Global Information Assurance Certification Paper

GIAC Security Essentials Certification (GSEC)
Practical Assignment
Version 1.4b

Case Study in Information Security

Deploying perimeter firewalls mailing encrypted log reports.

Written by:
Xavier Guilbeault

Xavier@secureops.com

Submitted on the: 2002-09-28

Index table

Goal

The goal of this paper is to detail the different steps involved in the deployment of dedicated perimeter firewalls and to configure them to mail log reports in an encrypted format. The software used through the procedure is only open-source software, ensuring a minimum cost in that area.

Once the reader followed all the steps explained, he will have deployed an OpenBSD firewall, logging all suspicious traffic and emailing the reports to an email account after encrypting them with gnupg (Gnu Privacy Guard). The installation of such a firewall will put in place a perimeter defence for a single host or a LAN , depending on the desire of the reader.

1)Before

It is very common in information technology to see employees work from home. As network connectivity increases, more and more people get to work remotely. This situation causes some problems on the point of view of security. Because of the different types of careers that touch computers, a large number of people may not be sufficiently security aware to ensure that there is no security threat. Given such a case, the security of the computers, being unmanaged, may be at risk because of the awesome quantity of attacks taking place. As an example, records from the Internet Storm Centre shows that, only for queries targeting the windows share system, from 30000 to 343807 "attacks" a day are registered [1]. "The primary means of securing a private network against penetration from a public one is a firewall" [2], states a whitepaper from Cisco Systems, highlighting the necessity of such a perimeter defence.

However, as says Bruce Schneier, founder of counterpane and maintainer of the crypto-gram mailing list, "Security is not a product; it's a process" [3]. Therefore, series of steps must be taken in order to pass from the plain workstation connected to the internet to a fully protected private LAN. One of these concerns the installation of a firewall and/or intrusion detection system. In a situation where the different workers connect to the internet and exchange their work directly, with no form of protection than a basic antivirus, it is next to impossible to stop an attacker to compromise the system, even less for you to detect that such a compromise as been made

For that reason, the importance of monitoring the different log files is capital if we want to be sure that we are safe. Such a process will inform the administrator of any unusual activity and can this way provide a better response time. However, it cannot be taken for granted that the system installed through this document will be unbreakable, as such a though can only provide a false sense of security which can be way more dangerous than other threats. "Name one vendor that hasn't been taken down. They all have." Declares Greg Shipley, director of consulting services for security firm Neohapsis. [4] For that reason, the administrator taking care of these systems must try to stay informed on all security problems concerning his systems, or at least concerning the different software used.

The installation of a personal firewall on a home machine does provide a base

protection but raises the issue of the trust in the user for not disabling it (consciously or not). Such a choice involves more issues in the point of view of security. Also, you cannot ask to a non-technical person with no implication in information security to monitor the logs of the IDS you installed.

In face of such a situation, it is the responsibility of the Security Officer to manage that the private LAN be secured. A separate and dedicated firewall answers to the precedent concerns and minimizes the probabilities of a compromise because of the few services it runs.

1-2) Why OpenBSD ?

OpenBSD is an operating system very similar to UNIX which is being developed by a team of open-source developers renowned for their paranoid vision of computer security. This makes them one of the most proactive team in the field of information security by their approach at addressing issues and auditing the system. The main goal of this project is to deliver the most secure OS available, and they do answer to that requirement. . "The only other OS distributor [than Microsoft] that does regular security audits on source code is the OpenBSD project."[5] As a consequence to this commitment, their web-page states proudly: "One remote hole in the default install, in nearly 6 years!"[6]. In overall, the emphasis given on security by this project makes it the ideal open-source solution for protecting a network.

1-3) The services

To take out as much as possible any possible threat, the bare minimum services will be triggered. Only sshd (the OpenSSH secure shell server) will be accepting incoming connections from the internet, the other service (mail) being accessible only from the localhost. OpenBSD, at the time of writing this paper, is at version 3.1, which will be the version used through all this document.

1-3-1 The firewall

- **pf** :The firewall will be set up with pf (Packet Filter) which is now the default firewall shipped with OpenBSD since version 3.0. It is a state full packet filter that also deals with packet forwarding.

1-3-2 The logging facilities -

- **iplog 2.2.3** : Iplog is an open source software which logs all suspicious TCP/IP and UDP traffic to a file.
- **snort 1.8.7** - Snort is a full-blown IDS, all open source, with an extremely active community. The software offers very good results. However, its installation will not be covered in this document, to use snort, you can refer to the "Snort Users Manual" available at http://www.snort.org/docs/writing_rules/ .
- **pf** - Rules will be added to the firewall to log all traffic, giving the ability to analyse in depth any suspicious activity.

1-3-2-1 Why multiple logging facilities ?

Because of the subtleties involved in some attacks, we want to have all the information possible when investigating an event. The software mentioned above may sometimes miss, or misinterpret, some traffic that may be of valuable use to the monitoring/forensics. It is mainly for that reason that all of the software is set up. It is evident that monitoring all log files can be a painful process, this is why it is recommended to monitor only one program's log files (iplog or snort) and to go dig deeper in the log files if the situation requires it.

1-3-3 Mail

- **Postfix 1.1.6:** Postfix is an efficient MDA (Mail Delivery Agent), that will take care of sending log reports to the address of the administrator
- **gnupg 1.0.6**: (Gnu Privacy Guard) - Free implementation of PGP (Pretty Good Privacy) that will be used to encrypt outgoing emails, ensuring that the log files content stays private

2) During

2-1) Installation of the system

1) Requirements.

Multiple other set-ups are possible, but if you have a configuration like this one or close to it, you will have enough CPU power to accomplish the necessary tasks. Notice that the architecture used here is i386 but that a similar setup can be made with different ones.. OpenBSD supports the following architectures: alpha, amiga, hp300, i386, mac68k, macppc, mvme68k, sparc, sparc64 and vax.

Configuration :
- Pentium 75 MHz
- 32 Mo of ram
- 600 Mb hard drive
- 2 Ethernet cards

2) Installation of OpenBSD

2-1) FTP

This may be the easiest way to install OpenBSD on your computer. The installation procedure is very straight-forward and should not cause any problem. The only point that will be looked at is the partitioning, due to the fact that a lot of existing resources detail the installation procedure.

For links to such resources, see the end of the present paper.

2-1-1) Getting the boot floppies

You can download the boot floppies from the following location (Note: change 3.1 by the latest release): ftp://ftp.openbsd.org/pub/OpenBSD/3.1/i386/

You will need the file names "floppy31.fs". If the boot disk does not boot with this file or that some hardware is not detected, you can download the other floppy images that are: "floppy31B.fs" and floppy31C.fs".

2-1-1-1) Creating boot floppies from Windows.

Once you have the file, you must copy it to a floppy disk. If you are on windows, you will have to use an utility called rawrite (ntrawrite for windows NT/2000). You can find the program(s) at the following location:
ftp://ftp.openbsd.org/pub/OpenBSD/3.1/tools/

Then, type on the command line:

C:\>rawrite file.img a:
or
C:\>ntrw file.img a:

2-1-1-2) Creating boot floppies from UNIX .

If you are on UNIX, you may use the dd utility. Issue the following command at the shell prompt:

 # dd if=floppy.image of=/dev/floppy

Where /dev/floppy is your floppy disk drive (/dev/fd0a on OpenBSD, /dev/fd0 on Linux).

2-1-2) Booting from the floppies.

Once you have the floppies, boot the system you wan t to set up with the disk. Since installing OpenBSD is not the main goal of this paper, the instructions covered here are very minimalist. Once again, for a complete installation procedure you can take a look at the different resources at the end of this paper.

2-2) CD-ROM

Since OpenBSD is made by volunteers programmers giving their time and dedication to the community, you may encourage them by ordering a CD -ROM of the latest release (http://www.openbsd.org/orders.html) .Once you have the CD, you may boo t from it and install the software directly.

2-3) Partitions and packages

On a 600Mo hard drive, the partition scheme would be the following:
/ - 336Mo
/var - 150Mo
/tmp - 50 Mo
swap - 64Mo

Once the partitioning is done, you must select the following packages from the location you install OpenBSD :

- bsd
- base31.tgz
- comp31.tgz
- etc31.tgz
- man31.tgz
- misc31.tgz

Note : Notice that "31" stands for version 3.1 of OpenBSD, if the version changed by the time you read this paper, please consider installing the latest packages .

2-4) Base configuration

Once the base system is installed you should end up with a login prompt after rebooting the system. If this is not the case, then restart the installation and/or try to see what is happening, "Google Groups" (www.google.com) is an excellent source of solutions to common problems.

2-4-1) Adding new user

First thing you should do after logging as root is to create a new user. You do that by typing the commands (for more information on the command, type: "*man adduser*"):

#adduser

You will probably want to add it to the group wheel so you can "*su*" to root.

2-4-2) Base services (sudo , ssh , stopping inetd)

2-4-2-1) Sudo

Sudo is a program that allows normal users to run programs as root (or another user). It is very useful for the reason that it gives the ability to track who did which command (a username being more useful than the generic 'root').

Since you will be the only one accessing your firewall, place the following line in the file '/etc/sudoers' that you edit with the program 'visudo' which is a maintenance utility for "sudo" configuration files. You must then run as root the following

command:

#visudo

Then add:

user ALL = ALL (ALL)

Where *user* is the name of the user you created. Th is will allow you to have complete control over the system. After this, you issue a command in this way:

 # sudo command

You will be prompted with a password, remember that this is NOT the root password, but the one of the user you are logged in with.

Once this is made, it is recommended generating a new root password that will be extremely long and difficult. Since there is no more reason to log in as root, it does not matter if you cannot remember the password, as long as it is extremely resistant to crackers (you can change it after with "sudo passwd root") .After this step is taken, you may want to give a particular look at the authentication logs of the firewall for a root login, which would be a sign of compromise.

2-4-2-2) ssh

The sshd server is started by default on OpenBSD, so there is very little configuration to do. The only thing you may want is to transfer your different ssh keys to the host. To learn more about ssh, type:

 # man ssh
and
 # man sshd

2-4-2-3) stopping inetd

Although no service is run by default on inetd , I recommend disabling all of the ones that are placed, as they could be used to gain information on your system (date, uptime...).

To do so, change the line:

inetd=YES
by
inetd=NO

in the file "*/etc/rc.conf* ".

2-4-2-4) Configuring PF and IP forwarding

To enable pf to start on boot of the machine, change the variable " *pf=NO*" in "*/etc/rc.conf*" to "*pf=YES*".

Since the computer we install is going to do NAT (Network Address Translation) we must enable it to do IP forwa rding.

To do so, edit the file " */etc/sysctl.conf*" and uncomment the following line (you do that by deleting the "#" character at the beg inning of the line) :

net.inet.ip.forwarding=1 # 1=Permit forwarding (routing) of packets

The feature will then be initialised at the next reboot of the machine.

2-5 Software installation

2-5-1 Introduction :

The software introduced below is all available via the ports tree of OpenBSD. Before installing any of these, you will have to download the " *ports.tar.gz*" archive and decompress it in the /usr directory. You do that with:

#ftp ftp.openbsd.org
*ftp> cd /pub/OpenBSD/**X.X/arch*** (XX being the last release, now being 3.1 and arch being the system's architecture , i386 for Intel)
ftp> get ports.tar.gz
ftp> bye
#cd /usr
#sudo mv /location/to/ports.tar.gz ./
#sudo tar xvfz ports.tar.gz
#sudo rm ports.tar.gz (optionnal)
#sudo chown -R your_user ports (This way you will be able to compile programs without being root)

2-5-2) iplog

Iplog is a very useful util ity that let you analyse the network traffic on one of your interfaces. Iplog will be used to log traffic than will be analysed in the monito ring process. The program has a little function that gives the ability to fool nmap by responding differently to th e crafted packets it sends. This may protect us from script-kiddies in the case where a new OpenBSD vulnerability may be found, since the attacker will not have a reliable way of guessing the OS we run.

2-5-1-1) Installation of iplog

To install iplog, you must first compile it, then install the different binaries. To do so, type the following commands:

 # cd /usr/ports/net/
 # sudo make
 # sudo make install
 # man iplog

Then you will want to make sure iplog starts up at boot time. For this, add the following lines to "*/etc/rc.local*" which is a script file that is read at boot time :

```
#Iplog startup
if [ -x /usr/local/sbin/iplog ] ; then
        /usr/local/sbin/iplog -d -z -V -s -i interface
        echo iplog started...
fi
```

Change "interface" by the name of your interface connected to the internet.

2-5-2) Postfix

Postfix will be used as a mail transfer agent, ensuring of the delivery of mails to the appropriate SMTP relay server.

Note: Any other MDA will do the same job, the reason postfix is used here is because of the ease of use compared to *sendmail*.

2-5-2-1) Postfix installation

As *iplog*, *postfix* is available via the ports tree. To be installed, it must first be compiled, to do so, issue the following commands at the command prompt:

```
#cd /usr/ports/mail/postfix
#sudo make
#sudo make install
```

2-4-2-2) Postfix configuration

Here are some links pointing to resources on how to manage postfix.

From http://www.postfix.org - The Postfix Home Page
http://www.postfix.org/motivation.html - Postfix Overview - Introduction
http://www.postfix.org/basic.html - Postfix Configuration - Basics
http://www.postfix.org/receiving.html - Postfix Anatomy - Receiving Mail

The configuration showed below is very minimal and does not give the full merit to the usefulness of postfix but it will be sufficient for the setup presented.

The postfix configuration files are found in "*/etc/postfix*". Below are the most important and are necessary for the firewall to send mail correctly. While it should work correctly with the following files, the author recommends opening a new tty

(CTRL+ALT+F*, where * is a number between 1 and 12) and typing the following command:

 # sudo tail -f /var/log/maillog

This will print the status of the mail system and will help you greatly if any problem arises.

2-4-2-3) "main.cf" - Postfix main configuration file

Location : /etc/postfix/main.cf

 # Main.cf - Postfix main configuration file
 # Option that will ensure that the mail will not be bounced back to the sender's
 # Very useful option, particularly in testing phase.
soft_bounce = yes
queue_directory = /var/spool/po stfix
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix
mail_owner = postfix
default_privs = nobody
myhostname = localhost
mydomain = localdomain
myorigin = $myhostname
mydestination = $myhostname, localhost
 # Ensure that we do not setup an open -relay SMTP server that spammers could use...
mynetworks = 127.0.0.0/8
 # Enter here the ip address of the SMTP relay host that this computer
 # should connect to.
relayhost = [xxx.xxx.xxx.xxx]
 # Redirect messages depending on the recipient.
recipient_canonical_maps = hash:/etc/postfix/recipient_canonical
 # Ensure we give a valid hostname as a mail server (bob@localhost would not be
 # accepted by well configured m ail servers.
sender_canonical_maps = hash:/etc/postfix/canonical
alias_maps = hash:/etc/mail/aliases
alias_database = hash:/etc/mail/aliases
smtpd_banner = $myhostname ESMTP Access denied

2-4-2-4) canonical - sender aliases

Description: This file is us ed to resolve the sender's email address. If you do not do this, your mails may be bounced by the recipient's server because of the invalidity of the address. For example, if you call you machine "firewall" and you mail a message using user "admin", the em ail address of the sender will be "admin@firewall", which isn't a valid hostname and will therefore be rejected by the server. For information on the syntax:

#man canonical

Here is a base config file that should answer your needs:

user@localhost *your_email@your_server.net*
user *your_email@your_server.net*

root@localhost *your_email@your_server.net*
root *your_email@your_server.net*

This file must be converted by the postmap command , the result, an indexed file in dbm or db format, is used for fast searching by the mail system. So when it is complete, you must issue the following command:

 # postmap /etc/postfix/canonical

2-4-2-6) Start postfix

To start postfix :
#sudo postfix start

It is recommended to keep an eye on *"/var/log/maillog"* for any errors that might appear.

And add the following lines to the " */etc/rc.local*" file so postfix restarts on boot:

```
# Postfix startup
if [ -x /usr/local/sbin/postfix ] ; then
        /usr/local/sbin/postfix start
        echo Postfix started...
fi
```

2-5) setting up gnupg

2-5-1) Introduction

Gnupg (that stand for GNU privacy guard) is a free implementation of PGP. Is gives the ability to encrypt emails with the use of a prime numbers key pair. For more information on gpg , see http://www.gnupg.org .

More precisely, **The GNU Privacy Handbook** is a very good resource to start-up with gnupg and to understand how it works:
http://www.gnupg.org/gph/en/manual.html

2-5-2) Installation

Gnupg is available from the ports tree and is installed by the following commands:

#cd /usr/ports/security/gnupg/
#make
#sudo make install

2-5-3) Configuration

Once installed, you will want to import your own public key so that you will be able to encrypt your mails. To import the key, enter the following command :

 # gpg –import key.pgp

It is recommended not creating your key pair on the firewall itself, because if you forget to move your private key from there and the firewall gets compromised , the attacker will be in possession on your key and will therefore be able to easily decrypt all the messages that you send.

You may also create a key pair for the firewall itself that would be used to authenticate it as the sender (with a signature). Bu t by doing this, you will have to make a NULL password key, so the perl scrip can use it, which can be dangerous if someone breaks into the firewall and starts sending false reports, giving you the impression that all is okay. But, if you do not use that s igning process, anyone spoofing the senders address can encrypt his/her own report file and make you think it comes from the firewall...

Once the firewall key is created, use it to sign your own public key.

To enable signing , add the " -s" option to the call to gpg in the script. To do so, you will have to change the following line :
if(system("/usr/local/bin/gpg -e -o /tmp/$date \.asc -t -a -r $opt_k"))
by
if(system("/usr/local/bin/gpg -e -o /tmp/$date \.asc -t –s -a -r $opt_k "))

2-6) Firewall setting

2-6-1) needs

The needs of the firewall you need may vary greatly from the ones that are proposed here, in any case, it is greatly recommended to read the pf-howto (http://www.deadly.org/pf-howto) and the appro priate man pages (pf.conf, pfctl, pflogd, ftp-proxy, pf) so that you ensure you really understand what you are doing. In other cases, the following rules should be enough in most cases.

What we want here is a firewall with a minimal number of open ports. So the default policy should be to block all inbound traffic that was not initially started by an internal host. This is based on a CERT ® Security Improvement Module which states "that all network traffic that is not explicitly permitted should, by defau lt, be denied" [7].

Therefore, the only open port on the firewall itself will be ssh (22).

Also, we want to make it transparent to the home user, so he will be able to do his normal activity without interference. That being browsing the web, and using all network related clients. Pf answers to that concern since it is a state full packet filter that keeps in memory the state of the connection so that it does not block legitimate traffic initiated from an internal host. Also, since the program handles packet forwarding, there is no need for another software.

2-6-2) rules of firewall and NAT

Description: NAT (Network Address Translation) provides a way to share an internet connection between multiple hosts. The host doing the NAT is connected to the internet (here, the firewall) and forwards the traffic initiated from internal hosts to their destination. All replies from the destination are then forwarded back by the firewall to the internal computer which started the connection.

The configuration file is the same as pf ("/etc/pf.conf") . Which syntax can be learned in its manpage ("man pf.conf").

```
############################
# pf.conf - packet filter configuration file

# variables, change the interfaces here for the appropriate ones
externalNic="fxp0"
internalNic="fxp1"

# The packets are run through normalization/defragmentation.
scrub in all
scrub out all

# NAT RULES

nat on $externalNic inet from $internalNic/24 to any -> $externalNic

# Default policy - block all inbound traffic - logging activated
block in log on $externalNic from any to any

# Hide the firewall from pings
#block in log on $externalNic proto icmp from any to any

# Block in all non -reglementary addresses that should not be seen on the internet.
# We also log them because of its anormal nature
block in log quick on $externalNic from { 127.0.0.1/8 , 172.16.0.0/12,10.0.0.0/8, 192.168.0.0/24} to
any

# Let access to the sshd server for remote administration
pass in quick on $externalNic proto tcp from any to $externalNic/32 port = 22 keep state

# Let internal traffic pass
pass out proto icmp from any to any keep state
pass out proto tcp from any to any keep state
pass out proto udp from any to any keep state
############################
```

To apply the rules, issue the following command:

#sudo pfctl -f /etc/pf.conf

and to display them:
#sudo pfctl -s rules (firewall rules)
or
#sudo pfctl –s nat (NAT rules)

2-7) Final configurations

2-7-1) Generating reports

To generate and send the reports, a little Perl script will be used by cr on, executing it at a given interval. The script, called reporter.pl, is available in Annex A and is very simple. It basically takes 3 options from the command line:

Reporter.pl usage:

reporter.pl –e email@address -k key_identifier –l logFile1,logFile2

-e email@address.net :	*to specify the email address to send reports to.*
-k key	*To specify the public key to use to encrypt the outgoing reports, this can be the email address associated with your public key.*
-l log_file2,log_file2	*Comma separated list of log files, specifies the different files to include in the report. You must write the full path of each log files.*

You will also have to create the directory "/var/log/firewal l/" for the script to work.

2-7-2) Cron

Cron is used to issue commands on the system at a given time interval. To add "reporter.pl" to the crontab, issue the following command:

 # sudo crontab -e

Then add the following line at the end of the file:

*0 0 * * * /location/to/script/reporter.pl –e email@address *
-k email@address -l /var/log/messages,/var/log/pflog

Make sure the script is not writable by anyone an d that the log files are readable.

2-8-3) Assigning IP addresses

For the IP address assignation, you can allocate static address to your internal network as well as configuring a DHCP server (" *man dhcpd*").

The files containing the IP addresses given to your cards are of the pattern "/etc/hostname.if" where "if" is the network card.

For example, let say the internal NIC is called " *fxp0*", being assigned the address 192.168.0.1, and the external one, "*fxp1*", obtaining its address by DHCP from the ISP's server. We would see the following files:

#cat /etc/hostname.fxp1
dhcp
#cat /etc/hostname.fxp0
inet 192.168.0.1 255.255.255.0 NONE
#

3) After

Once this is done, the system is secured and ready to be used in production. The worker will be protected by th e firewall; all suspicious activity will be logged then mailed to the administrator. All of the reports are encrypted before they are sent and stored on the system (you may want to delete them), thus insuring the confidentiality of the data. As well, the a dministrator will always have access to the machine with "ssh" if a problem arises, giving a very good response time. If the OpenSSH server dies, the administrator will have to gain physical access to the firewall as no other remote control software will b e running on the machine.

The goal of isolating the workstation from the internet is achieved and no direct connection can be made to it (unless you specify a redirect rule in the pf configuration). The risk of seeing the machine compromised and not being notified is minimized and full reports of activity will be available for eventual forensics.

All of this is be transparent to the user, who can also add as much internal machines to his LAN, all of them being protected by the firewall. However, the fire wall offers no protection against trojans downloaded and executed, so it may be important also to monitor the internal traffic of the LAN to see if some unusual activity is taking place.

Citations :

1 – **Internet Storm Center** - http://isc.incidents.org/port_details.html?port=139

2 - **White Paper** - ***Internet Security for Small Businesses*** **-** *Cisco Systems, Inc.* - http://www.cisco.com/warp/public/cc/pd/rt/800/prodlit/fire_wp.htm

3 – **Crypto-Gram Newsletter - December 15, 1999** – [http://www.counterpane.com/crypto -gram-9912.html#SecurityIsNotaProductItsaProcess]

4 - **By Robert Lemos** - ***ZDNet News*** – **Bugs bust open 'unbreakable' Oracle 9i** - [http://zdnet.com.com/2100 -1104-831204.html]

5 - **Gary Rogers** - ***osOpinion.com*** - **Microsoft's New Focus on Security** - [http://www.osopinion.com/perl/story/16160.html]

6- **OpenBSD official page** – [http://www .openbsd.org]

7- **Configure firewall packet filtering** - *From the CERT® Security Improvement Modules* – [http://www.cert.org/security -improvement/practices/p058.html]

Bibliography:

OpenBSD – proactively secure UNIX -like OS - www.openbsd.org
GnuPG (Gnu Privacy Guard) - Free implementation of PGP - www.gnupg.org
Snort - The Open Source Network Intrusion Detection System – www.snort.org

Ressources :

- http://www.openbsd.org/faq/faq4.html
  Official overview of the OpenBSD installation proced ure (FAQ).

- http://www.openlysecure.org/openbsd/installation/openbsd -2.7.html
  Installing OpenBSD 2.7.

- http://www.nomoa.com/bsd/installation.htm
  Learn how to configure removable storage devices, manage software packages, administer users, installing the bash shell and other miscellaneous tips.

- http://www.dpsite.com/OpenBSDInstall.html
  Installation guide for OpenBSD 2.9, very similar to the 3.1 installation.

ANNEX A

Initial label editor (ent er '?' for help at any prompt)
>a a
offset: [63]
size []336M
Rounding to nearest cylinder :
FS type : [4.2BSD]
mount point: [none] /
> a b
offset: []
size: [] 64M
Rounding to nearest cylinder :
FS type: [swap]
>a d
offset: []
size [] 150M
Rounding to nea rest cylinder :
FS type : [4.2BSD]
mount point: [none] /var
>a e
offset: []
size [] 50M
Rounding to nearest cylinder :
FS type : [4.2BSD]
mount point: [none] /tmp
>q
Write new label? : [y]

ANNEX B

Reporter.pl

```
#!/usr/bin/perl
#####################################################
# Reporter.pl – generates a report from log files, encrypt its content
#                    and mail it to the specified address
#####################################################
# Script written by Xavier Guilbeault   # xavier@secureops.com #
# ############################## ######################

use strict;
use Getopt::Std;

use vars qw($opt_l $opt_k $opt_e);

getopt("l:k:e:");

if(!$opt_e) {
                print "You must specify a des tination email address for the reports to be
sent\n";
                &usage;
                exit;
}

if(!$opt_k) {
                print "A PGP key must be specified \n";
                &usage;
                exit;
}

if(!$opt_l) {
                print "At least one log file must be specified \n";
                &usage;
                exit;
}

my @files=split(/,/,$opt_l);


my $hostname;

open(HOST,"hostname |") or die "Could not execute hostname :$!";

while(<HOST>) {
                chomp($_);
                $hostname=$_;
}
```

```
my $logDir = "/var/log/firewall/";

my $date;

open(DATE,"date + \"%b%e-%Y\"|") or die "Could not execute date : $! ";

while (<DATE>) {

        chomp ($_);
        $date = $_;
}

close(DATE);

# Open Report file

open(REPORT,">","/tmp/".$date.".txt") or die "Could not open /tmp/$date  \.txt : $!";

# Print header

print REPORT "Report for the $date  \n";
print REPORT " -"x40;
print REPORT  "\n"x3;


# Cycle through each log files specified on the command line

my $file;

foreach $file (@files) {
        open(LOG,"<",$file) or die "Could not open $file : $!";

        print REPORT "<>" x 20;
        print REPORT " \nLogs contained in $file: \n";
        print REPORT "<>"  x 20;
        print REPORT " \n";

        while(<LOG>) {
                print REPORT $_;
        }

        print REPORT "<>" x 20;
        print REPORT " \n"x3;

        close(LOG);
}


close(REPORT);
```

```
if(system("/usr/local/bin/gpg  -e -o /tmp/$date\.asc -t -a -r $opt_k /tmp/$date\.txt")) {
        print "Error while encrypting report\n";
        exit;
}

if(system("/bin/rm /tmp/$date \.txt")) {
        print "Error while removing /tmp/$date \.txt\n";
        exit
}

#copy encrypted report to a given location on the computer

if(system("/bin/mv /tmp/$date.asc $logDir")) {
        print "Error while copying /tmp/$date.asc \n";
        exit;
}

# mail the report

if(system("/bin/cat $logDir$date.asc | /usr/bin/mail  -s \"Report from $hostname,
$date\" $opt_e"))
{
        print "Error while sending mail\n";
        exit;
}


sub usage {
        print "\nUsage:\n";
        print "reporter.pl -l files -k key -e email\@address.net\n";
        print "-l files : list of comma separated files to parse and  \n";
        print "                        include in the report. \n";
        print "-k key : The key identifier of gpg , can be email address  \n";
        print "                        or hex key value. \n";
        print "-e email\@address.net : email address to mail the report to.  \n\n";
}
```