulnerabilities)
- 阻斷服務與分散式阻斷服務攻擊(DoS/DDoS)

CIDF模型

- 為了提高IDS產品、組件及與其他安全產品之間的相互溝通，Common Intrusion Detection Frame闡述了一個入侵偵測系統（IDS）的通用模型。
- 元件：
  - 事件產生器（Event generators）
  - 事件分析器（Event analyzers）
  - 回應元件（Response units）
  - 事件資料庫（Event databases）

CIDF模型架構


Event generators
Event analyzers
Event Databases
Response units
Response

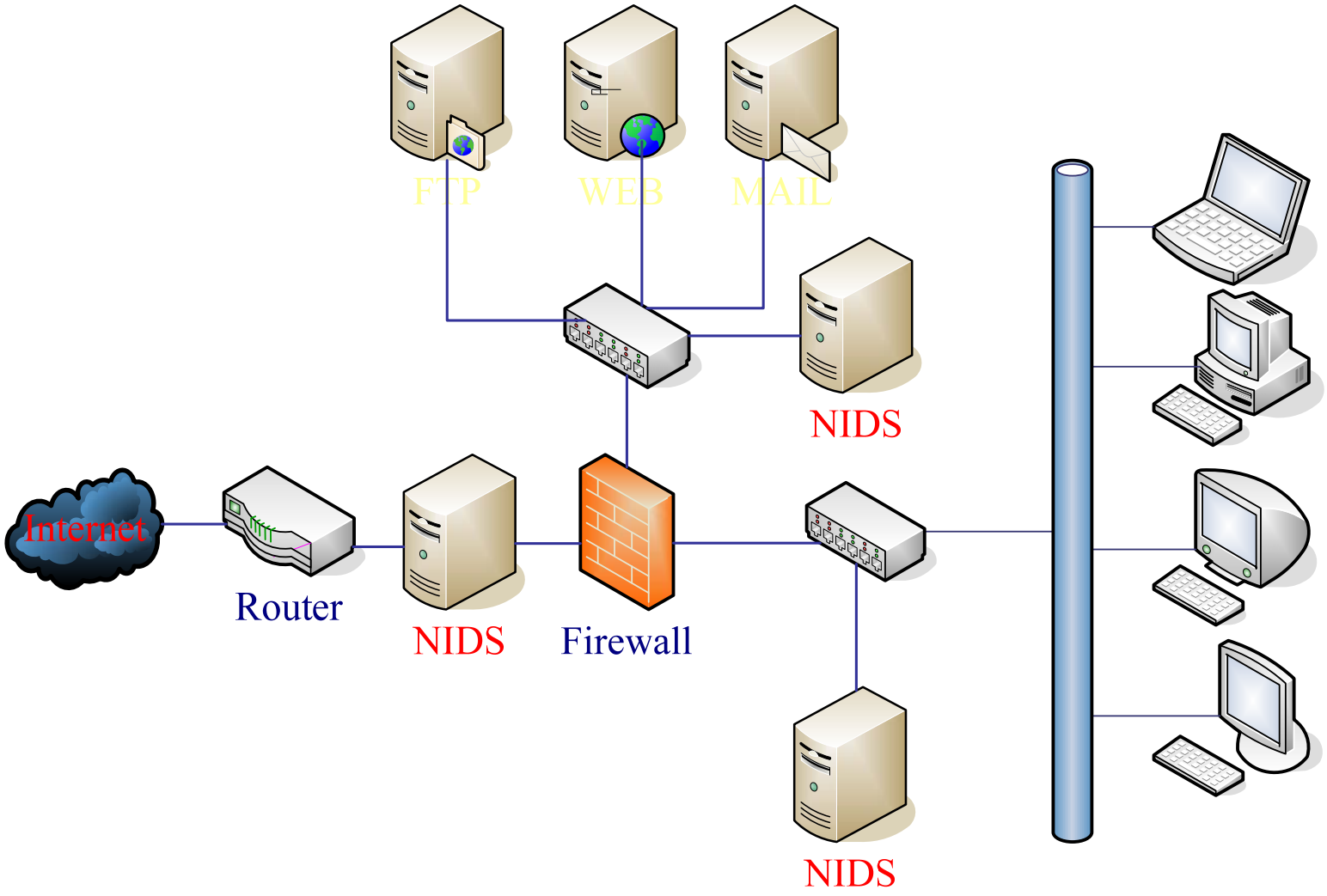
入侵偵測系統之種類

- 網路型入侵偵測系統Network-based IDS,簡稱NIDS
- 主機型入侵偵測系統Host-based IDS,簡稱HIDS
- 網路節點入侵偵測系統Network Node IDS,簡稱NNIDS

NIDS

- 安裝於被保護的網段中
- 雜亂模式監聽
- 分析經過這網段的所有封包
- 不會增加網段中主機的負載
- 產品：eTrust、Snort

NIDS範例



NIDS放置的位置

- 放在防火牆外 ：優點是IDS能夠看到所有來自Internet的攻擊者對系統的各種攻擊手段；缺點就是IDS的負荷會加重。
- 放在防火牆內：也有人認為應該把偵測器放在防火牆內，這樣可以用設置良好的防火牆把大部分的“幼稚腳本”阻止在防火牆外，而讓IDS把注意力集中在高水準的攻擊上。而且這樣可以把IDS保護在防火牆內，免於遭受攻擊。

NIDS放置的位置

- 防火牆內外都放IDS：如果組織的經費充足的話，可以在防火牆的內外都放IDS，這樣就可以得到以上兩種方法的優點。
- 這種情況下，一般放在防火牆內部的IDS是用來作為緊急告警的裝置。

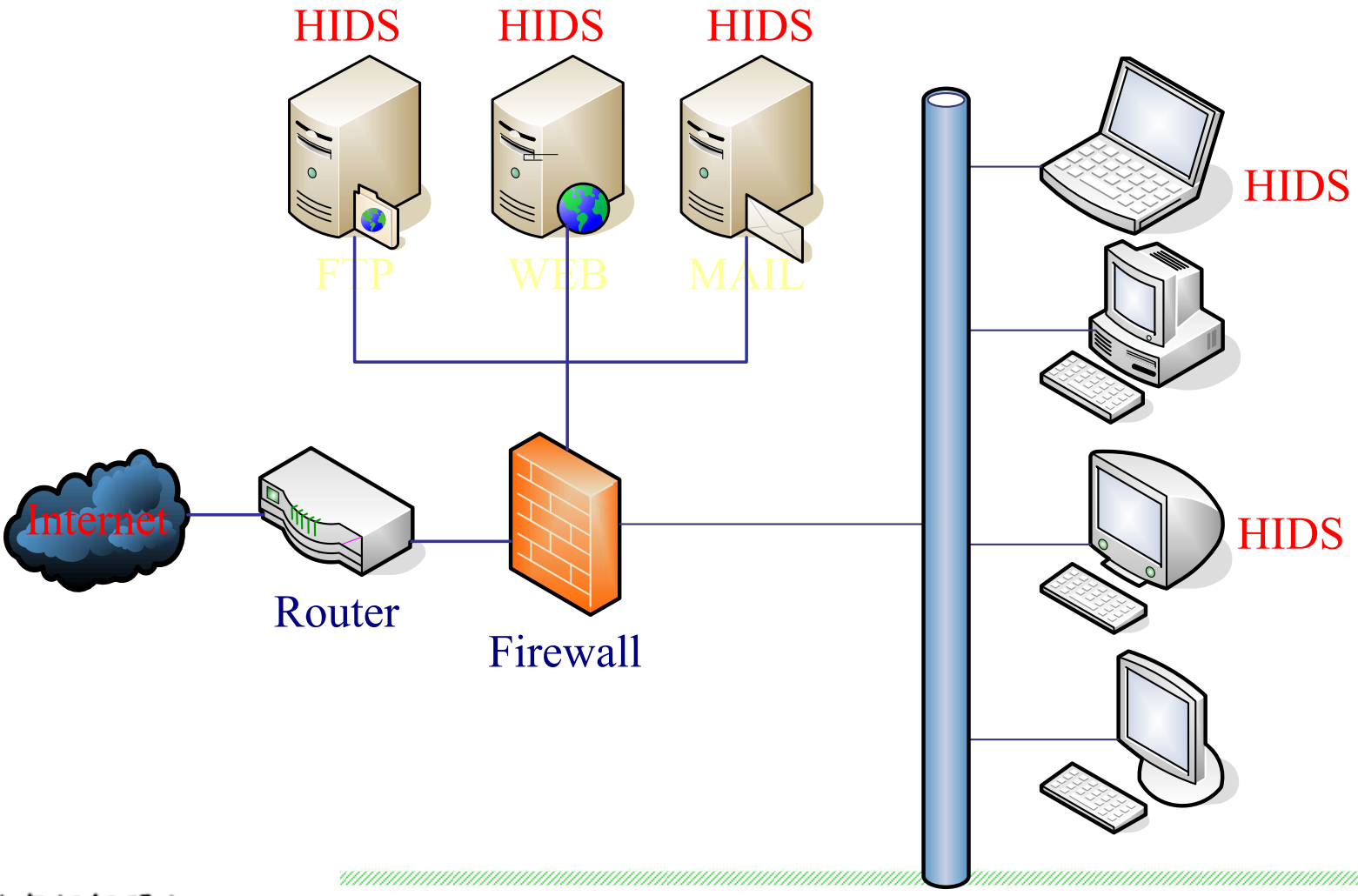
NIDS的優勢

- 花費較少
- 可分析封包
- 防止入侵的證據被移除
- 即時偵測和回應
- 不良意圖的偵測
- 不受作業系統影響

HIDS

- 安裝於被保護的主機中
- 主要分析主機內部活動
  - 系統LOG
  - 系統Process
  - 文件完整性檢查
- 佔用一定的系統資源
- 產品：Enterasys Dragon Host Sensor 、Tripwire

HIDS範例


HIDS
HIDS
HIDS
FTP
WEB
MAIL
HIDS
Internet
Router
Firewall
HIDS

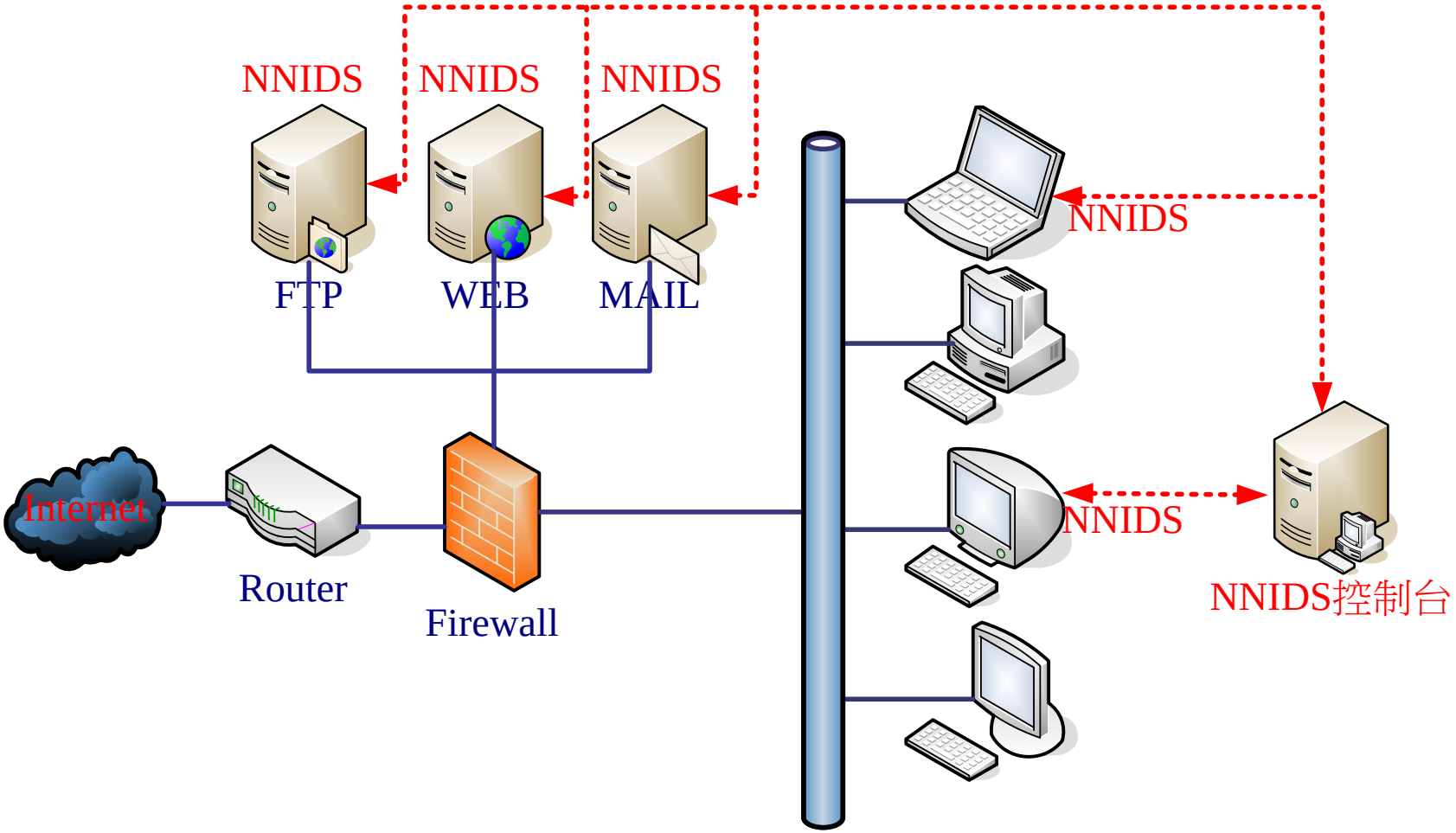
HIDS的優勢

- 可確認攻擊是成功的
- 監控系統特定的活動
- 可偵測加密封包及交換網路環境中的攻擊
- 監控系統關鍵部份
- 不須新增額外硬體

NNIDS

- Network Node IDS也稱作**Stack-Based IDS**
- 安裝於網路節點的主機中
- 結合了NIDS及HIDS的技術
- 適合於高速網路環境：NIDS因為效能的關係，在高速網路下是不可靠的，因為有很高比例的封包會被丟棄，而且交換型網路經常會妨礙NIDS看到的封包。NNIDS將NIDS的功能委托給單獨的主機，進而解決了高速網路和交換網路的問題。
- 產品：BlackICE Agent 、Tiny personal firewall with CMDS、ISS RealSecure Desktop Protector

NNIDS範例


NNIDS
NNIDS
NNIDS
FTP
WEB
MAIL
NNIDS
Internet
Router
Firewall
NNIDS
NNIDS控制台

IDS的偵測技術

- 基於特徵（Signature-**based**）
  - 維護一個入侵特徵的資料庫
  - 準確性較高
- 基於異常（Anomaly-**based**）
  - 統計模型
  - 專家系統
  - 誤報較多

IDS的限制

- 沒有主動防禦的能力：IDS只有告警的能力，無法主動防禦入侵行為。
- 誤報率偏高：目前多數的IDS利用特徵資料庫以判斷是否為入侵行為，但有些正常封包的特徵和入侵行為的特徵十分類似，但修改特徵資料庫之後又造成漏報。
- 漏報率偏高：目前的IDS系統還無法有效的識別出未知的入侵，也就是造成安全假象。

IDS的限制

- 性能普遍不足：現在市場上的IDS產品多依賴單一主機，因現今網路流量十分龐大，這種IDS產品已不能適應交換網路技術和高頻寬環境的發展，一旦資源耗盡，就無法運作了。
- 加密封包無法辨識：越來越多攻擊用加密封包，使得IDS監控網路流量的能力產生盲點，因IDS是擷取網路封包進行分析的，如果封包加密，就無法辨識其內容，也就無法進行分析。

何謂入侵防禦系統

- Intrusion Prevention System
- 可視為IDS功能的延伸，用以彌補IDS功能之不足
- 可主動偵測入侵行為並主動防禦
- 其餘的限制性與IDS相同

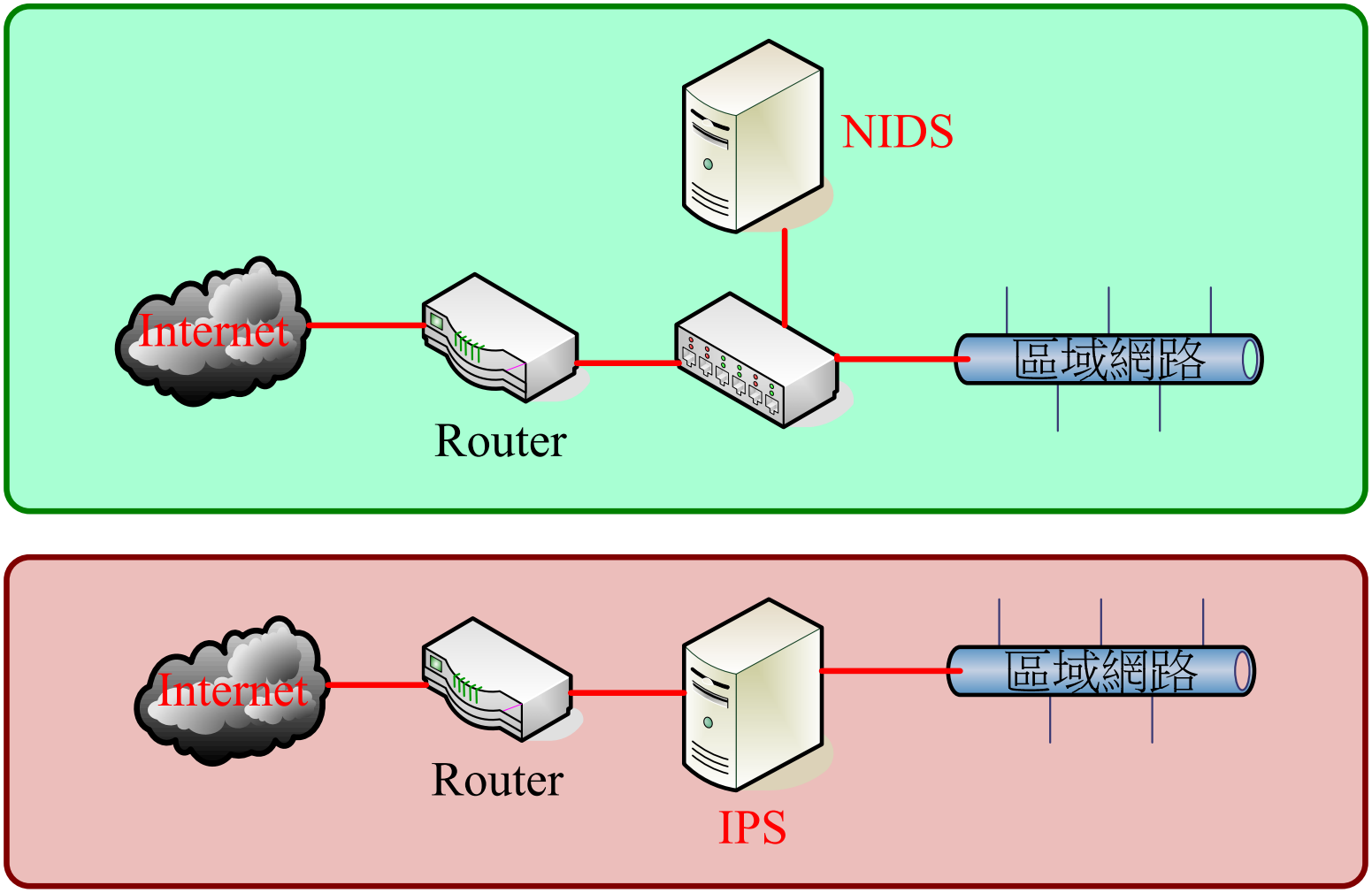
入侵防禦系統之種類

- 主機型入侵防禦（HIPS）：用於保護伺服器和主機系統不受入侵行為的攻擊
- 網路型入侵防禦（NIPS）：透過偵測流經的網路流量，提供對網路的安全保護，一旦辨識出入侵行為，NIPS就阻斷該網路連線
- 應用型入侵防禦（AIPS）：將主機型的入侵防禦擴展成為位於應用伺服器之前的資訊安全設備，主要針對應用程式的攻擊進行防禦。

入侵偵測系統及入侵防禦系統之差異

- 傳統的網路IDS(NIDS)系統用於被動地監測網路，根據規則資料庫和策略來尋找異常行為並提出告警訊息。如果NIDS突然出現故障，業務並不受影響，網路封包依然繼續流動，只是無法針對異常行為告警而已，故障對用戶是透明的。
- IPS系統是主動的在線設備，能丟棄攻擊的網路封包，或者在網路封包到達主機前切斷連線，如果出現故障，將影響到整個網路連線。

NIDS與IPS之差異



UTM簡介

- 集中式威脅管理(Unified Threat Management)
- NSS的定義
  www.nss.co.uk/utm/introduction.htm，UTM設備必須符合「可支援防火牆、VPN、IPS、內容過濾（Virus、Web、Mail、Spyware）等功能的單一設備」
- IDC（國際數據資訊）的定義，UTM設備至少要具備防火牆、VPN、ID&P和閘道防毒等4種功能

UTM優缺點

- 優點：整合多種功能在一台硬體設備，減少不同建案時期建置之設備所發生的設備衝突或不同廠商技術支援的整合性問題。
- 缺點：UTM設備強調多合一，產品功能之完整度與彈性，比不上單一功能的專屬設備。
- 產品：Fortinet FortiGate、ISS Proventia M10E、SonicWALL Pro4100 、友旺 SV2000 、ZyXEL ZyWALL 70 UTM、臨通 NS-720

UTM部署模式

- 路由模式(Routing Mode)：替換掉原有網路設備及資安設備
- 透通/橋接模式(Transparent/Bridge Mode)：不更動原有網路架構，保留原有資安設備
- 代理模式(Proxy Mode)：最能完整檢查網路流量，但DNS或員工電腦要配合修改設定值，同時傳輸效能一定會受到影響

Module 13-2入侵偵測實作(**)

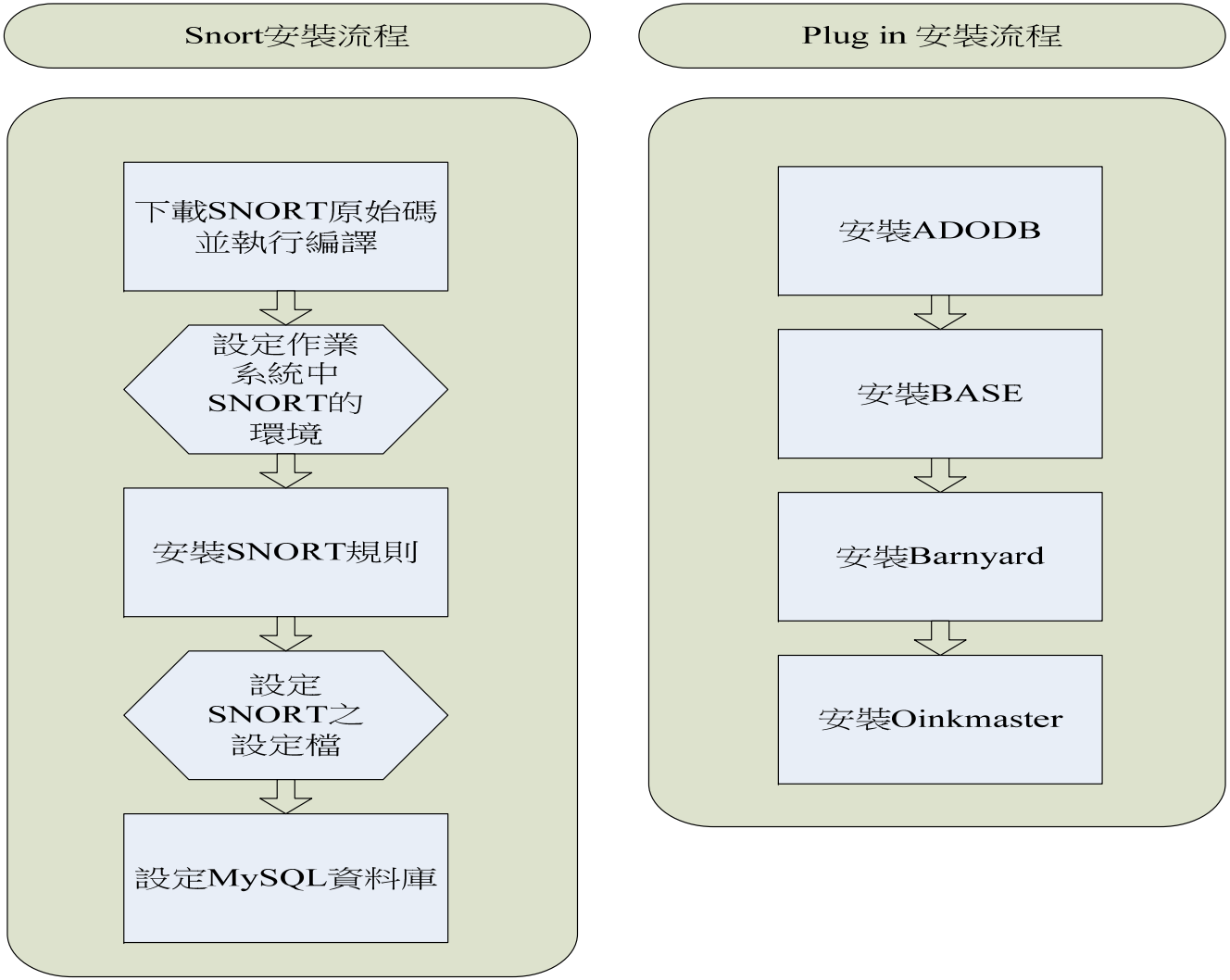
大綱

- Snort簡介
- 安裝前置作業
- 安裝snort
- Snort rules分類
- Snort.conf設定
- 安裝snort的plugin

建置IDS-以SNORT為例

- Snort是一套免費的、跨平台的NIDS，可用來偵測網路上的異常封包。
- 檢查所有經過的封包，並利用特徵比對的方式判斷是否有可能的入侵行為。
- 規則是使用開放的方式來發展的，所以可以自行加入偵測規則以加強入侵行為的偵測。
- SNORT官方網站：http://www.snort.org/

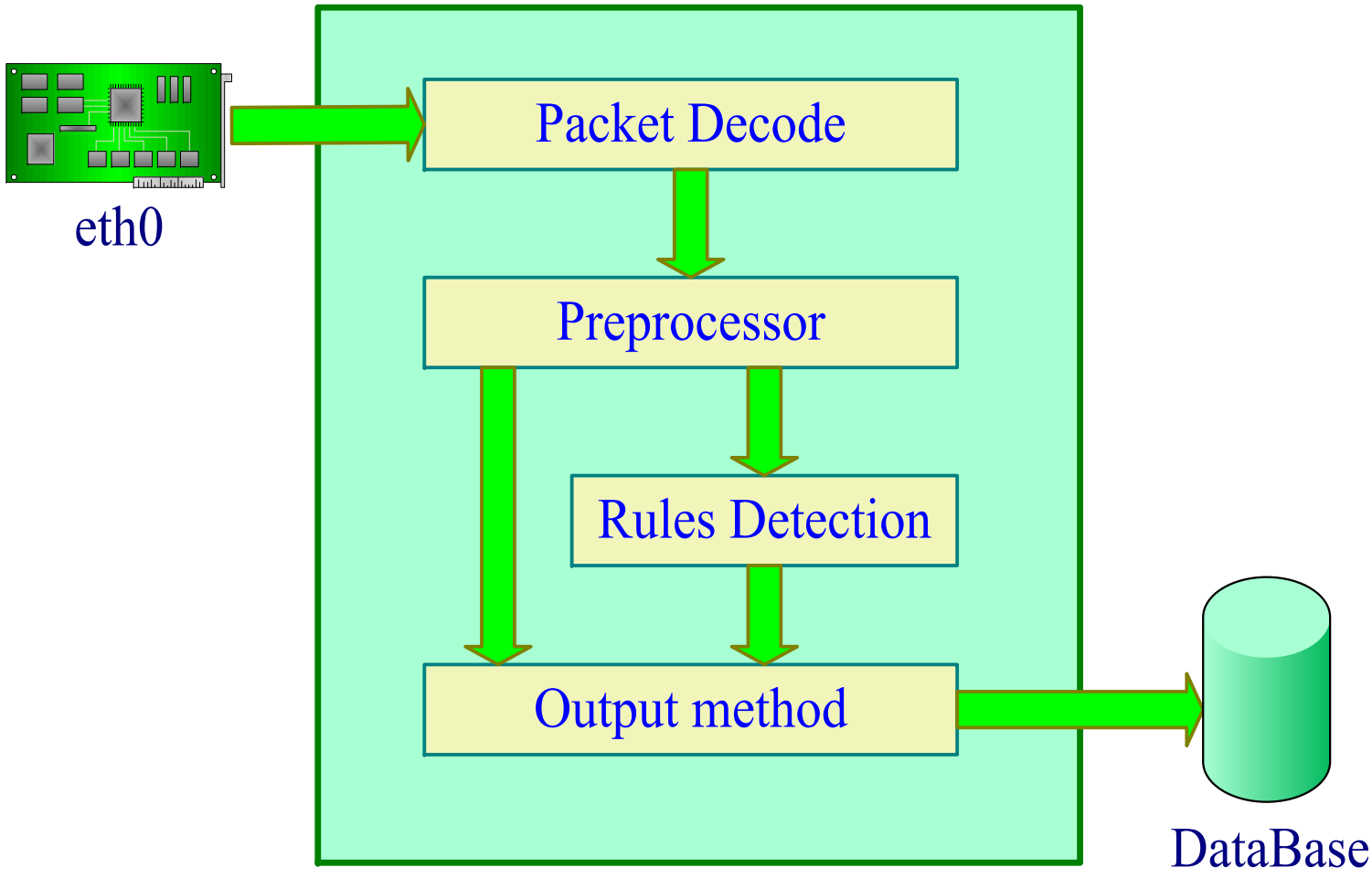
Snort 安裝流程


Snort安裝流程
下載SNORT原始碼
並執行編譯
設定作業
系統中
SNORT的
環境
安裝SNORT規則
設定
SNORT之
設定檔
設定MySQL資料庫
Plug in 安裝流程
安裝ADODB
安裝BASE
安裝Barnyard
安裝Oinkmaster

Snort執行的模式

- Sniffer mode：將封包擷取後顯示在螢幕上。
- Packet logger mode：將封包擷取後，存到硬碟中，可存成tcpdump格式。
- NIDS mode：讓snort分析擷取到的封包比對特徵資料庫以判別是否為入侵行為並告警。
- Inline mode：將snort當成IPS，從iptable讀取封包比對特徵資料庫後，告訴iptable是否要把它丟掉或是讓它通過。

Snort NIDS模式架構圖



Snort規則的回應動作

- alert：使用所選擇的告警，並將封包記錄下來
- log ：將封包記錄下來。
- pass ：讓封包通過不做任何事。
- activate ：使用所選擇的方式告警，並啟動另一個dynamic rules。
- dynamic ：保持idle直到被activate rules啟動，然後將封包記錄下來。

建置IDS-以SNORT為例

- Snort的Plugin
  - BASE(Basic Analysis and Security Engine) 是PHP-based的分析工具。
  - Barnyard是處理Snort unified file format的程式，可讓Snort專注於抓取及分析封包，無須浪費資源於處理LOG
  - Oinkmaster是用來Update及管理Snort Rules的程式

安裝前置作業

- 使用Fedora Core 5
- SELinux及iptables先停用
- 修改yum server位置，改至台灣的站台(以新竹縣教育網為例)，下載速度會比較快，如有缺少的RPM直接以YUM下載安裝，比較方便
  - 修改/etc/yum.repos.d/目錄下的
    - fedora-core.repo
    - fedora-extras.repo
    - fedora-updates.repo

安裝前置作業

#vi /etc/yum.repos.d/fedora-core.repo

baseurl=ftp://apt.nc.hcc.edu.tw/pub/fedora/linux/core/$releasever/$basearch/os/

#vi /etc/yum.repos.d/fedora-extras.repo

baseurl=ftp://apt.nc.hcc.edu.tw/pub/fedora/linux/extras/$releasever/$basearch/

#vi /etc/yum.repos.d/fedora-updates.repo

baseurl=ftp://apt.nc.hcc.edu.tw/pub/fedora/linux/core/updates/$releasever/$basearch/

安裝前置作業

- 安裝必要的RPM

  #yum –y update (更新RPM)

  #yum –y install mysql mysql-bench mysql-server mysql-devel mysqlclient10 php-mysql httpd gcc pcre-devel php-gd mod_ssl glib2-devel gcc-c++

  (安裝Snort需要的RPM)

- FC5依安裝方式不同會安裝不同的RPMs，通常FC5安裝好後只有mysql mysql-server php-mysql httpd等RPM，因此要安裝其它不足的部份。

安裝前置作業

- 開啟apache及mysql服務
  - chkconfig --level 345 httpd on
  - chkconfig --level 345 mysqld on
  - service httpd start
  - service mysqld start
- 測試Apache的PHP是否成功
  - vi /var/www/html/test.php
  - <?php echo phpinfo() ?>

Apache PHP已成功

- http://localhost/test.php

PHP Version 5.1.4

System	Linux linux.david999.idv.tw 2.6.15-1.2054_FC5 #1 Tue Mar 14 15:48:33 EST 2006 i686
Build Date	May 8 2006 08:43:39
Configure Command	'./configure' '--build=i386-redhat-linux' '--host=i386-redhat-linux' '--target=i386-redhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib' '--libexec localstatedir=/var' '--sharedstatedir=/usr/com' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=../config.cache' '--with-config-file-path=/etc' '--with-config-file-scan-dir=/etc/php.d' '--disable-debug' '--with-pic' '--disable-rpath' '--without-pear' '--with' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--enable-gd-native-ttf' '--without-gdbm' '--with-gettext' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-png' '--with-pspell' '--with-expat-dir=/usr' '--with-pcre-regex=/usr' '--with-zlib' '--wi enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-sockets' '--enable-sysvsem' '--enable-sysvshm' '--enable-sysvmsg' '- vars' '--enable-trans-sid' '--enable-yp' '--enable-wddx' '--with-kerberos' '--enable-ucd-snmp-hack' '--with-unixODBC=shared,/usr' '- memory-limit' '--enable-shmop' '--enable-calendar' '--enable-dbx' '--enable-dio' '--with-mime-magic=/etc/httpd/conf/magic' '--with libxml-dir=/usr' '--with-xml' '--with-apxs2=/usr/sbin/apxs' '--without-mysql' '--without-gd' '--without-odbc' '--disable-dom' '--disable- unixODBC' '--disable-pdo' '--disable-xmlreader' '--disable-xmlwriter'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php.ini

開始安裝Snort

#cd /usr/local/src

#wget http://www.snort.org/dl/current/snort-2.6.0.tar.gz

#tar －xvzf snort-2.6.0.tar.gz

#cd snort-2.6.0

#./configure --with-mysql --enable-dynamicplugin

#make

#make install

設定SNORT環境

```
#groupadd snort
#useradd –g snort snort –s /sbin/nologin
#mkdir /etc/snort
#mkdir /var/log/snort
#chown snort:snort /var/log/snort
#cp -ar /usr/local/src/snort-2.6.0/etc/*  /etc/snort
```

Snort Rules分類

- VRT rules 為snort.org的官方rules，由Sourcefire Vulnerability Research Team (VRT)提供，每一條rules均經VRT嚴格測試。
  - subscription release 須付費，即時更新
  - registered user release 只要在snort.org註冊即可下載，比subscription release 晚5天
  - unregistered user release 不定期發布
- Community Rules 由開放原始社群提供，VRT僅提供基本的測試。

Sourcefire VRT Certified Rules - The Official Snort Ruleset (registered user release)

The Sourcefire VRT has added multiple rules to detect attempts to exploit vulnerabilities in the Microsoft DHCP Client Service, Microsoft Excel and Microsoft Word.

view advisory | view changelog

VRT Certified Rules for Snort CURRENT	*(snortrules-snapshot-CURRENT.tar.gz)*	Download
MD5: 7e0bb11cc27332ee6974820bf1246821	VER: CURRENT	RELEASED: 2006-08-02
VRT Certified Rules for Snort v2.4	*(snortrules-snapshot-2.4.tar.gz)*	Download
MD5: e84922df35a4e2107223550cc3a19ae1	VER: 2.4	RELEASED: 2006-08-02
VRT Certified Rules for Snort v2.3	*(snortrules-snapshot-2.3.tar.gz)*	Download
MD5: d1ee05fec27c6679c4287fc017f00fd6	VER: 2.3	RELEASED: 2006-08-02
VRT Certified Rules for Snort v2.2	*(snortrules-snapshot-2.2.tar.gz)*	Download
MD5: ae26936e423f5e0826711b6ff6a54165	VER: 2.2	RELEASED: 2006-08-02
VRT Certified Rules for Snort v2.1	*(snortrules-snapshot-2.1.tar.gz)*	Download
MD5: 6e3ab807e44aa3a03f86dd7170b1fd6b	VER: 2.1	RELEASED: 2006-08-02
VRT Certified Rules for Snort v2.0	*(snortrules-snapshot-2.0.tar.gz)*	Download
MD5: fa809e1ef3ac4718f4265337f84c7ded	VER: 2.0	RELEASED: 2006-08-02

安裝Snort Rules

- 使用VRT rules (registered user release)
- 下載snortrules-snapshot-CURRENT.tar.gz至/etc/snort中

#tar xvzf snortrules-snapshot-CURRENT.tar.gz

- 會自動建立rules的目錄，裡面就是放snort的rules

Snort Configure設定步驟

- 設定區域網路變數
- 設定動態載入函式庫
- 設定預處理程序
- 設定輸出
- 設定命令模式中的選項
- 設定使用之規則

設定snort.conf

- 剛才把/usr/local/src/snort-2.6.0/etc/內所有的檔案拷貝到/etc/snort/下，這時預設的snort.conf也一併拷到/etc/snort/中了。
- 依照區域網路環境設定snort.conf
  - var HOME_NET 192.168.123.0/24(使用CIDR設定內部網路，如有多個子網路，中間使用逗號分隔)
  - var EXTERNAL_NET !$HOME_NET(不屬內部網路的都算是外部網路)

設定snort.conf

- var RULE_PATH /etc/snort/rules(設定rules放置的位置，請使用絕對位置以避免出錯)
- preprocessor stream4_reassemble:both,ports default(開啟Stateful Inspection狀態式過濾檢查、TCP串流重組以偵測IDS Evasion)
- output database:log,mysql,user=snort password=*password* dbname=snort host=localhost(將LOG存入MySQL資料庫中)

設定MySQL

#mysql　進入mysql介面

mysql>set password for
root@localhost=password(‘your*password*’);

設定root密碼

mysql>create database snort;　建立snort資料庫

mysql>grant INSERT,SELECT on root.* to
snort@localhost;

讓snort針對root的物件只有insert及select的權利

設定MySQL

mysql>set password for
snort@localhost=password(‘*yourpassword*’
);

設定snort的密碼

mysql>grant
CREATE,INSERT,SELECT,DELETE,UPDATE
on snort.* to snort@localhost;

讓snort這個帳號對snort的所有物件有
CREATE,INSERT,SELECT,DELETE,UPDATE的
權利

mysql>exit 離開mysql介面

設定MySQL

#mysql －u root －p < /usr/local/src/snort-2.6.0/schemas/create_mysql snort

Enter password:*yourpassword*

建立snort資料庫內的table schema

#mysql －p

Enter password:*yourpassword*

已設定root密碼，因此需要密碼進入mysql介面

設定MySQL

mysql>show databases;

看有沒有snort這個資料庫

mysql>use snort;

開啟snort資料庫

mysql>show tables;

看snort資料庫有那些table，如果是空的話，請重建立table schema

測試

- 建立測試的rules，在/etc/snort/rules/local.rules中加入以下一行
  - Alert tcp any any -> any any (msg:”test”; sid:1000001;)
- 啟動snort

#/usr/local/bin/snort －Dq －u snort －g snort －c /etc/snort/etc/snort.conf

在/var/log/messages會說明eth0進入雜亂模式及初始化成功的訊息

```
Aug 10 23:28:38 linux snort[2481]: Var 'eth0_ADDRESS' defined, value len = 27 ch
ars
Aug 10 23:28:38 linux snort[2481]: , value = 192.168.123.0/255.255.255.0
Aug 10 23:28:38 linux kernel: device eth0 entered promiscuous mode
Aug 10 23:28:38 linux kernel: audit(1155223718.812:11): dev=eth0 prom=256 old_pr
om=0 auid=4294967295
Aug 10 23:28:38 linux snort[2481]: Initializing daemon mode
Aug 10 23:28:38 linux kernel: device eth0 left promiscuous mode
Aug 10 23:28:38 linux kernel: audit(1155223718.824:12): dev=eth0 prom=0 old_prom
=256 auid=4294967295
Aug 10 23:28:38 linux kernel: device eth0 entered promiscuous mode
Aug 10 23:28:38 linux kernel: audit(1155223718.888:13): dev=eth0 prom=256 old_pr
om=0 auid=4294967295
Aug 10 23:28:38 linux snort[2482]: Var 'eth0_ADDRESS' redefined
Aug 10 23:28:38 linux snort[2482]: PID path stat checked out ok, PID path set to
 /var/run/
Aug 10 23:28:38 linux snort[2482]: Writing PID "2482" to file "/var/run//snort_e
th0.pid"
Aug 10 23:28:38 linux snort[2482]: Daemon initialized
Aug 10 23:28:38 linux snort[2481]: Daemon parent exiting
Aug 10 23:28:42 linux snort[2482]: Snort initialization completed successfully (
pid=2482)
Aug 10 23:28:42 linux snort[2482]: Not Using PCAP_FRAMES
```

在MySQL中的資料庫的event也會增加

```
[root@linux log]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 5.0.22

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use snort
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select count(*) from event
    -> ;
+----------+
| count(*) |
+----------+
|    12397 |
+----------+
1 row in set (0.00 sec)

mysql>
```

安裝ADODB

- ADODB 是 Active Data Objects Data Base 的簡稱，它是一種 PHP 存取資料庫的函式元件

#cd /usr/local/src

#wget http://nchc.dl.sourceforge.net/sourceforge/adodb/adodb491.tgz

#cd /var/www

#tar －xzvf /usr/local/src/adodb491.tgz

安裝BASE

#cd /usr/local/src

#wget
http://nchc.dl.sourceforge.net/sourceforge/secureideas/base-1.2.6.tar.gz

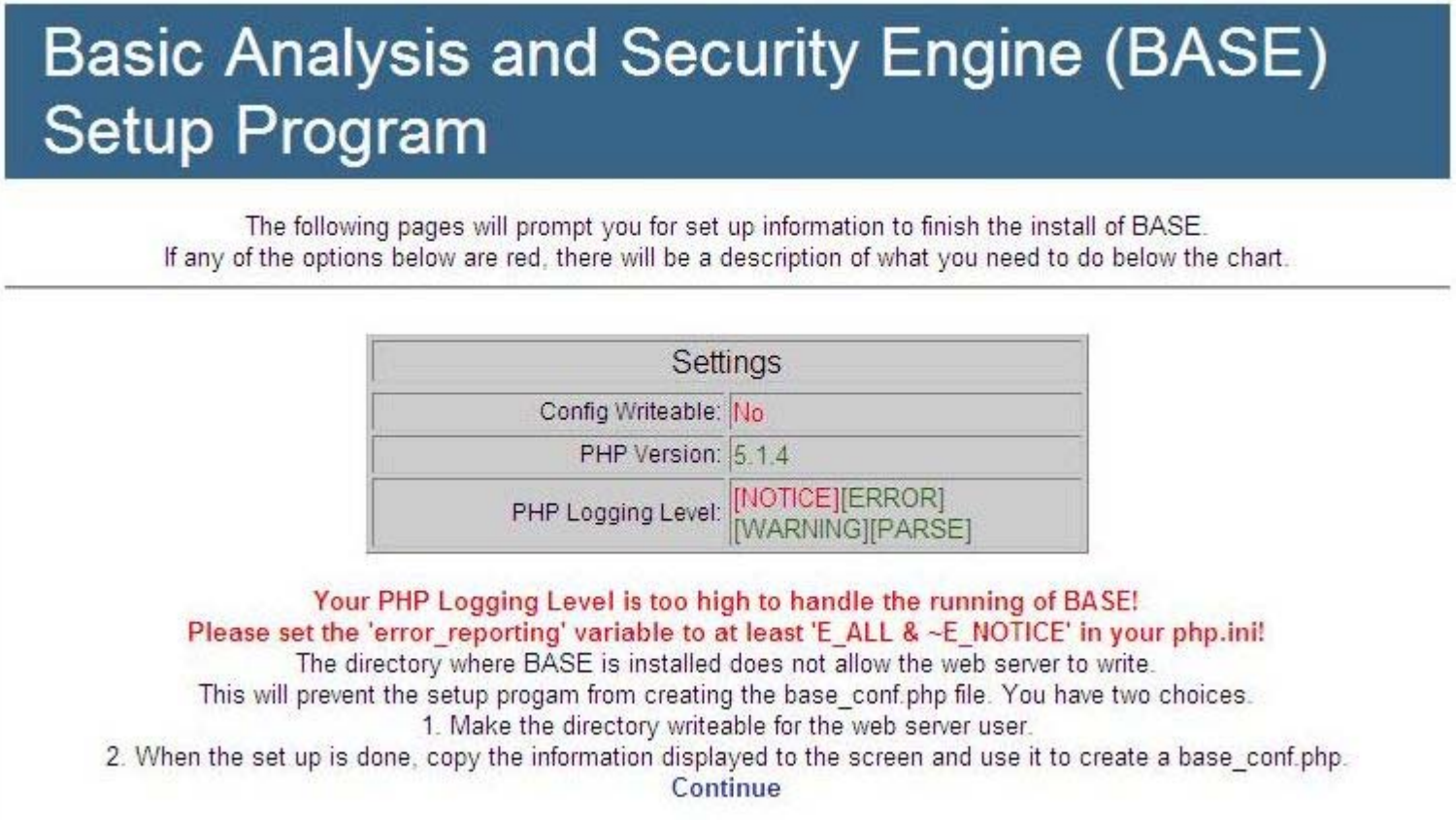
#cd /var/www

#tar －xvzf /usr/local/src/base-1.2.6.tar.gz

#mv base-1.2.6 base

https://192.168.123.22/base

- 會有錯誤訊息



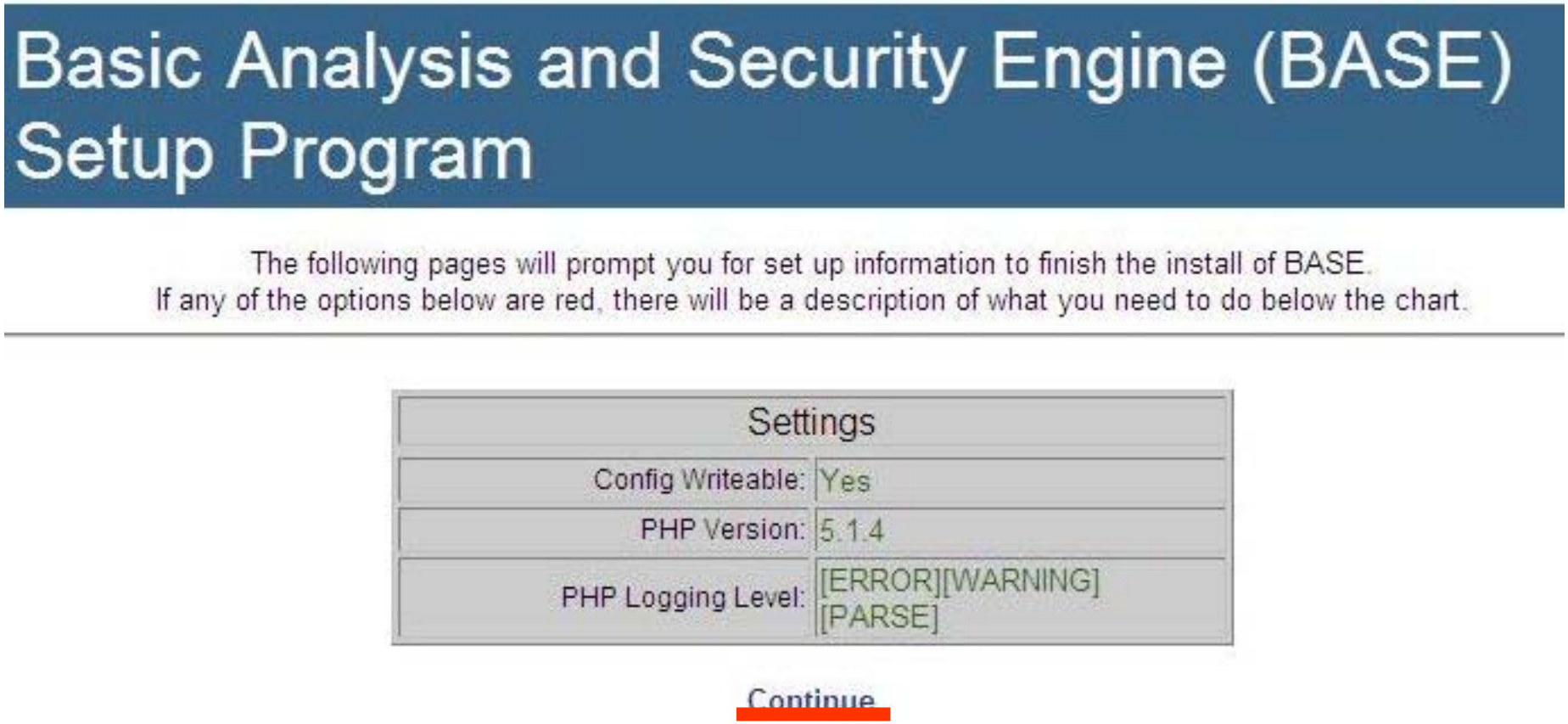
依錯誤訊息修改設定

- 在/etc/php.ini中將error_reporting = E_ALL 改成 error_reporting = E_ALL & ~E_NOTICE
- 在/var/www/html目錄下執行

#chmod 777 base

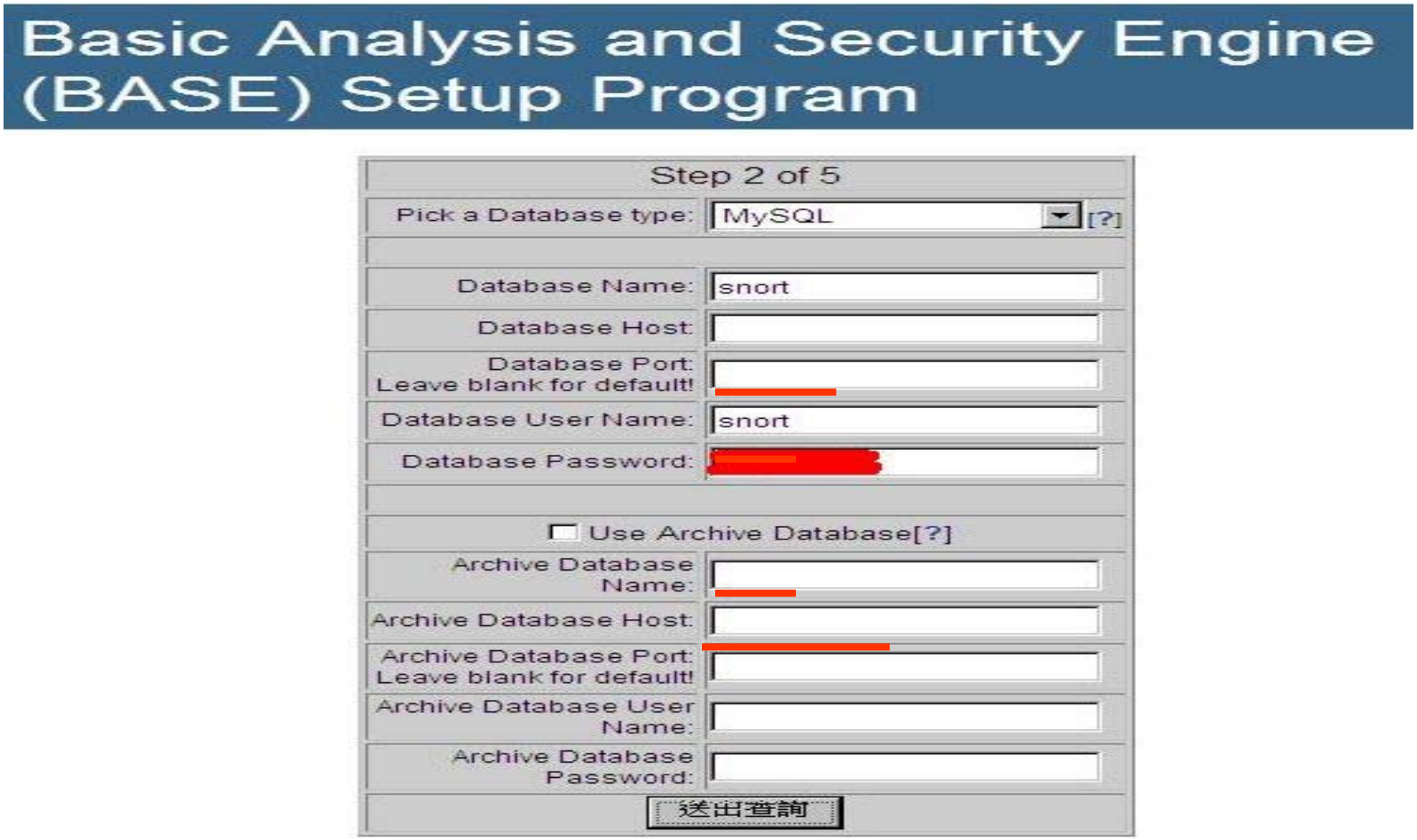
- 這是暫時性的，讓BASE的設定檔可經由WEB寫入，設定完記得要chmod 755 base，將權限拿回來。

這時就不會有錯誤訊息了



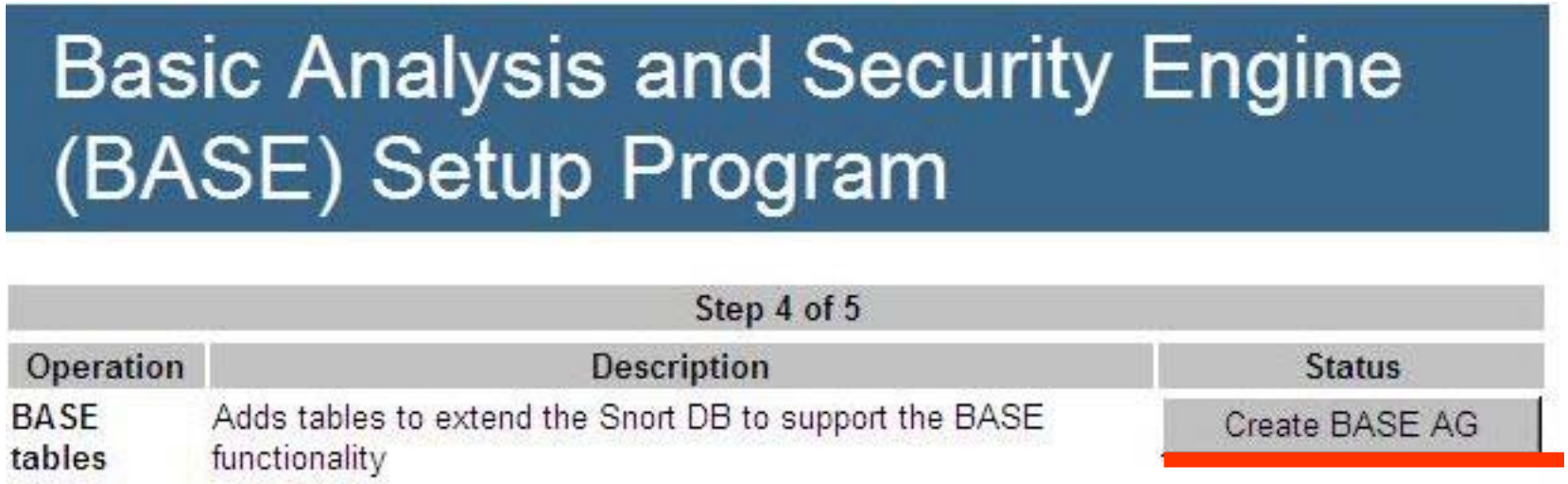
設定語系及ADODB位置

設資料庫型態、名稱、使用者名稱及密碼



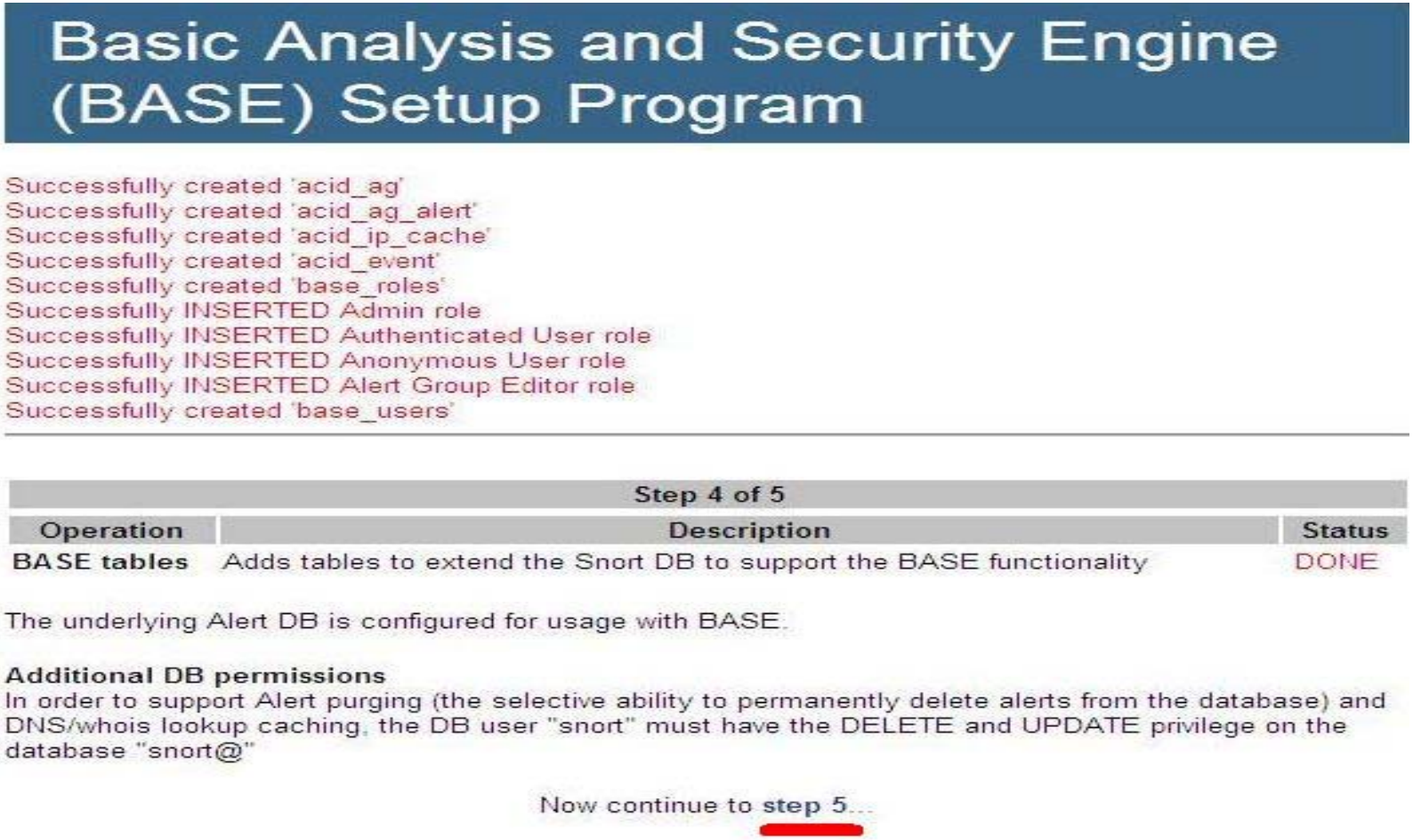
Basic Analysis and Security Engine (BASE) Setup Program

於snort資料庫中建立BASE所需的table



- 會在/var/www/html/base中建立一個base_conf.php檔放置BASE的一些設定

說明建立那些table



Basic Analysis and Security Engine (BASE) Setup Program

Successfully created 'acid_ag'
Successfully created 'acid_ag_alert'
Successfully created 'acid_ip_cache'
Successfully created 'acid_event'
Successfully created 'base_roles'
Successfully INSERTED Admin role
Successfully INSERTED Authenticated User role
Successfully INSERTED Anonymous User role
Successfully INSERTED Alert Group Editor role
Successfully created 'base_users'

Step 4 of 5

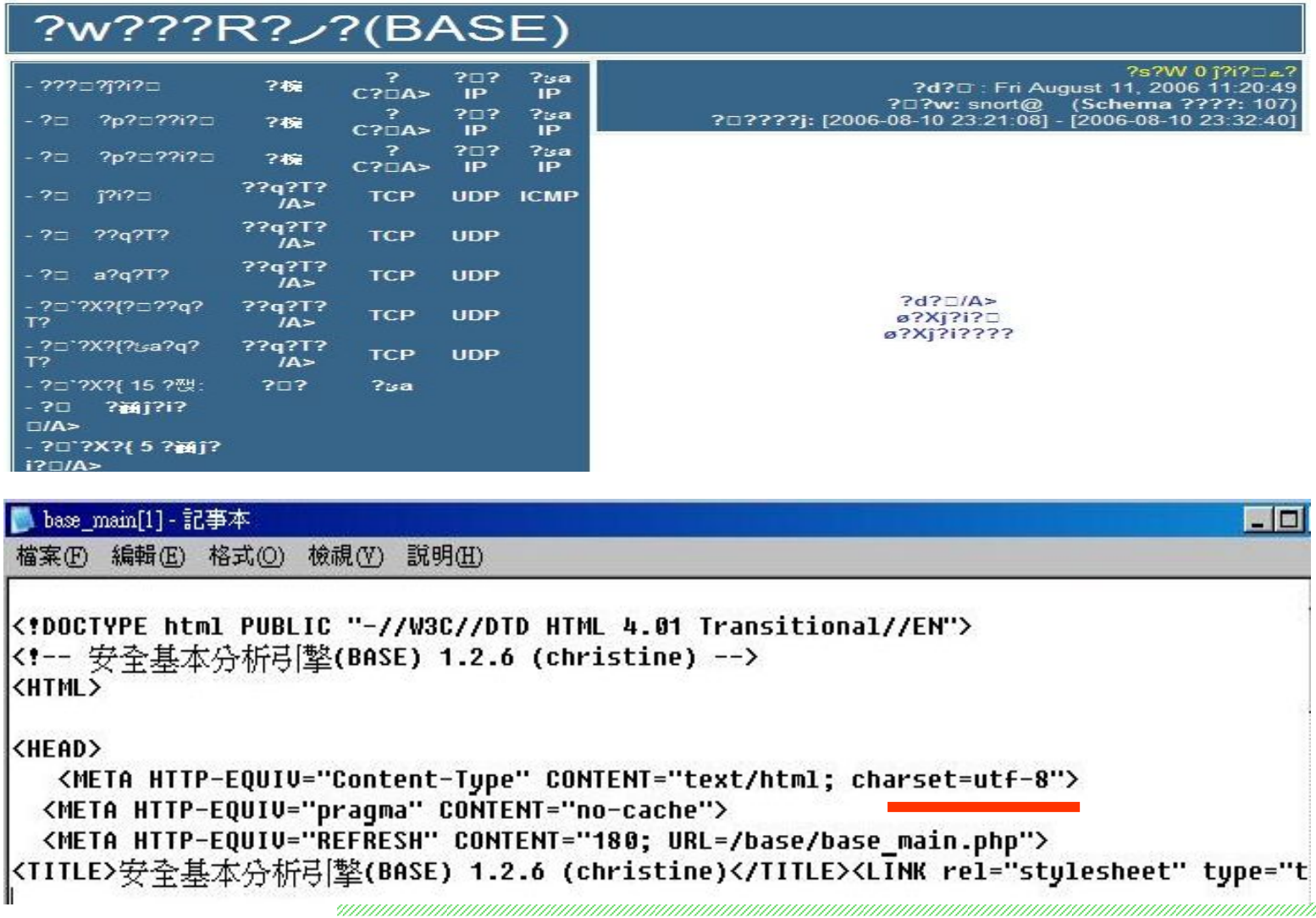
Operation	Description	Status
BASE tables	Adds tables to extend the Snort DB to support the BASE functionality	DONE

The underlying Alert DB is configured for usage with BASE.

Additional DB permissions
In order to support Alert purging (the selective ability to permanently delete alerts from the database) and DNS/whois lookup caching, the DB user "snort" must have the DELETE and UPDATE privilege on the database "snort@"

Now continue to step 5...

使用中文語系會有亂碼


?w???R?ノ?(BASE)
?s?W 0 j?i?□e?
?d?□ : Fri August 11, 2006 11:20:49
?□?w: snort@ (Schema ????: 107)
?□????j: [2006-08-10 23:21:08] - [2006-08-10 23:32:40]
?d?□/A>
ø?Xj?i?□
ø?Xj?i????
base_main[1] - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!-- 安全基本分析引擎(BASE) 1.2.6 (christine) -->
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
<META HTTP-EQUIV="pragma" CONTENT="no-cache">
<META HTTP-EQUIV="REFRESH" CONTENT="180; URL=/base/base_main.php">
<TITLE>安全基本分析引擎(BASE) 1.2.6 (christine)</TITLE><LINK rel="stylesheet" type="t

亂碼原因

- 這是因為預設BASE的繁體中文語系是用Big5編碼的，但是在CHARSET中卻設成UnicodeUTF-8編碼，因此瀏覽器預設會以UTF-8編碼，造成瀏覽器誤判，而成亂碼現象，這時只要修正瀏覽器的編碼方式，或修改BASE的預設CHARSET，在/var/www/html/base/languages/chinese.lang.php中的DEFINE(‘_CHARSET’,’big5’);，瀏覽器就會預設以big5碼來編碼。

基本上就安裝完成了,也支援中文

點選[繪出警告資料]時出現錯誤

解決方式

- 這是因為有些Graphic Library沒有安裝，安裝即可，方式如下:

#pear install Image_Graph-alpha Image_Canvas-alpha Image_Color Numbers_Roman

安裝Barnyard

#cd /usr/local/src

#wget
http://nchc.dl.sourceforge.net/sourceforge/barnyard/barnyard-0.2.0.tar.gz

#tar –xvzf barnyard-0.2.0.tar.gz

#cd barnyard-0.2.0

#./configure --enable-mysql

安裝Barnyard

#make

#make install

#cp /usr/local/src/barnyard-0.2.0/etc/barnyard.conf /etc/snort

#vi /etc/snort/barnyard.conf (修改以下設定)

Configure hostname:linux

Config interface:eth0

Output log_acid_db: mysql, database snort, server localhost, user snort, password *yourpassword*,detail full

建立檢查點

#cd /etc/snort

#vi bylog.waldo (內容如下)

/var/log/snort

snort.log

1155273392

0

第三行是snort.log的時間戳記

第四行是表示從第0筆紀錄開始塞進mysql

啟動barnyard

```
#/usr/local/bin/barnyard -c
   /etc/snort/barnyard.conf -g /etc/snort/gen-
   msg.map -s /etc/snort/sid-msg.map -d
   /var/log/snort -f snort.log -w
   /etc/snort/bylog.waldo &
```

可能發生的錯誤

```
[root@linux snort]# /usr/local/bin/barnyard -c /etc/snort/barnyard.conf -g /etc/
snort/gen-msg.map -s /etc/snort/sid-msg.map -d /var/log/snort -f snort.log -w /e
tc/snort/bylog.waldo &
[1] 4387
Barnyard Version 0.2.0 (Build 32)
[root@linux snort]# Opened spool file '/var/log/snort/snort.log.1155223262'
ERROR: No input plugin found for magic: a1b2c3d4
Fatal Error, Quitting..
Exiting

[1]+  Exit 1                  /usr/local/bin/barnyard -c /etc/snort/barnyard.con
f -g /etc/snort/gen-msg.map -s /etc/snort/sid-msg.map -d /var/log/snort -f snort
.log -w /etc/snort/bylog.waldo
[root@linux snort]#
```

除錯及測試

- 這是因為放在/var/log/snort/中的snort.log不是unified格式，可能是之前有開過snort而那時設的output不是unified格式，把它們刪除再重新執行就好了。

#rm –rf /var/log/snort/*

- 測試log有沒有塞進mysql中

#mysql –uroot –pyourpassword –D snort –e "select count(*) from event"

設定Snort及Barnyard開啟順序

- 先啟動snort再啟動barnyard，可以以下做成script，於系統開機時執行

```
#!/bin/bash
/usr/local/bin/snort –Dq –u snort –g snort –c
   /etc/snort/snort.conf
/usr/local/bin/barnyard –c
   /etc/snort/barnyard.conf –g /etc/snort/gen-
   msg.map –s /etc/snort/sid-msg.map –d
   /var/log/snort –f snort.log –w
   /etc/snort/bylog.waldo &
```

oinkmaster

- 由於Snort的rules像掃毒程式一樣需要定期更新，而每次下載新的rules後再手動修復之前的設定又十分麻煩，因此有一個免費工具Oinkmaster可以幫你做到這一切。
- 可定義那些rules需要更新而哪些rules需要跳過；哪些SIDs需要修改；哪些SIDs應該啟用而那些SIDs應該禁用。

安裝Oinkmaster

#cd /usr/local/src

#wget
http://nchc.dl.sourceforge.net/sourceforge/oinkmaster/oinkmaster-2.0.tar.gz

#tar －xvzf oinkmaster-2.0.tar.gz

#cd oinkmaster-2.0

#cp oinkmaster.pl /usr/local/bin

#cp oinkmaster.conf /etc

修改oinkmaster設定檔

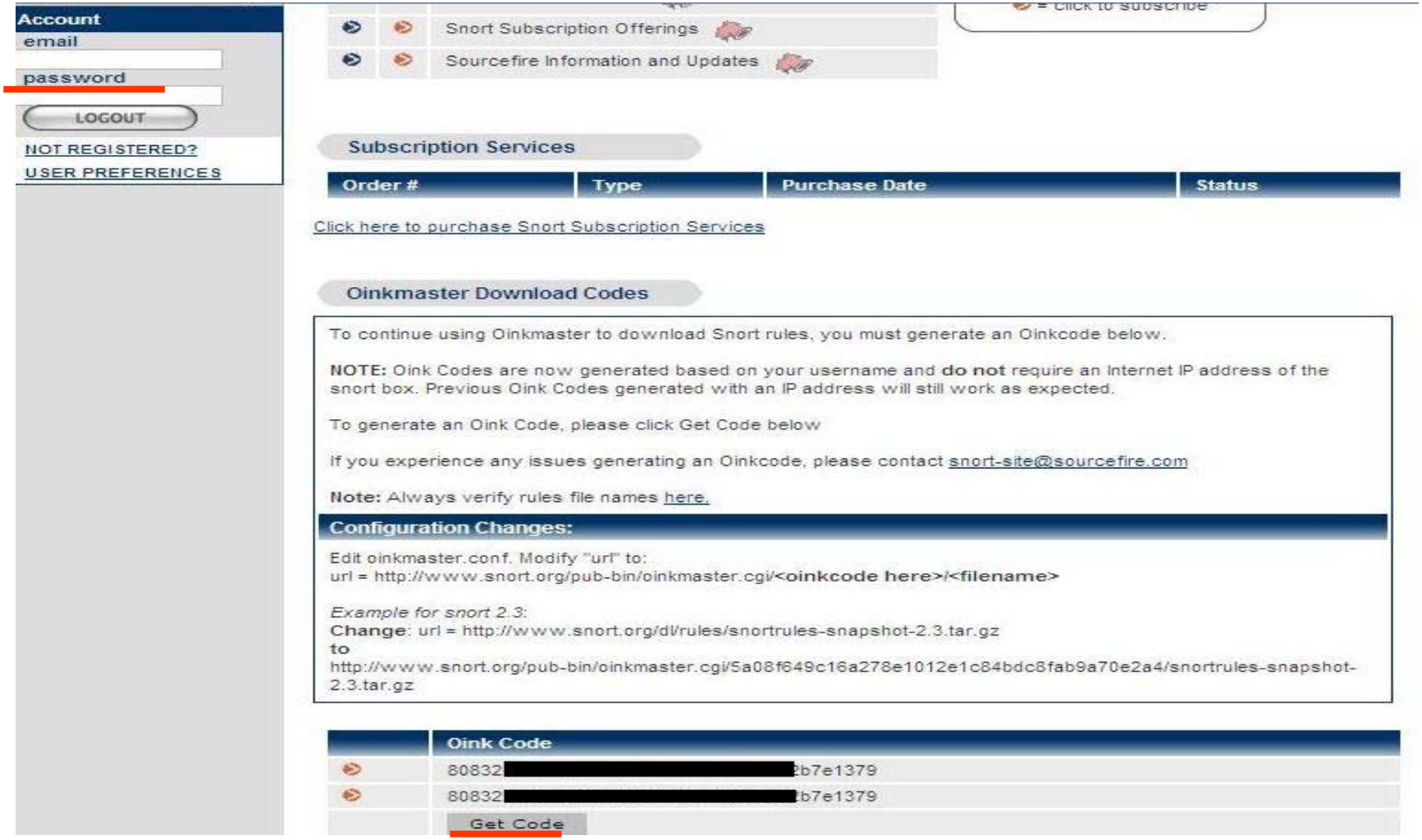
#vi /etc/oinkmaster.conf

url = http://www.snort.org/pub-bin/oinkmaster.gci/[oink code]/snortrules-snapshot-CURRENT.tar.gz

#chown –R snort:snort /etc/snort/rules

- 何謂oink code：因為現行snort的規則改以VRT方式發行，須要註冊才可下載使用，因此使用oink code方式，讓註冊的人可以得到一組編碼，以利用oinkmaster下載rules

取得Oink Code


Account
email
password
LOGOUT
NOT REGISTERED?
USER PREFERENCES
Snort Subscription Offerings
Sourcefire Information and Updates
Subscription Services
Order #
Type
Purchase Date
Status
Click here to purchase Snort Subscription Services
Oinkmaster Download Codes
To continue using Oinkmaster to download Snort rules, you must generate an Oinkcode below.
NOTE: Oink Codes are now generated based on your username and do not require an Internet IP address of the snort box. Previous Oink Codes generated with an IP address will still work as expected.
To generate an Oink Code, please click Get Code below
If you experience any issues generating an Oinkcode, please contact snort-site@sourcefire.com
Note: Always verify rules file names here.
Configuration Changes:
Edit oinkmaster.conf. Modify "url" to:
url = http://www.snort.org/pub-bin/oinkmaster.cgi/<oinkcode here>/<filename>
Example for snort 2.3:
Change: url = http://www.snort.org/dl/rules/snortrules-snapshot-2.3.tar.gz
to
http://www.snort.org/pub-bin/oinkmaster.cgi/5a08f649c16a278e1012e1c84bdc8fab9a70e2a4/snortrules-snapshot-2.3.tar.gz
Oink Code
80832 [redacted] b7e1379
80832 [redacted] b7e1379
Get Code

Oinkmaster的參數

- -o <OUTPUT_DIRECTORY>：指定下載完後的rules要放置於那個目錄。
- -b <BACKUP_DIRECTORY>：指定備份之rules要放置於那個目錄，oinkmaster會把舊的rules製做成檔名為rules-backup-2006XXXX-0919XX.tar.gz的備份檔，並放置於本目錄。
- 可以把oinkmaster放在crontab裡定時執行

Module 13-3入侵防禦實作(***)

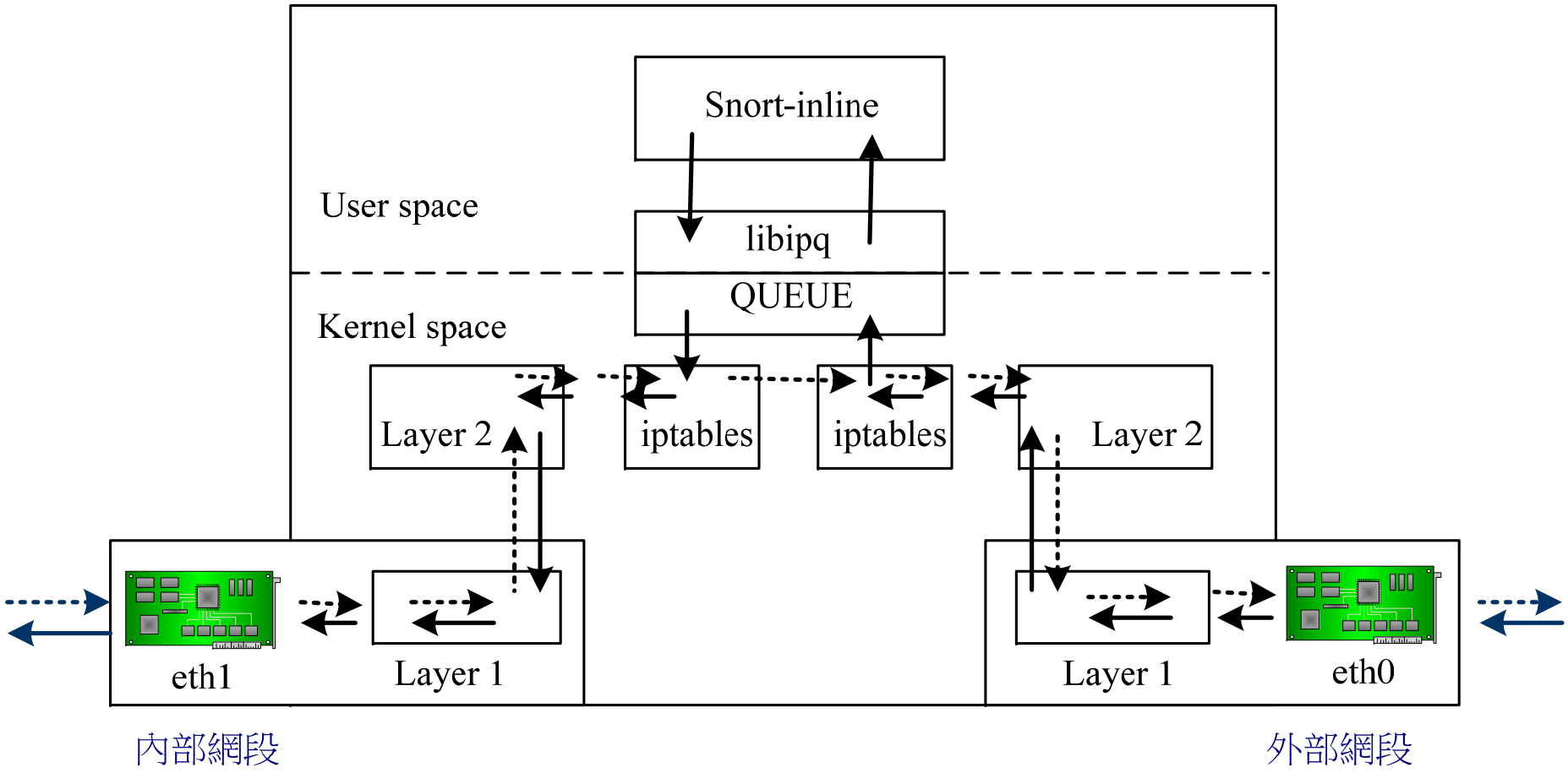
大綱

- Snort-inline簡介
- Snort-inline運作模式
- Snort-inline新增的規則行動
- Bridge mode linux
- Snort-inline安裝前準備
- 安裝Snort-inline
- 修改rules-Snortconfig介紹

Snort-inline IPS簡介

- Snort-inline是由snort 這個著名的免費入侵偵測系統所修改而來的，snort是從libpcap來擷取網路上的封包，snort-inline為了與IPtables溝通，因此它是用libipq從IPtables的QUEUE中擷取封包資料，比對入侵偵測規則後，就可知道這是不是入侵行為，然後利用新的規則型式告訴IPtables這些封包是否要被丟棄，或是讓它通過。

Snort-inline 運作模式



Snort-inline 新增的回應動作

- drop：告訴iptables丟棄封包，並透過snort的方式記錄封包。
- reject：告訴iptables丟棄封包，並透過snort的方式記錄封包；當是TCP協定時送一個TCP reset回去，如果是UDP協定的話，送一個icmp port unreachable回去。
- sdrop：告訴iptables丟棄封包，並不記錄封包。

Snort-inline 新增的規則行動

- replace：在Snort-inline規則的回應動作選項裡所使用，可以讓rules修改封包內容，不過修改內容的資料長度要和原始封包的內容資料長度一樣。
- 範例:

Alert tcp any any <> any 80 (msg: “tcp replace”; connect:”GET”; replace:”BET”;)

Snort-inline將尋找TCP 80port封包內容有GET的, 並且以BET來替代GET

Bridge Mode Linux

- Bridge Mode Linux最早是在2.2版的Kernel出現的，現在2.6版的Kernel大部份的Linux套件都支援
- 使用Bidge Mode的好處
  - 可以隱藏防火牆
  - Bridge Mode是沒有IP的，可以確保設備不會被攻擊者發現。

安裝bridge工具程式

#yum install bridge-utils

- 先將欲加入的網路卡IP清除掉

#ifconfig eth0 0.0.0.0 up

#ifconfig eth1 0.0.0.0 up

- 建立新的bridge

#brctl addbr ***bridgename***

```
[root@[redacted] root]# ifconfig
br0       Link encap:Ethernet  HWaddr 00:40:95:30:2A:98
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:55609 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2578615 (2.4 Mb)  TX bytes:0 (0.0 b)
```

Bridge工具程式用法介紹

- 顯示bridge的狀態：會顯示bridge的狀態、ID及bridge內所有的網卡。

#brctl show

```
[root@[redacted] root]# brctl show
bridge name     bridge id               STP enabled     interfaces
br0             8000.004095302a98       no              vmnet1
                                                        eth1
```

Bridge工具程式用法介紹

- 刪除bridge

#brctl delbr ***bridgename***

- 新增網卡到bridge：不能將一張網路卡加到多個bridge裡面，也不能將bridge加到bridge裡

#brctl addif ***bridgename interface-name***

- 從bridge刪除網卡

#brctl delif ***bridgename interface-name***

Bridge工具程式用法介紹

- 顯示bridge中的MAC值：會顯示流入流出的MAC值

#brctl showmacs ***bridgename***

```
[root@[redacted]root]# brctl showmacs br0
port no  mac addr                is local?       ageing timer
  2      00:00:1c:[redacted]       no              174.03
  2      00:00:1c:[redacted]       no                4.49
  2      00:03:ba:[redacted]       no              174.42
  2      00:10:b5:[redacted]       no                1.90
  2      00:20:ed:[redacted]       no                1.05
  2      00:20:ed:[redacted]       no                1.05
  2      00:20:ed:[redacted]       no                0.36
  2      00:20:ed:[redacted]       no                0.36
  2      00:20:ed:[redacted]       no                0.98
  2      00:20:ed:[redacted]       no                0.98
  2      00:40:95:[redacted]       yes               0.00
  1      00:50:56:[redacted]       yes               0.00
  2      08:00:20:[redacted]       no              174.81
```

Bridge工具程式用法介紹

- 給予bridge網路位址：這是額外的選項，預設bridge是沒有IP的，如果為了管理的目的，可以賦予bridge IP位址

#ifconfig ***bridgename IP*** netmask ***netmask*** up

Snort-Inline 安裝前準備

- 安裝kernel source：在complier時會使用到kernel的source code，因此要先安裝kernel的source code。

#yum –y install kernel-devel

- glibc參考的kernel header與現在kernel使用的不同,因此要將/usr/include/linux連結到現在kernel的header上

#mv /usr/include/linux /usr/include/linux.orig

#ln –s /usr/src/kernels/2.6.17-1.2174_FC5-i686/include/linux /usr/include/linux

Snort-Inline 安裝前準備

- 安裝PCRE:snort及snort-inline在2.1.0版之後，rules中有以pcre（perl compatible regular express）寫成的rule，因此要安裝pcre才可正確編譯

#yum －y install pcre

Snort-Inline 安裝前準備

- 安裝iptables：snort-inline是用libipq從iptables的queue中擷取資料，因此在編譯snort-inline時需要“libipq.a” and “libipq.h”這兩個檔案，在一般linux中是放在iptables-devel這個RPM中，因此要先安裝起來，讓complier時可以使用

#yum –y install iptables-devel

Snort-Inline 安裝前準備

- 安裝libnet：這是snort-inline在使用新的規則行動-reject，當是TCP協定時產生一個TCP reset封包，如果是UDP協定的話產生icmp port unreachable封包時所需使用的函式庫，要注意的是snort-inline使用的是1.0.x版，使用最新版1.1.X版與snort-inline並不相容。

Snort-Inline 安裝前準備

#cd /usr/local/src

#wget
http://www.packetfactory.net/libnet/dist/deprecated/libnet-1.0.2a.tar.gz

#tar –xvzf libnet-1.0.2a.tar.gz

#cd libnet-1.0.2a

#./configure

#make

#make install

Snort-Inline 安裝前準備

- 安裝libpcap：snort-inline同時也保留了snort的功能，可利用libpcap從網路介面抓取封包資料並存成tcpdump的格式。

#yum –y install libpcap

#yum –y install libdnet libdnet-devel

安裝snort-inline

- 安裝snort-inline：以上的前置作業都做完之後，就可以開始安裝snort-inline

#./configure --enable-inline

#make

#make install

安裝snort-inline

- 從snort 2.3.0 RC1之後，官方版本的snort原始碼，就已經包含了snort-inline的原始碼在內，不過complier之後，執行檔還是為snort，而且也是放在/usr/local/bin之下，因此complier完之後記得將執行檔改名，以免混淆。

安裝後準備

- source code中etc目錄中的snort-inline.conf拷貝到/etc/snort-inline中，然後依照前面修改snort.conf的步驟修改snort-inline.conf組態檔，注意rules放置的位置
- 把下載的rules拷貝到/etc/snort-inline/rules中等待轉換規則
- 要注意的是執行時須以-Q的模式執行，才會從iptables的QUEUE中讀取封包資料

修改RULES

- snort預設的rules都是只有告警及紀錄的功能，因此我們要將它的rules修改為snort-inline的規則模式，這時snort-inline才會把比對到的封包依snort-inline的規則將它丟棄或則是修改封包的內容
- 要考量的是那些rules需要被修改，那些不需修改

snortconfig

- 用snortconfig這個Perl程式修改我們的snort-inline rules的行動欄

#cd /usr/local/src

#wget http://www.shmoo.com/~bmc/software/snortconfig/Net-Snort-Parser-1.32.tar.gz

#tar xzvf Net-Snort-Parser-1.32.tar.gz

#cd Net-Snort-Parser-1.32

#perl Makefile.PL

snortconfig

#make

#make test

#make install

- snortconfig可以用預先定義好的組態檔來大量更改snort的規則，其語法如后：
- snortconfig -file <SNORT_CONFIG> -config <CONFIG> [-verbose] [-directory <OUTPUT_DIRECTORY>] [-honeynet] [-inline]

snortconfig

- -file <SNORT_CONFIG>：指定snort或snort-inline組態檔，snortconfig將參考snort或snort-inline組態檔內rules放置的位置執行修改動作。
- -config <CONFIG>：指定snortconfig的組態檔，依組態檔的內容修正rules。
- -directory <OUTPUT_DIRECTORY>：指定修改完後的rules要放置於那個目錄。

snortconfig

- -honeynet：因為Honeynet要預防的是內部網路，因此使用此選項時會將內部網段及外部網段顛倒過來。
- -inline：要使用這個選項，才可以在snortconfig的組態檔中使用snort-inline的特殊行動，如drop 、sdrop 、reject、replace等。

snortconfig

- Snortconfig組態檔中可針對rules的檔名、sid及classification等三種方式來修改snort 的規則，說明如后：
- files：修改符合rules檔名中之所有rules。
- sid：修改所有rules中有定義sid且符合組態檔中的sid編號的rule。
- classification：修改所有rules中有定義classtype且符合組態檔中的classification的rule。

snortconfig

- snortconfig可修改alert、disable、drop、log、replace、replace_or_drop、reject、sdrop等八種snort rules的行動，其定義與snort及snort-inline的規則行動欄定義是一樣的，因此僅將不同的說明如下：
- disable：在rule前面加上#將其關閉。
- replace：如果rule是以封包內容為比對條件的話則以隨機、相同長度的資料取代原先的封包內容。

snortconfig

- replace_or_drop：如果rule是以封包內容為比對條件的話則以隨機、相同長度的資料取代原先的封包內容，如果rule不是以內容為比對條件的話，rule的行動欄則設為drop。

snortconfig

- 再以範例做為說明：

[files]

drop: porn.rules, virus.rules

[sids]

drop: 2122, 1866

[classifications]

replace: shellcode-detect

sdrop: kickass-porn, policy-violation

snortconfig

- 將porn.rules、virus.rules規則檔內所有rule的行動欄都改為drop。
- 把所有rules中sid為2212、1866的行動欄都改為drop。
- 把所有rules中classtype為shellcode-detect的行動欄都改為replace，都以隨機相同長度的資料取代原先的封包內容。
- 把所有rules中classtype為kickass-porn及policy-violation的行動欄都改為sdrop，也就是把封包丟棄並且不做記錄。

snortconfig

- 如果不知道那些rules需要被修改，那些不需修改建議以Honeynet Project(www.honeynet.org)的snortconfig組態檔修改rules。

#snortconfig –f snort-inline.conf –config HONEYNET.config –directory /etc/snort-inline/drop_rules –inline

snortconfig

- 須注意snort-inline.conf中rule放置的位置，並記得使用-inline參數
- 修改後記得把/etc/snort-inline/drop_rules中轉換好的所有rules拷貝到snort-inline.conf中rule放置的位置中

修改後的rules

root@linux:/etc/snort-inline/drop-rules

檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
drop tcp $HOME_NET 8002 -> $EXTERNAL_NET any (msg:"ATTACK-RESPONSES oracle one
our install"; flow:established,from_server; content:"Oracle Applications One-Ho
r Install"; reference:nessus,10737; classtype:bad-unknown; sid:1464; rev:5;)
drop tcp $HOME_NET 749 -> $EXTERNAL_NET any (msg:"ATTACK-RESPONSES successful k
dmind buffer overflow attempt"; flow:established,from_server; content:"*GOBBLE*
; depth:8; reference:bugtraq,5731; reference:bugtraq,6024; reference:cve,2002-1
26; reference:cve,2002-1235; reference:url,www.kb.cert.org/vuls/id/875073; clas
type:successful-admin; sid:1900; rev:10;)
```

root@linux:/etc/snort-inline/drop-rules

檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT ssh CRC32 overflow /bin/sh";
flow:established,to_server; content:"/bin/sh"; replace:"|23 EA 86 63 4B 96 84|"; refere
nce:bugtraq,2347; reference:cve,2001-0144; reference:cve,2001-0572; classtype:shellcode
-detect; sid:1324; rev:6;)
```

IPtables設定

- 之前說有snort-inline是從iptable的queue中讀取封包，因此須先安裝ip_queue module

#insmod ip_queue

- Iptables須將要透過snort-inline檢查的網路封包送到QUEUE target

範例:

iptables －A OUTPUT －p tcp --dport 80 －j QUEUE

啟動snort-inline

- 將snort-inline以-Q的模式執行起來，否則網路流量會無法通過防火牆，這是因為IPtables把封包都丟到QUEUE中等待snort-inline處理，但是snort-inline卻沒有執行，因此所有的封包都在QUEUE中而沒有辦法出防火牆了。

#snort-inline -D -d -c
/etc/snort_inline/snort_inline.conf -Q -i eth1

習 題

習題一

- 請說明入侵偵測系統的種類有那些?

習題二

- 請說明NIDS通常放置於網路上的那些位置，各有何利弊?

習題三

- 請說明IDS及IPS的差異為何?

習題四

- 請說明何謂UTM，其優缺點為何?

習題五

- 請說明SNORT的回應動作有那些?

習題六

- Snort-inline 新增的回應動作有那些?

習題七

- 使用bridge mode的好處有那些?

習題八

- IDS的限制有那些?

習題九

- 何謂NNIDS?它跟NIDS有何不同?

習題十

- UTM有那些部署模式?

習題十一

- 試說明SNORT執行的模式有那些?

習題十二

- Snort Rules分類有那些?

Module 13-4:專案實作(**)

專案目的

- 建置Snort應用環境。
- 利用實際操作的方式讓同學了解Snort工作原理。

專案描述

- 您是一個中小企業的MIS人員，在預算有限的情況下，請架設一個免費且可自動更新Rules的入侵偵測系統，並且須可以WEB介面來分析並查看最新的攻擊資訊，以做為安全政策制定的參考。

Hint:

請參考13-33頁實作至第13-86頁

參考文獻

1. Brian Tung .CIDF.Retrieved June 21,2006, from the World Wild Web: http://gost.isi.edu/cidf/
2. Brian Caswell（n.d.）.snortconfig manual. Retrieved March 29, 2004, from the World Wide Web: http://www.shmoo.com/~bmc/software/snortconfig/man.html
3. The Snort Project. Snort Users Manual Snort Release:2.6.0. Retrieved june 31, 2006, from the World Wide Web: http://www.snort.org/docs
4. Bridge-Linux Ethernet Bridging（n.d.）. Retrieved March 28, 2004, from the World Wide Web: http://bridge.sf.net
5. MySQL Reference Manual（n.d.）Retrieved December 11, 2003, from the World Wide Web: http://dev.mysql.com/doc/
6. Patrick Harper（2003, October 6）. Snort, Apache, SSL, PHP, MySQL and BASE Install on CentOS4, RHEL 4 or Fedora Core – with NTOP. Retrieved November 23, 2006, from the World Wide Web: http://www.internetsecurityguru.com
7. Snort-inline（n.d.）. Retrieved January 2, 2004, from the World Wide Web: http://snort-inline.sourceforge.net/index.html