逢甲大學

資訊工程學系專題報告

資訊安全軟體整合系統

學生

陳柏仰(四丁)
黃俊銘(四丁)
傅聖凱(四丁)

指導教授： 李維斌 博士

中華民國九十二年十二月

目錄

Appendix

第一章 序論

1.1 研究動機

有人說現在是個電腦的時代，不過更明確的是，現在是個網路的時代，網路技術之蓬勃發展，電腦被廣泛使用，網際網路對現在的人們已經是生活上不可或缺的一環，不論是個人的資料、或是公司商業資料都可透過網路作為媒介來存取，不過也因為網路的蓬勃發展，使得犯罪行為也慢慢轉移到網際網路上，惡意入侵，破壞電腦系統，竊取機密文件...等些犯罪行為也讓使用者防不勝防，因此在網路安全上的重要性也越來越高，現今軟體中，有關安全方面軟體慢慢逐漸增加，雖然對使用者是個好消息，不過，因為選擇太廣泛，且相同性質的軟體太多，如何選擇一個你需要的安全軟體就變的困難許多，況且，沒有一個軟體可以將所有事情都做好，整合部分也越見重要，如果能夠將各種的資訊安全軟體整合，也可以將好幾套軟體整合起來，以達到互相補齊不足的部分，對使用者也是一個很好的消息，未來安全軟體的趨勢應該也是以整合為主。

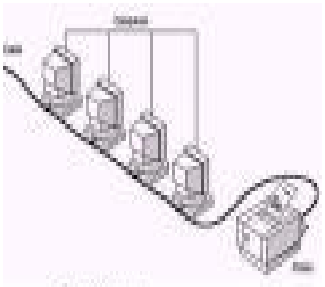
1.2 系統目的

由於Tripwire是個可以掃描硬碟是否有被異動的工具軟體,但是如果當駭客入侵時，Tripwire只能知道哪部分被修改，並沒辦法了解駭客是從哪來或者是系統哪裡出現問題或漏洞，無法偵查到駭客入侵的方法，如果沒有防範，下次駭客還是會用同一方式來進行攻擊或入侵修改，這時候只能靠別的軟體來提供其餘的功能支援，而在偵測軟體裡有一套功能非常強大的軟體 Nessus ，Nessus能做到掃描系統漏洞，進而將系統漏洞的解決方法用Report的方式提供給使用者，不過 Nessus 如果24小時都開著，也是一種資源的浪費，所以，除了能夠讓這兩個軟體能夠相輔相成以達到雙重防護的效果，所以我們才有想把這兩套軟體整合於一個介面上，使之易於使用，如果兩個軟體分開管理，會讓網安人員效率變低，如果說有一個易於管理的介面的話，可以達到讓網安人員能夠節省時間的效果，並可以將兩種軟體的優點集合起來，達到我們需要的功能，

第二章 系統架構與設計

2.1 系統架構圖

入侵者(Hacker)

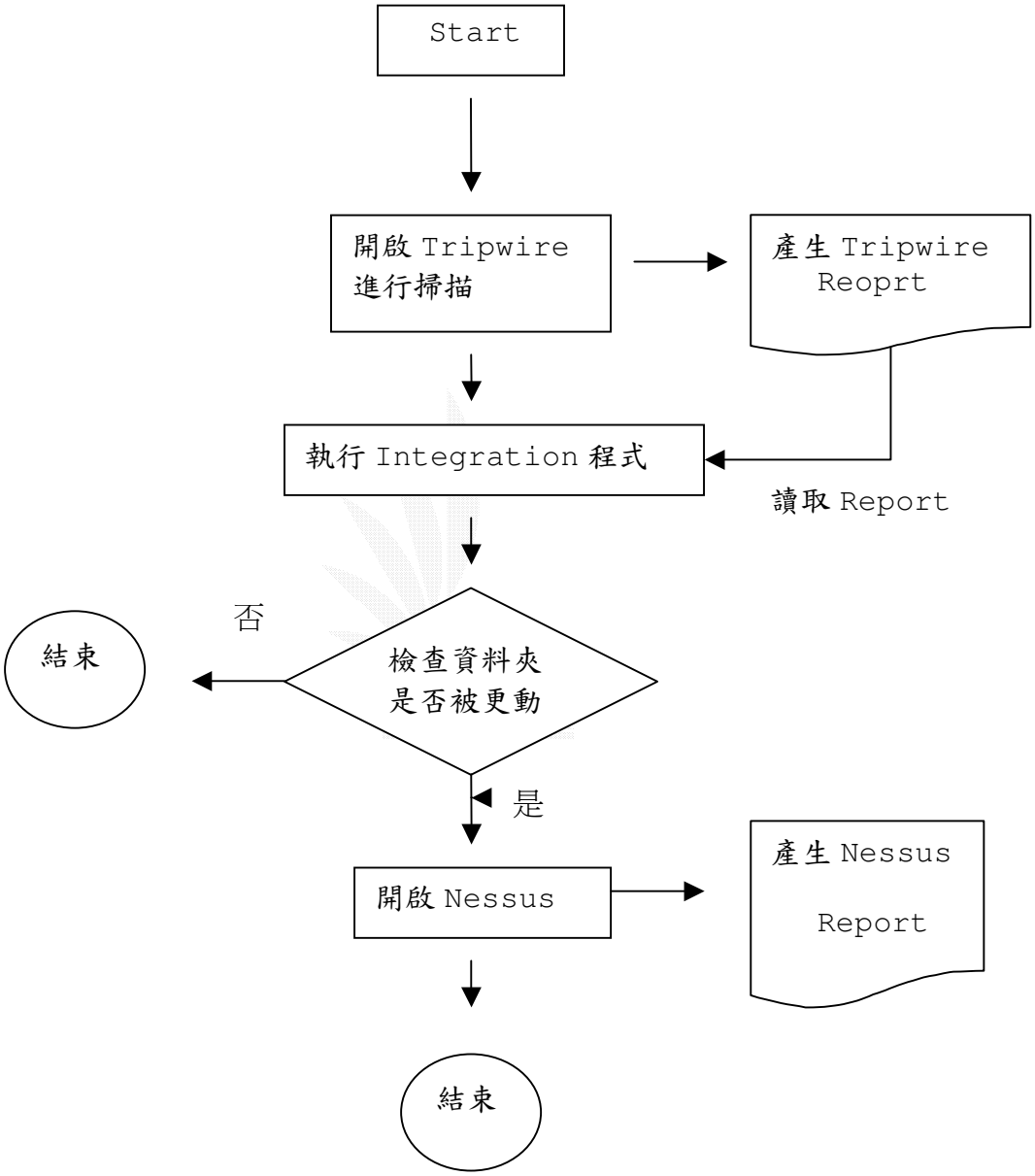


網際網路

Integration
整合防護系統

個人電腦

2.2 系統流程圖


Start
開啟 Tripwire 進行掃描
產生 Tripwire Reoprt
執行 Integration 程式
讀取 Report
檢查資料夾是否被更動
否
結束
是
開啟 Nessus
產生 Nessus Report
結束

2.3 系統架構

由於在Linux上寫程式有個好用的東西-**Shell**,可以讓有些複雜的程式簡化,也可以更好使用,所以我們這個專題也用到簡單的shell像一開始的Start

```
{
#!/bin/sh
/usr/sbin/tripwire -m c
/root/integration
}
```

使用Shell的好處就是由於Shell 本身有管線(pipe)的概念,所以能確保排程正確性tripwire產生完最新的report後,在執行integration

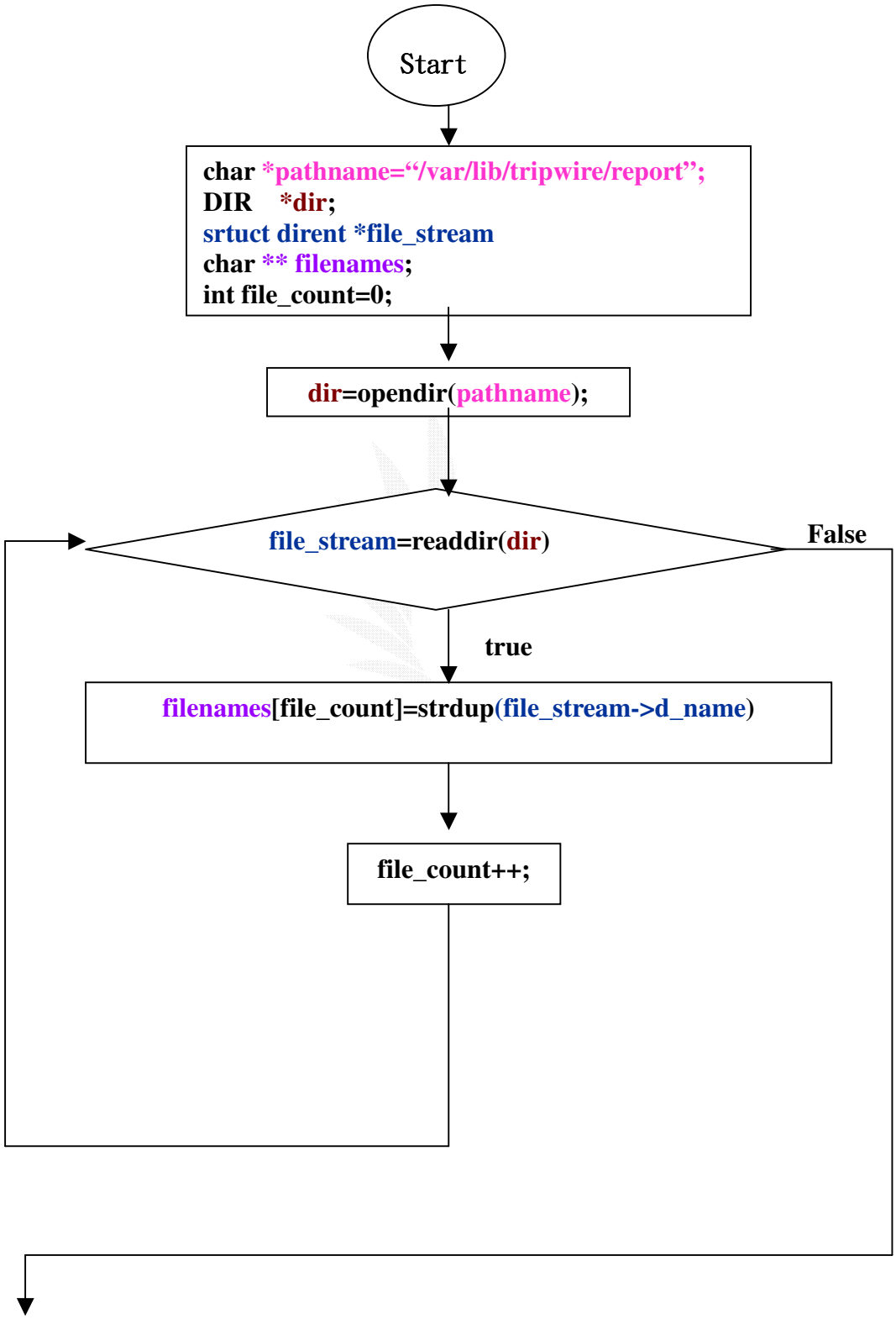
在**integration**這個小程式裡包含了**兩個部分**,一個是**讀取**Tripwire的Report部分,我們使用**OpenDir**和**ReadDir**這兩個funtion來做開啟和讀取,接下來有點麻煩的是Tripwire的檔案名稱格式是

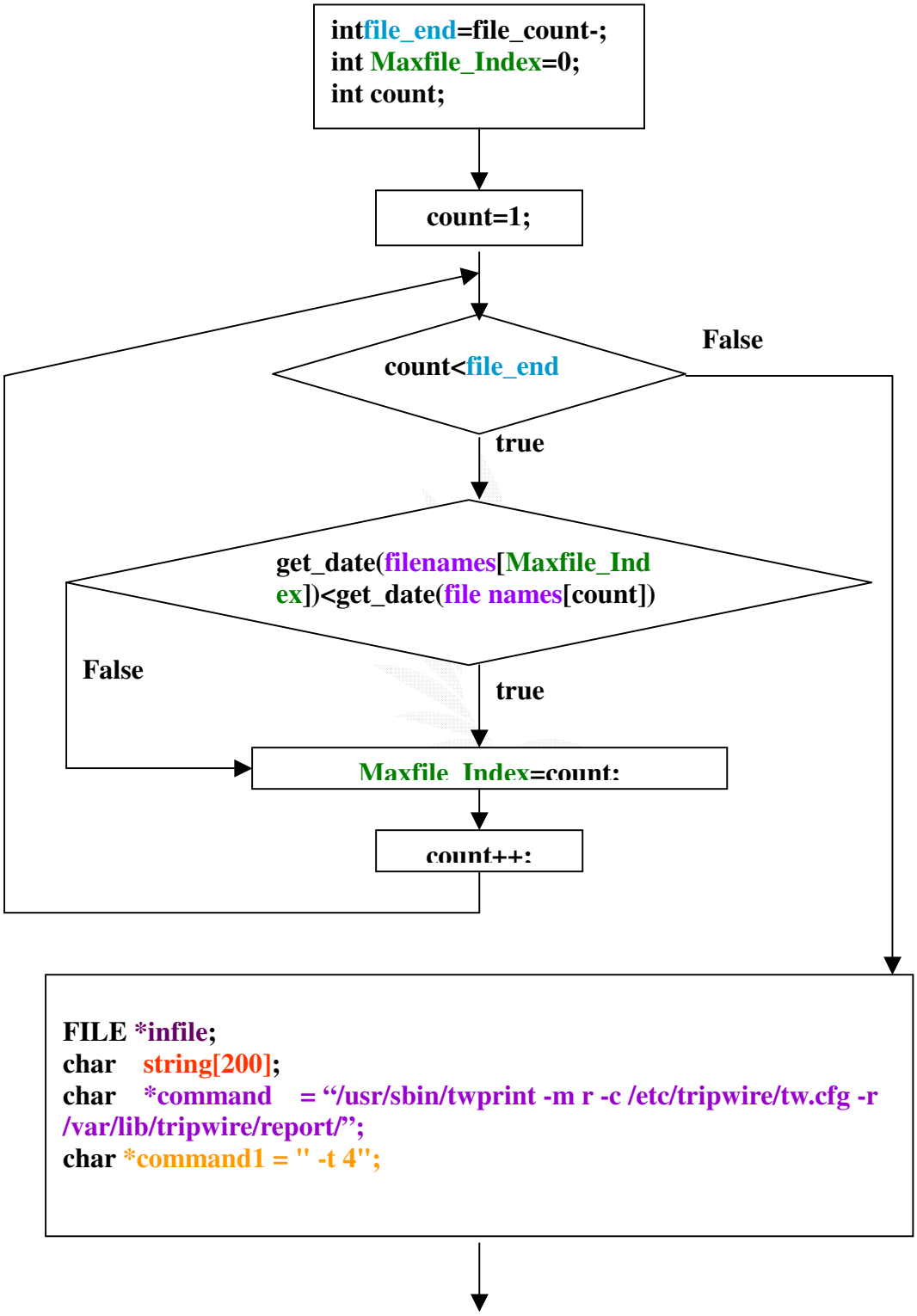
主機名稱-年月日 - 時分秒.twr

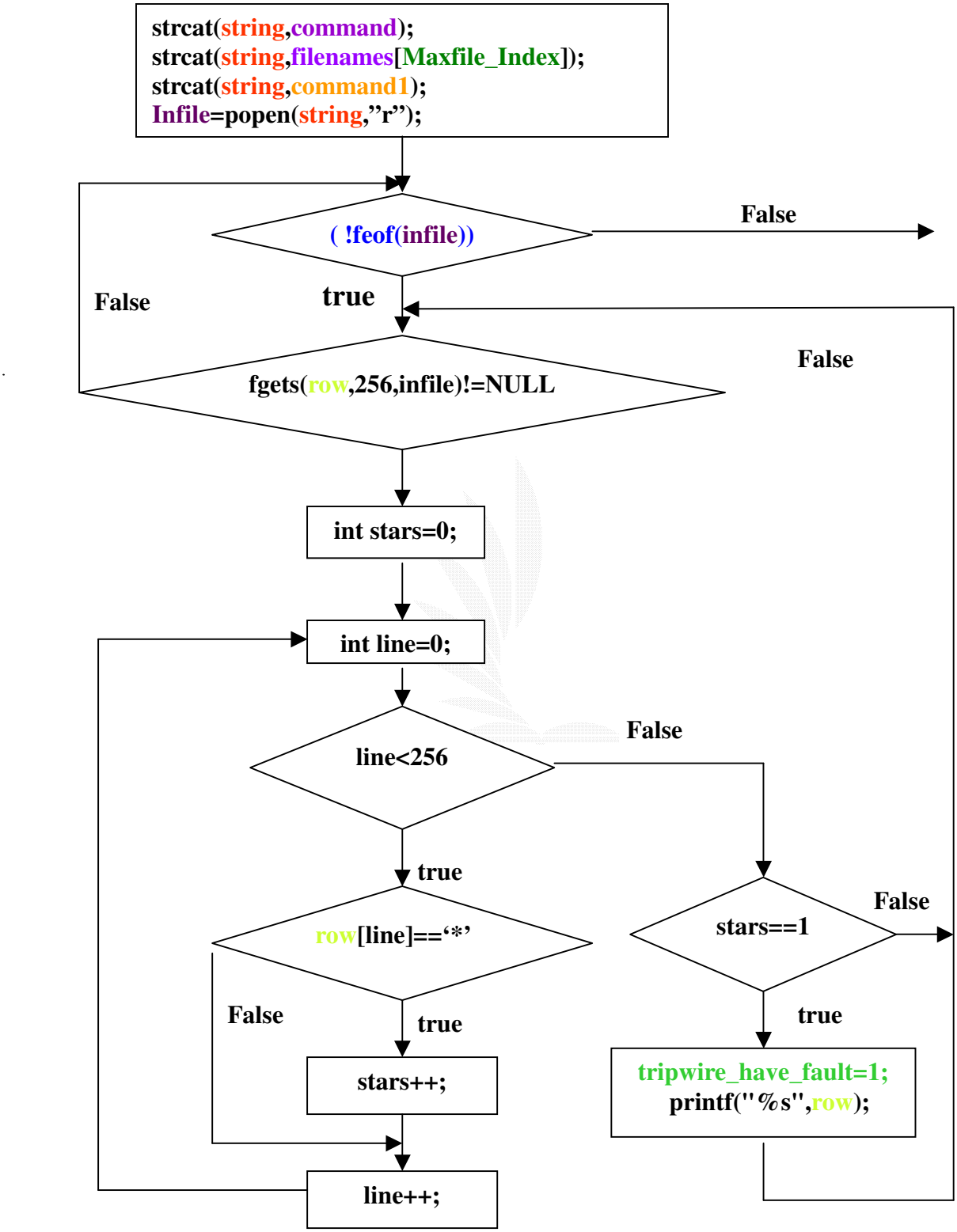
我們必須找出最新的Report來讓我們的程式作檢查的動作,我們採取的方法是將檔案名稱中的日期部分轉成數字來做比對,就可以得到最新的Report,這部分裡所需要的funtion在<String.h>裡都可以找到.

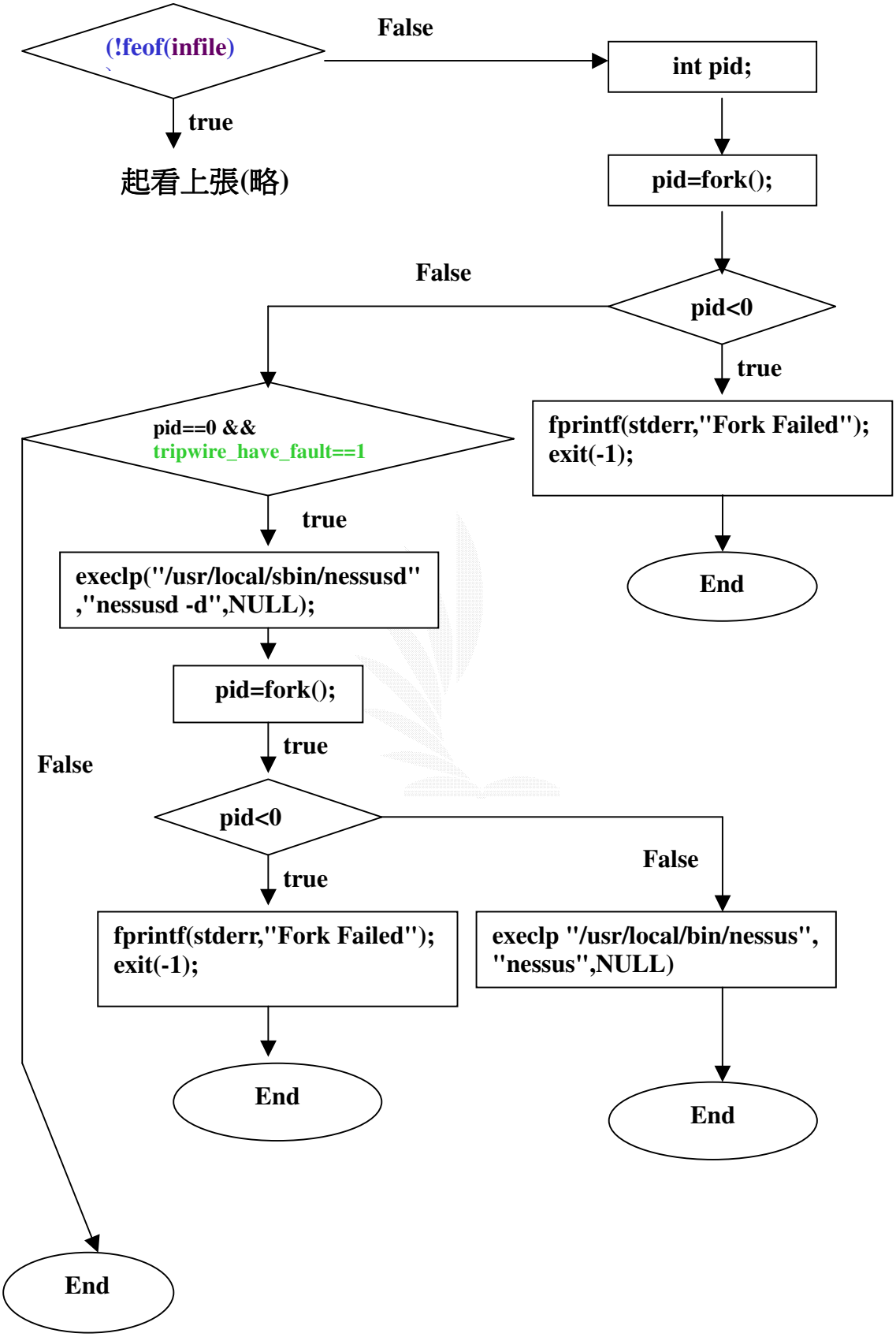
接下來程式裡的另一個部分,也就是檢查Tripwire的Report部分,我們利用popen來開啟最新的Report,因為Tripwire的Report檔格式有經過處理過,必須要用他們內建的指令才能將Report給顯示再螢幕上,所以我們只好用popen做個轉向到我們程式裡再開始一行一行掃描,當發現Report裡有資料夾被更改的紀錄後,我們就可以開啟Nessus,開始作系統漏洞的掃描,並且一方面將被更改後的資料夾作還原的動作,如果Report裡沒有被更改的紀錄會自動結束

2.4 程式流程圖


Start
char *pathname="/var/lib/tripwire/report";
DIR *dir;
srtuct dirent *file_stream
char ** filenames;
int file_count=0;
dir=opendir(pathname);
file_stream=readdir(dir)
False
true
filenames[file_count]=strdup(file_stream->d_name)
file_count++;


intfile_end=file_count-;
int Maxfile_Index=0;
int count;
count=1;
count<file_end
False
true
get_date(filenames[Maxfile_Index])<get_date(file names[count])
False
true
Maxfile_Index=count;
count++;
FILE *infile;
char string[200];
char *command = "/usr/sbin/twprint -m r -c /etc/tripwire/tw.cfg -r /var/lib/tripwire/report/";
char *command1 = " -t 4";


strcat(string,command);
strcat(string,filenames[Maxfile_Index]);
strcat(string,command1);
Infile=popen(string,"r");
(!feof(infile))
False
true
False
fgets(row,256,infile)!=NULL
False
int stars=0;
int line=0;
line<256
False
true
row[line]=='*'
stars==1
False
False
true
true
stars++;
tripwire_have_fault=1;
printf("%s",row);
line++;


(!feof(infile)
False
true
起看上張(略)
int pid;
pid=fork();
False
pid<0
true
fprintf(stderr,"Fork Failed");
exit(-1);
End
pid==0 &&
tripwire_have_fault==1
true
execlp("/usr/local/sbin/nessusd"
,"nessusd -d",NULL);
pid=fork();
true
False
pid<0
true
fprintf(stderr,"Fork Failed");
exit(-1);
End
False
execlp "/usr/local/bin/nessus",
"nessus",NULL)
End
End

第三章 安全軟體簡介

3.1 Tripwire軟體簡介

在玩tripwire 之前，我們首先必須知道，tripwire ***能做哪些事，不能做哪些事***，這是很重要的前提。因為畢竟tripwire 還是有無法杜絕防範的事情。Tripwire 為檢查檔案及目錄異動的工具，利用Tripwire可以很容易發現日常使用的系統會發生下列的事情。

Tripwire能做的事 ：

1 檔案內容被修改

2 檔案及目錄被增加

3 檔案及目錄被刪除

4 檔案及目錄的存取權限被修改

Tripwire無法做的事 ：

1.無法即時(real time)杜絕不當的存取

2.無法即時(real time)偵測出檔案的異動

3.無法得知誰修改了檔案

3.2 Tripwire實務操作細節

1. 取得軟體原始碼版本可至

   http://sourceforge.net/projects/tripwire/取得,請下載tripwire-2.3.1 以上的版本。編譯好的版本可至

   http://www.openna.com/download/tripwire/tripwire-download.htm 取得,下載tripwire-2.3.1-1.i686.rpm。PS:原本www.tripwire.org 上提供的tripwire-2.3-47 這個版本有安全上的bug。

2. 安裝Tripwire (這邊使用RPM的版本來安裝) a.用root，使用下列命令安裝rpm -ivh tripwire-2.3.1-1.i686.rpm b.在使用RPM安裝後,再執行/etc/tripwire/twinstall.sh 來完成tripwire 的安裝。執行此script時，其會詢問site keyfile passphrase與local keyfile passphrase，並用兩個key來sign policy與設定檔，避免入侵者改變tripwire的安裝設定，所以強烈建議您取個好的密碼。

Local-key 檔案(主機名稱-local.key):

此key file 是專為基線資料庫和回報檔的簽署以及認證時所使用。想要寫入受local-key保護的檔案時，需要local通行密碼(local passphrase)。

Site-key 檔案(site.key):

此key file 是專為tripwire組態檔和原則檔的簽署以及認證時所使用。想要寫入受site-key保護的檔案時，需要site通行密碼(site passphrase)。

所以，site key 被用來sign tripwire 的policy 檔，sign完後檔名通常是tw.pol；local key 被用來sign tripwire的設定檔，sign完後檔名通常是tw.cfg。再來請將tw.cfg 拷貝到/usr/sbin/ (v0.3 update) 。

PS: Tripwire的相關檔案都放置在/etc/tripwire 目錄下。

3.Tripwire的結構:

執行檔案:

/usr/sbin--------/tripwire

--------/twadmin

--------/twprint

--------/siggen

組態檔案:

/etc/tripwire-----/tw.cfg

-----/twcfg.txt

-----/tw.pol

-----/twpol.txt

-----/主機名稱-local.key

-----/site.key

其他檔案:

/var-/lib-/tripwire-------/主機名稱.twd

-------/report

在trupwire底下，有幾種常用語是經常會看到的，在此稍做說明:

mode: Tripwire總共有4個執行檔，其命令依據用途的不同而有所分別。在Tripwire中，因應各種作業模式所指定的命令，稱為mode。各種命令因作業模式不同，其參數值也會改變。因此，對於命令的模式有必要加以

正確的理解。

policy: tripwire為檢查檔案及目錄變動與否的工具，可以檢查的項目(property)共有18種。而撰寫著需要檢查的檔案及目錄內容的檔案 ，則稱為**原則檔**(policy file)。Tripwire會根據這個原則黨所撰寫的內容，對檔案及目錄進行檢查。

Report: 當以tripwire對檔案及目錄進行檢查時，將比對是否違反原則檔中的內容，並將其結果儲存在一個稱為**回報檔**(report file)的檔案裡。每當檢查出檔案及目錄產生異動時，其詳細內容都會紀錄在回報檔裡。

Baseline database: 這是tripwire的核心檔案。Tripwire會將所欲檢查的對象檔案及目錄資訊，儲存在這個資料庫裡(注意:並不是把資料本身褚存在這個資料庫裡)。當執行完整性檢查時，就會拿儲存在**基線資料庫**(Baseline database)裡的內容，和目前的內容做比對。因此，這個被作為比較對象的檔案擁有非常重要的功能。

4.簡介組態檔的位置及功能:

tripwire組態檔的位置在/etc/tripwire目錄底下。下面分別介紹在這個目錄裡，各檔案的功能:

tw.cfg : tw.cfg是tripwire的組態檔。Tripwire會根據這個檔案的設定內容來執行。這是一個二進位檔案(binary file)，它以site key來進行加密簽署。

twcfg.txt: 這是明文(clear text)形成的組態檔。

tw.pol : tw.pol是tripwire的原則檔。Tripwire會根據這個檔案的設定內容，進行檔案及目錄的檢查。這是一個二進位檔,它以site key來進行加密簽署。

twpol.txt: 這是明文形式的原則檔。Tripwire所提供的預設原則檔，為RedHat 7.0 所使用的版本。安裝Tripwire後，會以這個檔案為基底建立一個tw.pol檔。

5.執行檔的位置及功能:

Tripwire總共有4個執行檔，他們的預設位置在/usr/sbin目錄底下。下面分別介紹個檔案的功能:

*tripwire：*這是完整性檢查的執行工具。對基線資料庫的初始化或更新，以及原則檔的更新等，都用這個工具程式來執行。

twadmin： 組態檔即原則檔的管理程式。Key file的管理和加密簽署，也都用此工具進行。

twprint： 回報檔及基線資料庫內容的顯示程式。主要用來閱讀Tripwire所建立的回報檔。

siggen： 計算檔案的雜湊值工具。可以計算指定檔案的CRC32、MD5、SHA、HAVAL。

6. 簡述Tripwire作業流程:

a.建立原則檔(policy file)

b.建立基線資料庫(baseline database)

c.完整性檢查

d.確認Tripwire回報檔(report file)

e.更新資料庫

f.需要時更新原則檔

g.檢查

7.現在我們開始實際操作。首先自定policy file(原則檔)。

由於tripwire 安裝時是使用預設的policy ，其中可能會因個人Linux系統的不同，而不能完全適用。所以需要自行修改policy 來滿足個人的需求。tripwire 已經在 /etc/tripwire/twpol.txt 內有存放一個純文字的policy 檔，所以依這個檔案來修改。記得修改HOSTNAME=xxxx; <-- your hostname 另外，檔案內的$(SIG_HI) 或$(SEC_BIN) 為預設的自定property mask ，在此檔開始前面@@section FS就可找到。例如我們有安裝ssh ，則另外又因為/usr/sbin 在前面已經有設定了，所以不用加入/usr/sbin/sshd 的 rule。

範例： 建立簡單的原則檔 sampol.txt

```
(
   rulename = “Test Policy”      #規則區塊名稱
)
{
   /tmp/test → $(ReadOnly);      #test檔的檢查規則
}
```

解釋: 在Tripwire裡，要如何檢查檔案等的目的檔(object，檢查對象)是以項目遮罩(property mask)來指定。將目的檔與項目遮罩組合起來的撰寫方式如下：

/tmp/test → $(ReadOnly);

（目的檔） →（項目遮罩）

讓我們來看看幾個項目遮罩範例：

1.只檢查檔案的存取權限和檔案模式位元

/tmp/test -> +p;

2.檢查檔案的存取權限和檔案模式位元，擁有者的使用者ID，群組ID，MD5的雜湊值。

/tmp/test -> +pugM;

3.幾乎檢查所有的項目遮罩(並非成長型檔案)

/tmp/test -> +pinugtsdrbamcCMSH;

4.只檢查檔案是否存在

/tmp/test -> -pinugtsdrlbamcCMSH;

8.原則檔的簽署：

接下來我們的作業將全以root的身分來進行。首先，以自己

編輯好的原則檔(***sampol.txt***)複製到/etc/tripwire目錄底下。然而，這個原則檔是無法就此使用的。Tripwire無法使用明文(clear file)的原則檔，唯有加密簽署過的檔案才能使用。這個檔案預設位置在/etc/ tripwire/tw.pol。

要對明文支原則檔進行加密簽署，可利用”建立原則檔模式”的/usr/sbin/twadmin 命令。底下是簽署原則檔的命令列語法:

twadmin -m P -c tw.cfg -p tw.pol -S site.key sampol.txt

其中，-m P →指定模式

tw.cfg →組態檔

tw.pol →原則檔

site.key →site.key

sampol.txt →明文的原則檔

現在就讓我們實際將***sampol.txt*** 建立成經過加密簽署的原則檔:

/usr/sbin/twadmin -m P -c /etc/tripwire/tw.cfg -p

/etc/tripwire/tw.pol -S /etc/tripwire/site.key /etc/tripwire/sampol.txt

螢幕出現:

Please enter your site passphrase: ← 鍵入site通行密碼

Wrote policy file: /etc/tripwire/tw.pol ← 顯示成功的建立成tw.pol

如此，原則檔的建立才算大功告成。

9. 而當Tripwire裡最重要的原則檔準備好了，我們就可以街著往下著手另一項重點—**建立基線資料庫**(baseline database)。指示Tripwire建立基線資料庫後，Tripwire會根據剛才建立的tw.pol內容，將/tmp/test檔案的資訊收納到基線資料庫裡。

   但是，在這之前，要先建立好檢查對象(例如妳可以將妳資料庫裡最機密的檔案夾設為檢查對象)，也就是/tmp/test檔案。而這裡舉一個簡單的檢查對象:

   在/tmp路徑下新增一個檔名為test的文件資料，

   [root@linux / tripwire]# **echo This is Tripwire test >**

/tmp/test

[root@linux / tripwire]# cat /tmp/test

於是乎，建立一個只包含” This is Tripwire test”訊息的檔案。而這時候我們要進行建立基線資料庫的動作就必須使用”資料庫初始化(init)模式”的/usr/sbin/ tripwire 命令。

10. 資料庫初始化(init)模式與法如下:

初始tripwire 資料庫/usr/sbin/twadmin -m P /etc/tripwire/twpol.txt /usr/sbin/tripwire -m i

現在讓我們實際操作進行基線資料庫的初始化:

[root@linux tripwire]# usr/sbin/tripwire -m i

螢幕出現:

Please enter your local passphrase: <- 鍵入local 密碼

Parsing policy file : /etc/tripwire/tw.pol <- 要利用的原則檔

Generating the database.....

***** Processing Unix File System *****

Wrote database file : /var/lib/tripwire/linux.twd <-

建立資料庫

The database was successfully generated. <- 完成

因為這裡我們tripwire原則檔設定欲檢查的檔案只有一個而已(檔名:test)，所以基線資料庫應該很快就能建立完成。基線資料庫預設將以”/var/lib/tripwire/主機名稱.twd”的名稱來建立。

11. 第一次檢查:產生回報檔(report file)

tripwire命令的完整性檢查模式，有好幾種。這裡我們呈現的是最簡單的命令列:

tripwire -m c

其中，**-m** 指定模式

c 指定檢查

現在，讓我們來實際操作一次完整性檢查 。

[root@linux tripwire]# **/usr/sbin/tripwire -m c**

螢幕出現:

Parsing policy file: /etc/tripwire/tw.pol

**** Processing Unix File System ****

Performing integrity check……

Wrote report file :

/var/lib/tripwire/report/linux-(Hostname)-(Date).twr

接下來出現的就是回報檔(report file)了。

12. 回報檔的讀取方式:

預設的回報檔，根據組態檔的指定是建立在

/var/lib/tripwire/report/ 的目錄下。

組態檔(tw.cfg)內，預設回報檔的設定如下:

REPORTFILE = /var/lib/tripwire/report/$ (Hostname) -

$(Date).twr

預設的回報檔名，會按照如下的規則來命名:

主機名稱 - 年月日 - 時分秒.twr

而如果我們要印出回報檔來看的話，可使用twprint命令。

其命令語法如下:

twprint -m r -c tw.cfg -r

$(Hostname)-$(Date).twr -L local.key -t 0-4

其中，$(Hostname)-$(Date).twr →回報檔

-t 0-4 →指定報告詳細層級

現在我們來實際操作使用指定層級為0的報告來看看:

[root@linux tripwire]# /usr/sbin/tripwire -m r -c /etc/tripwire/tw.cfg -r/var/lib/tripwire/report/linux-(Hostname)-(Date).twr -t 0

於是，螢幕出現(層級0的違反狀態報告):

Note : Report is not encrypted

TWReport linux 20031118180325 V:0 S:0 A:0 R:0 C:0

其中，V:0 →總共違反規則數

S:0 →嚴謹層級

A:0 →追加的檔案

R:0 →刪除的檔案

C:0 →被修改的檔案

這次顯示的是與完整性檢查，完全不同的結果。這個報告詳

細層級為0的報告，是tripwire最簡單的報告，主要被用來輸出到Syslog。但是，從這個簡單的報告也可以讀取到各種資訊。所以，當完整性檢查過後，只要使用這個命令，就可以直接清楚的了解目前檔案是否有違反規則了。而不必一開始就讀取最龐大雜亂的回報檔(report file)。

13. 精簡policy file 使用下列命列:

    找出policy file 中有定義，但這台機器卻沒有此檔案或目錄的rule ，將它從policy file 中去除。

    /usr/sbin/tripwire -m c | grep Filename >> twnotfound.txt 當編輯完policy file 後，需要重新安裝policy，可使用下列命令重建Database：

    /usr/sbin/twadmin -m P /etc/tripwire/twpol.txt

    /usr/sbin/tripwire -m i

14. 移除純文字的policy 與設定檔由於tripwire 已經利用key將policy 與設定檔加密成tw.pol,tw.cfg了，所以原始的policy與設定檔便可移除了。

    rm /etc/tripwire/twpol.txt

rm /etc/tripwire/twcfg.txt

15.排程檢查:

排程執行tripwire 分析(每小時一次) 在 /etc/cron.hourly/ 下新增一個可執行的script，內容如下： #!/bin/sh /usr/sbin/tripwire -m c | mail -s "Tripwire Report from {some_host}" root@localhost

管理者每天以手動的方式進行Tripwire的完整性檢查，事非常不方便的事情。特別是進行整個系統的檢查時，必須花費時間、CPU、硬碟等負擔重的作業，應該在深夜或系統使用率最少的時間才來進行。Tripwire可透過cron進行排程檢查。

例如，每天深夜兩點，進行整個系統的完整性檢查，將其結果以電子郵件傳送，語法如下:

02 **** /usr/sbin/tripwire -check - email -report

如果想要每隔5分鐘使用Web規則名稱，檢查/var/www目錄，並且將完整性檢查的結果，以報告層級0(只有一行)使用電子郵件傳送，語法如下:

*** /5 **** /usr/sbin/tripwire -m c web -M -t 0**

16.更新設定檔取出現有的設定內容匯出成純文字檔 twadmin -m f > twcfg.txt修改twcfg.txt ，再更新回去twadmin -m F --site-keyfile /etc/tripwire/site.key twcfg.txt 刪除twcfg.txt

17.資料庫的更新:

若系統有安裝或修改檔案，需要更新tripwire 的資料庫，不然會一直收到警告的信 。使用下面命令：

/usr/sbin/tripwire -m u -r /var/lib/tripwire/report/{最後時間}.twr 之後會進入 vi 編輯報表，Tripwire的報表會在每一個違反策略檔案中所定義的規則的地方加上一個選擇框。可保留選擇框中的"x"，表示接受這個變化。如果把選擇框中的"x"移掉，表示資料庫不會更新這個變化。等結束編輯器並輸入本地的passphrase(密碼)之後，Tripwire就會更新並存檔資料庫。

18.更新policy 檔:

先將現有的policy 內容匯出成純文字檔twpol.txt，

/usr/sbin/twadmin -m p > /etc/tripwire/twpol.txt 再

修改/etc/tripwire/twpol.txt，修改完再更新回去，/usr/sbin/tripwire -m p /etc/tripwire/twpol.txt 更新回去後，再執行一次完整性檢查，看看policy 是不是想要的。/usr/sbin/tripwire -m c 記得還要刪除 /etc/tripwire/twpol.txt PS:當然也可以使用twadmin -m p /etc/tripwire/twpol.txt 來更新策略檔，但更新完後還需要重新初始化tripwire資料庫。所以不建議使用 。

19. 附錄: Tripwire組態檔

Tripwire的組態檔tw.cfg，用來保存Tripwire所使用的檔案位置，以及運作模式。由於tw.cfg是經加密簽署的檔案，無法像一般的文字檔一樣印出來。以下所列出的是安裝時所預先定義的tw.cfg組態檔。(其明文範本檔為twcfg.txt，存放在/etc/tripwire目錄裡)

```
Root        = /usr/sbin
POLFILE     = /etc/tripwire/tw.pol
DBFILE      =/var/lib/tripwire/$(Hostname).twd
REPORTFILE
            =/var/lib/tripwire/report/$(Hostname)-$(D
```

ate).twd

SITEKEYFILE =/etc/tripwire/site.key

LOCALKEYFILE = /etc/tripwire/$(Hostname)-local.key

EDITOR = /bin/vi

LATEPROMPTING = false

LOOSEDIRECTORYCHECKING = false

MAILNOVIOLATIONS = true

EMAILREEPORTLEVEL = 3

REPORTLEVEL = 3

MAILMETHOD = SENDMAIL

SYSLOGREPORTING = false

MAILPROGRAM = /usr/sbin/sendmail -oi -t

在組態檔裡所撰寫的變數名稱，必須使用大寫字母。而定義變數值，則不分字母大、小寫。

下面有兩個變數是安裝時事先定義的，無法改變其值。

HOSTNAME:

Tripwire所執行的電腦主機名稱

DATE:

日期與時間，以文字的方式構成

而在Tripwire2.3.1-2之後的版本，組態檔裡新增了兩個可定義的變數。

TEMPDIRECTORY:

由tripwire預設所產生的暫存檔(temp file)目錄，可以指定任意的位置。雖然可以直接利用預設所產生的/tmp，但是，因為任何人都可以將檔案寫入/tmp裡，這這個位置可是一點都不安全。使用TEMPDIRECTORY變數，可定義成只能由root寫入的目錄。如此，更可以提高其安全性。

GLOBALEMAIL:

使用原則檔的emailto屬性時，只要在組態檔內定義GLOBALEMAIL的電子郵件地址。

必備變數：

組態檔可以定義各種變數，但是，下列五種變數為必須定義的變數：

POLFILE：預設的原則檔

DBFILE：預設的資料庫檔

REPORTFILE：指定完整性檢查結果，所產生的回報檔名稱

SITEKEYFILE：指定預設的site-key

LOCALKEYFILE：指定預設的local-key

選項變數：

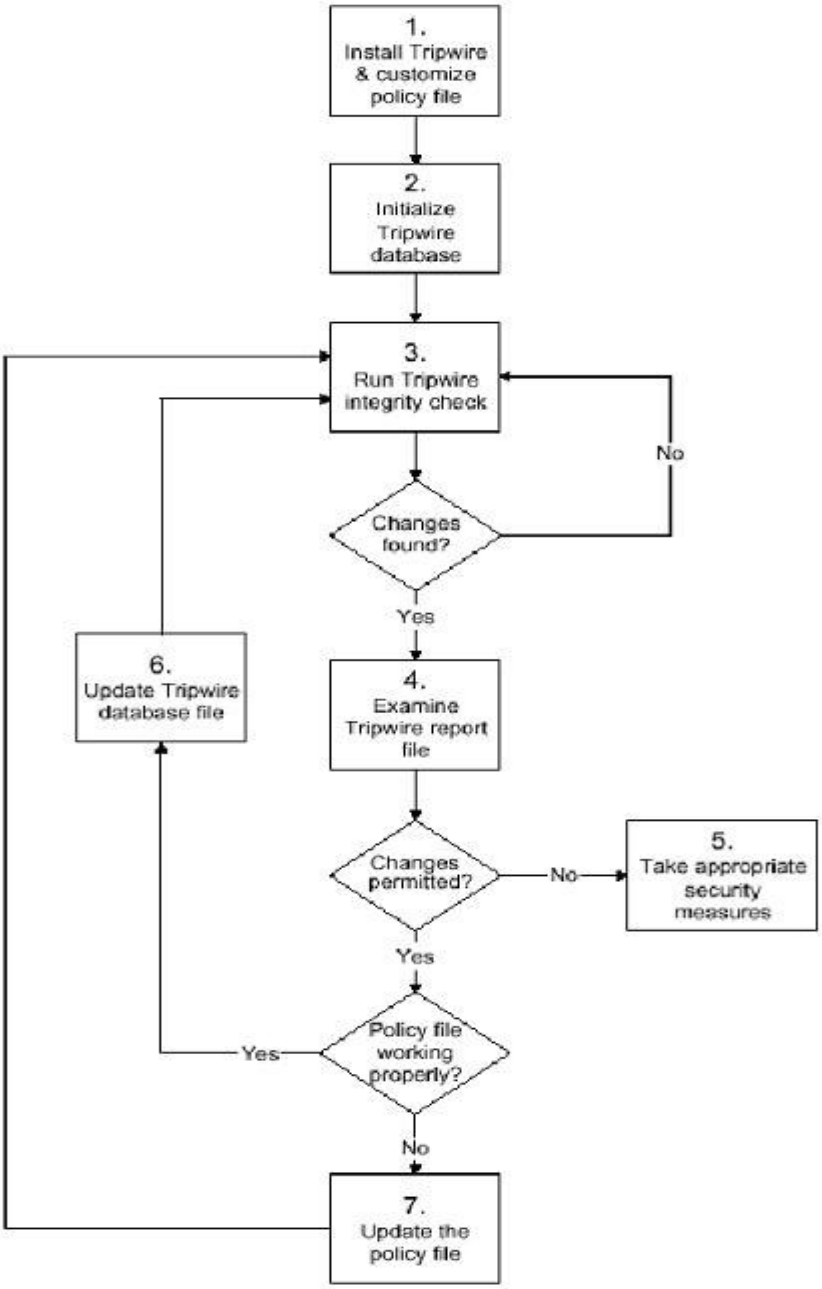
EDITOR：指定互動模式所使用的編輯器絕對路徑。預設指定為Vi編輯器

LATEPROMPTING：設定為true時，因為將保存在記憶體內的密碼時間縮短，會將通行密碼的提示盡量延遲。

SYSLOGREPORTING：設定為true時，會將資料庫的初始化、完整性檢查等訊息，以user.notice層級輸出到syslog。

REPORTLEVEL：這是當使用twprint命令顯示回報檔內容時，指定回報預設的詳細層級的變數。可以指定的回報詳細層級值為０~４。

3.3 Tripwire設定流程圖



3.4 Nessus 軟體簡介

Nessus 是一免費的網路安全檢測工具,它是在 1998 年被法國的 Renaud Deraison 發展出來

主要是因為當時免費的檢測工具如 SATAN 以漸漸不能檢測當時網路環境上各式各樣新興的服務 ，而商業的版本又太為昂貴，所以作者決定提供一個免費而又能快速更新的網路安全檢測工具，Nessus 能針對指定的網路進行安全檢查，找出該網路是否有導致 Cracker 進行攻擊的漏洞

Nessus 分為 Server 端和 Client 端為一分散式的架構,主要是在 Server 端新增使用者後即可以 Client 端登入，再藉由一些規則設定來決定掃描地區

因為 Nessus 的系統掃描是屬於攻擊程式，所以隨意讓每一個人使用是非常危險的，不過經由適當的設定，可以避免不當的使用

3.5 Nessus 主要特點

1、Plug-in 的組成方式：Nessus 的各種檢測程式都寫成外掛的模組，使用者能輕易的增加檢測程式至Nessus 程式內而不需去閱讀核心引擎的程式碼 。

2、主從式的架構：Nessus 是由二個部份所組成，server部份主要是負責作檢測(攻擊)的執行，它只能安裝在Unix-like作業系統下，而Client 的部份主要是一前端介面，能提供使用者去登入nessus 的server、選擇所要執行的檢測並且顯示檢測的結果，而Client 主要有三種選擇：win32、Unix-like 和Java。

3、能同時檢測無限制的主機：假如你執行檢測的Server夠強，Nessus 並不會限制同時檢測多少數量的主機 。

4、聰明的辨識主機上的服務 ：Nessus 並不認定特定的Port 必定是某項服務，所以它能檢測改變固定Port 的服務(如web 的服務並不是在Port 80)。

5、重覆服務的檢測：假如一台主機上有二個以上相同的

服務(如二個ftp 服務)，Nessus 都會對其執行檢測。

6、NASL 語言：在Nessus 下檢測程式的撰寫，可以用C 語言或是Nessus所發展的受測語言NASL(Nnessus Attack Scripting Language)撰寫，而用NASL 發展測試程式的優點是執行效率較C 語言佳，而且將來如果在Windows 下發展server 端，則用NASL 所撰寫的檢測程式會較容易移殖。

7、支援多種格式的報告結果：假如Client 端是在POSIX 的系統下執行，它能輸出的報告格式有.txt、.html、LaTex 等格式。

3.6 選擇Nessus的原因

- 安全性高
- Client-Server架構
- 易於取得,免費軟體
- Client端支援 linux/Windos系統
- 易於更新和修改
- 可依自我需求做變更

3.7 Client 與Server 間的安全性

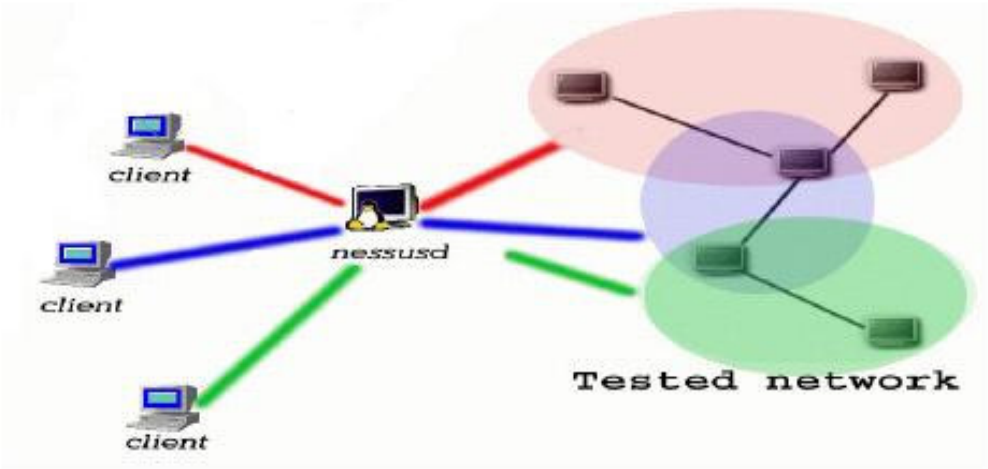
分散式架構為Nessus 的優點，但也因此在執行檢測的工作時Client 端須登入到Server 端才能執行檢測的工作，而中間通訊的過程中，如何確保其安全性就非常重要，而Neussus 利用多種加密和訊息摘要演算法來確保其溝通安全性，其支援之加解密演算法如下列：

1、支援CBC 模式之{blow/two}fish，blowfish160，3des block ciphers andRIPEMD160, SHA, MD5 mac (CRC)

2、128 位元之金鑰長度

3、支援壓縮格式

4、使用任意長度之El Gamal 非對稱式金鑰



3.8 Client 與Server 之調整設定

當安裝完成後，基本上不用任何調整已可執行但如要調整其功能，也可藉由下列所介紹的命令與參數做調整：

1、Server-Nessusd 之詳細參數與組態檔調校介紹：

（1）指定第一次使用時，使用者之密碼：

nessusd —make-user=username,passwd 或者也可使用

nessusd -P username,passwd

注意：使用者帳號與密碼間不可有空格，須用逗號隔開

（2）查看金鑰資料庫：

nessusd --list-keys 或使用

nessusd -L

（3）nessusd 命令下詳細參數功用：

-D,--background

二者都是令server 以daemon 形式執行

-c<config-file>,--config-file=<config-file>

使用替代性組態檔代替/nessus/etc/nessusd.conf 此

組態檔

-a<address>, --listen=<address>

讓Server 只監聽某一IP 位址的要求，例如nessusd -a 192.168.1.1則此時Server 只會監聽從此位址傳過來的要求，這參數適用在當nessusd 是執行在gateway 下，或你不想讓外部使用者使用nessusd。

-p<port>, --port=<port>

指定埠號已代替預設的port 3001

-v, --verison

寫入版本編號並離開

-h, --help

顯示全部之命令

-d, --dump-cfg

讓server 放棄其組態

（4）金鑰管理參數

此類參數只能使用在nessusd 已被root 所啟動成daemon 形式。

-C, --change-pass-phrase

藉由個人的密碼，讓nessusd 保護private key，在每次啟動server 時，皆須填入密碼, 才能啟動nessusd 假如密碼忘記時， 可以在 /nessus/etc/nessus/nessusd.private-keys 此檔案下刪除密碼後再不填入密碼。

-L, --list-keys

列出存放在使用者金鑰資料庫的值 。

-K *<key>*, --delete-key=*<key>*

從使用者金鑰資料庫中刪除使用者的key，<key>參數內可以是主機位址或使用者帳號。

-U <user-name>, --list-user-pwd=<user-name>

列出儲存在資料庫內，特定使用者的資訊，這是在使用者多次登入失敗時，提供管理者查詢其密碼之用

-P *<user-pwd-mod>*, --make-user=*<user-pwd-mod>*

增加、刪除或修改使用者帳號或密碼，使用者之密碼只用在最初client 與server 通訊時使用，其後client 端就不需再填入密碼。

（5）組態檔之調校：

預設nessusd 之組態檔是存放在

/nessus/etc/nessusd.conf 中，其組成方式為

<keyword>=<value>，而#後之文字表示註解之用，下列介紹一些關鍵值：

plugins_folder：其plugins 所在位置，預設為/nessus/lib/nessus/plugins，但你也能改變它。

Logfile：記錄檔之路徑，你能填入syslog 假如你想讓訊息存入

syslogd，你也能填入stderr 讓記錄放在標準輸出。

Max_threads：同一時間可以測試主機個數的最大值，必須考慮到你網路頻寬與server 所在機器之性能。

Users：使用者資料庫所在位置。

Rules：規則資料庫所在位置。

Language：輸出檢測報告時所用之語言，目前可用英語與法語。

Check_read_timeout：執行檢測時等待結果傳回時的時限，假如網路擁塞你應增加此值。

Peks_username：這為了讓nessusd 在private key 資料

庫中，識別它們自己。

Peks_keylen：公開金鑰的最小長度。

Peks_keyfile：private key 資料庫的存放路徑。

Peks_usrkeys：使用者的public key 與密碼資料庫的存放位置。

Peks_pwdfail：允許登入失敗的最大次數。

（6）使用者資料庫

其功用為限制特定使用者使用此server，其組成方式為

user:password [rules]

以下就是一個簡單的範例：

帳號eason, 密碼為0000

eason:0000

deny 192.168.1.1/32 #不允許eason 去檢測

192.168.1.1 Class C 間的所有主機

accept 192.168.0.0/16 #允許此使用者去檢測

192.168.0.0 Class B 間的主機

default deny #預設值是拒絕

附註：在nessus 中可設定規則的地方共有三處rules database（對整

個server 設規則),users database(對server 上某一使用者設規則)和user rules(由client 端自行設定),其設定的效力高低就是上列的順序rules DB>users DB>user rules 如果最上層的限制不能檢測某主機則下層的不能自訂可檢測此一主機。

3.9 Client-nesssus 之文字介面指令與參數介紹

(1)、文字介面指令

如在無x window 環境下執行nessus 則須以指令模式執行相關功能,在nessus 命令後加上下列參數:

Server:欲連接的server 主機名稱或是其IP 位址。

Port:server 所監聽需求所在之埠號。

Login:使用者帳號名稱。

Trgfile:內含欲執行檢測主機位址或名稱之檔案。

Resultsfile:執行完檢測後,檢測結果欲儲存之檔案。

(2)、參數介紹:

要執行nessus 的參數,須在其前加人"-"符號,以與文字介面命令做區隔,其參數有:

-v, --version：顯示版本編號，並結束執行。

-h, --help：顯示所有可用之參數。

-n, --no-pixmaps：無圖片形式，可減少網路頻寬之負載。

-k<key-len>, --set-key-length=<key-len>：指定key 的長度，而不是預設的1024

-T<type>, --output-type=<type>：指定檢測結果儲存之格式，<<type>可以是’html’,’text’,’tex’或’nsr’

-q, --batch-mode：強迫結束，或進入批次指令模式。

-C, --change-pass-phrase：執行nessus 時需使用此密碼，才能使用。

-L, --list-keys：顯示在使用者金鑰資料庫的欄位。

-K<delete-key>, --delete-key=<delete-key>：從使用者金鑰資料庫刪除某一金鑰。

第四章 網路安全簡介

4.1 網路安全的重要性

不管是腳踏車、摩托車、還是汽車，大部分的人買了之後，幾乎都還會再加買一把鎖。大部分的人寫信都會用信封，信寫完了，將信封口封住後，才會丟到郵筒裡。但是今天我們用的的電腦，不論軟、硬體，都沒有鎖，都缺乏安全性。

電子郵件，除了需要密碼以便從主機取信、寄信外，我們的電子郵件幾乎和寄明信片一樣透明，只要是有心人，都可輕而易舉地看到信件的內容。我們的電腦也沒有「鎖」，大部分在公司上班的人都讓電腦開著就離開，這些電腦和電腦裡的資料就這樣擺在那兒，等這有心人士來拿。

事實上，不論在哪兒（政府單位、銀行、公司行號、研究單位、家裡），電腦裡的資料都是最珍貴的，最值得保險的。

整個電腦界或是網路社會對於電腦資料之安全性都沒有辦法提出解決方案，大部分的人或公司都認為電腦和網路還不是一

個很安全的地方，對於網路也還沒到完全信賴的程度。但是電腦和網路增加了資訊的傳輸，許多交易也漸漸地利用電腦及網路來完成，因而增加了陌生人破壞電腦及網路的機會。以前只有銀行用電腦，現在幾乎所有的公司、醫院、政府單位、百貨公司、倉儲、工廠、所有的金融體系都用電腦了，甚至連電梯、電話也都用電腦來控制。電腦儲存了太多個人資料，太多工作程序，當電腦和傳輸資料的網路的安全性不足時，該怎麼辦呢？而所謂的安全性，來自各種威脅，天災人禍、停電、系統本身的錯誤...。但是這些安全顧慮並不是個人或是幾家公司可解決的問題，有些可以靠政府立法維護，有些則是社會責任與道德。

4.2 影響網路安全的因素

網路實體為電話線、電纜、或是光纖的組合，「網路」本身沒有什麼危險性。網路安不安全的問題在於：用網路傳遞資料是否很容易遺失、毀損、或是被人截取。用網路傳輸資料到彼端之後，彼端電腦如何保管這些資料，這些安全性題，基本上牽涉到的因素很多：軟、硬體的技術問題，以及使用者的觀念和使用方式。

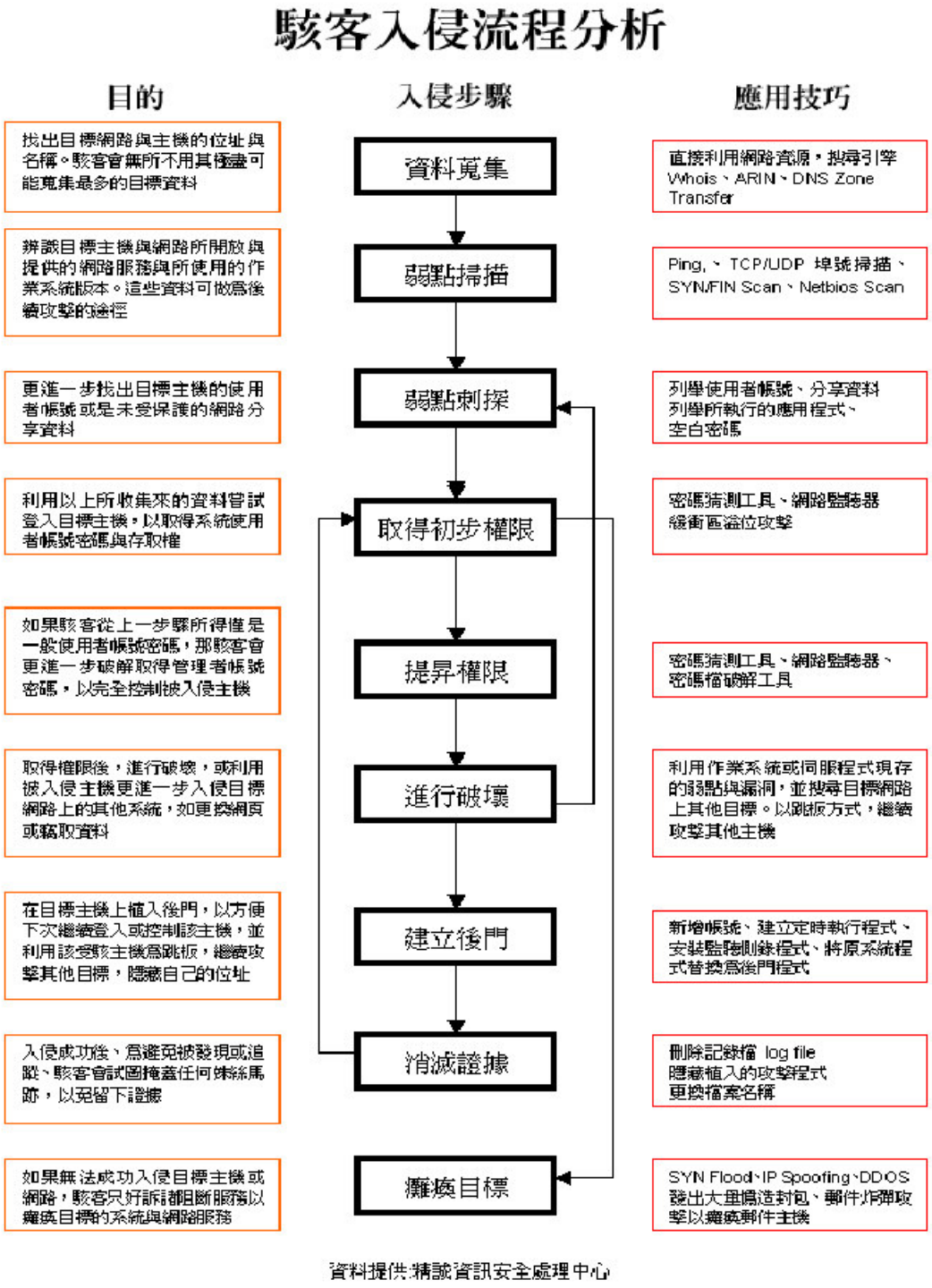
基本上網路安全的因素主要是來自兩點 。

1. 人為疏忽:自己本身自闢其他不安全路徑使整體網路有缺口、漏洞。

2. 惡意入侵:如駭客入侵(密碼破解、木馬)

人為疏忽或是不清楚如何設定網路導致不安 全部份,通常來自於自身對網路安全不夠熟悉,其實駭客也正針對這一點,一一探查漏洞並使網路之安全亮起紅燈,只要明瞭駭客的用意及手法,自然可以將漏洞一一填補 。

4.3 駭客入侵的方法



資料提供:精誠資訊安全處理中心

圖4.3.1 常見駭客入侵之流程圖

4.3.1 鎖定目標

當一個駭客在想入侵之前，一定會先對攻擊的目標做好完整的資料收集，就像強匪想洗劫銀行，不會貿然走進銀行，而去強奪財物，他們會事前做好演練，規畫犯案路線，收集有關銀行的資訊一包括守衛交班時間、監視器的位置、櫃台人數...等一切相關資料。駭客也是如此，他們必須先勘查目標的網路架構、系統漏洞之後，才會發動一次致命的攻擊，並且有系統地從眾多情報中去分析，找出關鍵性的缺口，然後再從缺口自由進出。

4.3.2 資料的來源

通常駭客會先從網頁上找資料：總公司住址、電話號碼、E-MAIL文件、相關新聞之類約有關文件。再來網路上有很多開放式的資料庫，也是駭客收集資料的天堂，如：臺灣的TWNIC資料庫(www.twnic.net)或是國外由Network Solutions(http://networksolutions．com)所架設的InterNIC資料庫。這些資料庫都可以利用whois的搜尋功

能，在輸入公司行號的網域名稱後，就可以得到公司所註冊網域相關資料，如：登記者 名稱、住址、電話、聯絡人，...等資訊。得到相關資料後，駭客就可以進一步利用DNS搜集更多的資訊，我們都知道DNS是分散式的資料庫，主要是負責IP和主機名稱之間對應查詢，假設DNS設定有漏洞，就很有可能造成資料外流。舉例來說，通常系統管理者都會有兩個DNS伺服器，也就是說有一個主DNS和一個次DNS。而次DNS的任務就是從主DNS取得新的資料庫進行更新的動作，或是萬一主DNS故障時，可以作為替代之用。但是，問題就出在這裡，如果DNS設定有問題，駭客可以偽裝成次DNS，向主DNS提出更新資料庫的動作，如此一來，駭客就可以得到公司內部主機所有的IP位置，也就是把自己國家的戰略地圖給予敵人一般。而駭客是如何向DNS取得這些資料？其實很簡單，在WINDOWS NT底下(或是UNIX)有一個nslookup程式，透過交談模式，就可以得到豐富的重要資訊。當駭客找到"地圖"之後，就開始去觀察地形-檢查它存取路徑(access path)和網路拓樸(network topology)，利用最簡單的tracert指令(在windows nt和unix或route底下叫traceroute)，它會

利用封包(packet)中的TTL(time-to-live)，每經過一台路由器(route)時，TTL值會被減去1直到0。因此，tracert指令就像一個計數器，每經過一個節點(路由器)就計算一次，這樣一來就很容易知道封包所行進路 線圖，作為探測目標網路拓樸最好的方式。不過，有些企業會加裝防火牆來攔劫、阻擋這些traceroute的探測封包，造成ICMP或UDP的封包沒有辦法ECHO回來，就會出現星號(*)的字樣。

4.3.3 駭客入侵--清查

假設資料收集就像是銀行搶匪在做地形勘查，清查就是去找銀行所有的出入口、門窗以及任何有破綻的地方。駭客透過資料的收集已經得到一個初步的輪廓，進一步就是要去找出哪些系統主機是有用的、哪些有正常運作(alive)，透過ping的清查和通訊埠的掃瞄(port scan)及其他駭客工具，就可以發現很多主機重要的漏洞。利用ping的指令可以找一個網路區段IP裡有哪些主機在運作，ping一般會對目標送出一個封包，如果對方主機是存在的就會自動送回一個回應封包(echo packet)，這樣就可證明目標主機正在運作。不過，也有例外

的情況發生，遇到具有安全觀念的系統管理者，就會將這個封包ICMP攔下，不會有任何回應封包，製造主機不存在的假象。這時候，駭客也不是省油的燈。依然可以用別的方式來證明主機是否存在。使用通訊埠的掃瞄試著去連結電腦主機的TCP和UDP的所有埠，檢驗它所提供的服務種類或是有在聽(Listen)的狀態。只要找到某幾個在聽的埠，通常主機一定是存在的，而且是攻擊的最佳缺口，通常TCP80埠都是存在的，因為它是一個常用的埠，大部份網域都會准許它經過路由器，甚至於進入核心區域，所以利用TCP PING掃瞄就會向主機送出TCP ACK的封包。如果對方TCP通訊埠沒關，是正常運作就會回應一個TCP SYN/ACK的封包，然後攻擊者電腦會自動送出一個ACK封包。如此一來就建立一個TCP的連線。最後，駭客會清查出所有潛在的目標主機，為下一個階段作準備。

4.3.4 駭客入侵-資源分享

當銀行搶匪收集完所有的資訊，決定好搶奪與逃亡的路線之後，進一步地就是針對銀行保險庫裡的現金、黃金、股票，做一個完整的搜查，去確認保險庫何時啟動、裡面到底

有多少有價值的東西等。此時的駭客也是如此，他們要去對目標系統作查詢的工作，找出可以分享的資源和使用者群組以及其應用程式和標誌。在微軟的作業系統裡，對於網路的資源共享部份，給予很大的方便，其用意也是希望大眾能互相交換資源，想不到的是，也正好給駭客一個入侵的管道。駭客使用WINDOWS底下的NET VIEW的命令就可以列出相關的網域和網域上所有的機器，更進一步地建立連線之後，透過NetBIOS可以去列舉遠端電腦系統上的資源分享，就可以清楚的看到分享的檔案資料，甚至於修改或刪除這些資料。此外，利用nbtstat的命令也可以找出遠端系統的NetBIOS表，分析其中的使用者資訊，如果運氣好的話，很快地就可以將使用者帳號密碼抓回來，好好的享用了！當然要防堵這些資源外洩，最好的方法就是關閉TCP和UDP埠從port 135-139全部濾掉，就可以避免駭客從NetBIOS的漏洞侵入。在UNIX底下也有類似的指令-finger，這算是UNIX系統中最老掉牙的招數。仍然是有駭客嘗試去用它，對付一些不負責任且無知的網路管理員，透過figer指令會自動列出使用者資訊，讓攻擊者很清楚猜測知道root的動態。唯一的解決之道，就是將finger的

功能關掉，就能省下很多不必要的麻煩。

4.3.5 駭客入侵-嵌入（Embedding）

在掌控攻擊目標主機之後，接下來就是希望能夠在被發現入侵行為時，也能繼續控制攻擊目標主機。技巧乃是在控制系統之時，便下載攻擊程式於此攻擊目標主機上，並將之安裝並隱藏存放於磁碟內。為了避免被偵測出來，入侵者多半會將這些程式碼覆寫（overwrite）在那些存在但卻很少被用到的系統檔案上。這些系統檔案有著固定的存放位置和存取路徑，然而卻很少被使用到。故存放於此，將使得攻擊程式不易被發現，因為它看起來就像是一般的系統檔案一樣。另一種相似卻更狡詐的變形方式，乃是將這些攻擊程式碼分割，並存放在一些存在且少用的系統檔上。所以即使被發現某一系統檔案有被異常的更動，也不會被認為跟入侵行為相關連，而且也很難將分散在各處的攻擊碼完全清除乾淨。

另一種隱藏這些攻擊洞口（attack bot）的方式，乃是在攻擊目標主機上安裝一後門（back-door） 後門（back-door），顧名思義乃是存在但不易被察覺的一種進入窗口。此程式會靜

靜地存放於磁碟當中，並等待一特殊的資料片段出現來啟動它。一旦被啟動了，此程式將會重組散佈於各處的攻擊程式片段，並可能會自動下載更強而有力的入侵程式於系統上。

請不要小看了這些攻擊方法，因為在下載資料的同時，此程式有可能將攻擊目標主機的所有安全系統給暫時終止。這也代表著這些下載行為，一點也不會被偵測或紀錄於安全系統之內。

4.3.6 駭客入侵-資料擷取和修改

在掌控攻擊目標主機之後，入侵者便可恣意而為做任何想做的行為。最常被發現的破壞行為，就是某些資料被擷取和修改。如同前面曾經提到過的，入侵者刪除系統log檔，以湮滅其入侵的證據；覆寫攻擊資料於系統檔案，以隱藏其攻擊資料等。此外，像在「電腦叛客」一書當中，Kevin將某一醫院的昂貴帳單，將之修改轉寄於其敵對之人。甚至更可怕的，則可能修改銀行帳戶存款金額。雖然此事尚未見於報章媒體上，然而就算曾經發生過，受害銀行通常不會將之公布於世，因為這將嚴重損及其名譽。

資料擷取和修改的另一意義，則是代表著入侵者將能夠對其所要傳送的資料作加密的工作，使之不再赤裸裸，而像是穿著一層保護衣一樣地傳送於網路上。例如：在網頁（HTTP Web）資料的傳送上，其攻擊的資料便以二進位（binary）的模式，將之假冒存放在JPEG，GIF...等圖檔上。此種方式將會讓大多數的安全系統以為只是正在傳送合法的圖形而已。

隱藏傳送的資料還有另一種方法，乃是經由統計、分析攻擊目標主機的通訊樣式（traffic pattern），也就是瞭解攻擊目標主機何時、何地、採取何種的通訊協定(protocol)和對哪部主機做通訊。藉由此，入侵者將可模擬相類似的通訊樣式，假造合法的通訊封包，以躲過安全系統的監測。

4.3.7 網頁入侵(Homepage Hack)

網頁的無遠弗介與廣告力量可以說是任何傳播媒體無法比擬。不過，網頁對駭客而言可是最喜歡的遊戲。近年來，總是有很多大陸駭客將臺灣政府網頁一再竄改，貼上五星旗圖片寫下一些反動字眼。而臺灣的駭客也不干勢弱，將大陸官方網頁也貼上青天白日的國旗，飄揚在海峽的對岸，兩岸網友互駭

的情況就這樣經常發生，因此網頁安全就成為一個重要課題。在網路上網頁的通訊埠為80、81、8000、8001和8010等等，大部份的駭客都是利用此通訊埠來攻擊，因為這些是唯一可以進入您網頁的節點，所以只要守住這道防線就可以防止駭客入侵。理論上是如此，但是事實上確沒有我們想像中的容易。在網頁資料裡，駭客可以很容易的從瀏覽器中的原始檔HTML文件，去獲得一大堆豐富資訊，包括電子信箱號碼、個人資料、電話號碼甚至於一些有價值的注解和Java原始檔案。一旦駭客得到這些資訊之後，便可以加以分析找出網頁密碼，進一步竄改網頁或是入侵電腦偷取重要資料。防範的對策就是盡量不要將一些重要資訊遺留在自己的HTML的原始檔中，注解只要用自己看的懂的方式來處理,就可以避免駭客從中獲取任何的情報。

4.4 網路安全的目標

- 保護資料免遭意外
- 保護資料免遭蓄意破壞
- 維持資料在需要時可取得

4.4.1 邁向網路安全的步驟

網路安全系統的基本目標與任何一種電腦系統的安全目標十分地相像:

1.保護資訊免遭意外破壞或修改

2.保護資訊免遭蓄意破壞或修改

3.當已授權之使用者需要資料時，就確保其可取得，而且是使用者可使用的格式

第五章 未來專題發展與補強

在這個專題裡我們在Recover的部分還有點小問題,目前我們初步是打算用shell寫,將保護的資料夾備份到Back_up資料夾裡,在發現Tripwire偵測到資料夾被更改後,就到Back_up資料夾裡將備份檔Recover到被更改的資料夾裡,不過這個部分還有點問題,因為假設駭客入侵,他直接更改Back_up資料夾裡的資料,就很麻煩了,所以我們想的另一個方案,是找尋Linux上的Back up軟體(Ex TAR)並且將此軟體跟我們原先兩個軟體再整合於同一平台,這也是本專題未來發展之一,我們可以用這種技術,慢慢的把我們需要的功能,去尋找到適合的軟體,將他們全部兜在一起,整合於一個平台上,以利於使用者使用,而且我們選擇Linux來做也是有原因的,因為Linux上的軟體,幾乎的是免費軟體,功能又強大,最重要的是,有大部分有OpenSource,目前我們程式部分還有待加強,不過當我們程式精進,接下來的整合,就不只是軟體之間的相互支援,而是可以將其內部功能抽出來,直接整合於一個程式當中,慢慢的兜出一個完整的防護系統,這也是我們最後的目標.

另外的發展就是在我們選用的軟體都為OpenSource的軟體,所

以其程式碼容易取得,在將來我們可以將我們所需的功能一部分一部分的從這些軟體中抽取出來,在將其製作成模組,做個完整的整合,或是加上一些新的網路安全技術(如誘捕技術)以達到更多層的防護系統,可以在資訊防護上達到更好的效果

第六章 專題心得與所遇難題

6.1 心得

從一開始完全不知道如何著手，慢慢開始先學習 Linux 的使用，一開始上手感覺難複雜的，因為 Windows 用習慣了，所以調適了一段時間，加上在 linux 上比較普遍的編輯器只有 gcc，gcc 不像 vc 一樣的強大，也沒有 vc 的簡易，加上我們組員的程式功力不是很好,所以一開始作起來困難重重,好不容易把 linux 弄好，接下來的 Nessus 及 Tripwire 兩個軟體也讓我們傷透腦筋，因為這兩個都是功能很強大的工具軟體，從一開始的設定外加 Report 的解讀，都花了好一陣子，不過，全依賴著老師,每個星期不斷的督促、解惑、提點,以及助教不厭其煩地替我們找尋相關資料、訊息，雖然一直遇到技術上、程式設計上的瓶頸，但他們未曾放棄，仍是在我們一籌莫展時伸出援手，使得我們心中感動不已。

這個專題讓我們學到了不只是技術層面的東西 ，也讓我們知道在網路安全這門領域裡，有太多太多不同的東西，我們需

要學的還有很多，也難怪人家說資訊不斷的進步，所以我們一直不能落後，這次專題除了讓我們在原本學過的C語言上有慢慢的進步外，還另外學到了Linux系統裡的排班，以及Shell如何運作，讓我們了解學習是永無止境的，尤其在資訊領域中，如果不求上進，很容易就被淘汰，唯有努力不懈才是學習最大的指標，還有也讓我明白小組的重要，一個好的專案還是要靠大家合作才能夠製作出來，因為一個人的力量也許不是很大，但是大家一起做除了互相激勵以外，還可以互相提供方法，互相討論，這樣做出來的東西也會比較完善，所以我們將會記取這次的經驗，將這種精神放在未來的課業上甚至於未來職場上，才能夠有所作為。

6.2 專題所遇困難

由於我們再大一到大三都是以學校活動為主,以致於在課業上所得之專業知識少了些,所以再一開始做專題時,一度不知如何下手,尤其是要面對的是完全陌生的Linux環境,還有在安裝Tripwire和Nessus這兩個軟體,也是因為不熟悉而遇到很多問題,尤其是在Tripwire部分,因為Tripwire在某些版本的Linux上跑起來都會有問題,這方面也讓我們苦惱很久,在弄熟之後,接著面對的就是把整個架構慢慢弄清楚還有程式的實行,都對我們有點難度,在程式方面我們重新學習,所以再一開始的實作上進度很慢.還有再讀檔和排班的程式實作上,也讓我們吃足了苦頭,因為Tripwire的Report檔以twr格式來儲存,如果用VI來開啟,會變成Report裡有許多特殊符號,所以我們原本想說用fopen直接開的方法失敗,後來去問同學,發現popen可以解決我們的問題,因為Tripwire有內建指令可以開啟Report並且輸出至螢幕上,再加上popen的使用才讓我們這部分順利解決,另外

要學習基本的Shell,還有就是要如何寫程式來分辨Tripwire的Report檔有沒檢查到被修改的部分,這方面也讓我們進度停頓了一下,不過還好的是雖然一度感到困難,但是我們整組不放棄的精神,還是將專題努力的完成

Appendix:

Reference

[1]Tripwire Open Source 官方網站說明文件

http://www.tripwire.org/

[2]Getting Started with Tripwire (Open Source Linux Edition)

http://www.linuxsecurity.com/feature_stories/tripwire-printer.html

[3] Tripwire for Linux 系統稽核，作者：伊原秀明，譯者：蘇秉豐，O'Reilly 出版，November 2001.

[4]永遠的 Unix

http://www.fanqiang.com/big5/

[5] http://safe.ip-market.com/article.php?sid=38，

http://sourceforge.net/projects/tripwire/

Tripwire介紹

[6]Nessus 官方網站說明文件

www.nessus.org

[7] 易學易用 Red Hat Linux 7.x+ CLE1.0

普悠瑪數位科技 著

碁峯資訊股份有限公司印行

[8] http://fetag.org/

飛鷹工作室駭客相關技術文件與安全防護