Managing ECSC Linux Firewalls

John Leach

john@ecsc.co.uk

Matthew Hall

matt@ecsc.co.uk

Managing ECSC Linux Firewalls
by John Leach and Matthew Hall

Revision History

Revision 0.1 30 Sep 2002
Initial layout

Table of Contents

Chapter 1. Introduction

This document covers many aspects of installing and managing ECSC Linux based firewalls including, but probably not entirely limited to:

- Configuring Squid (Web proxy);
- Configuring Qmail (Mail system);
- Configuring Apache (Web server);
- Configuring FreeSWAN based IPSEC VPNs;
- Configuring Netfilter using SgtBash;
- Configuring PPTP based VPNs;
- Configuring PPP Dial-In Access;
- Writing firewall policies;
- Troubleshooting problems;
- Auditing of configuration.

As many (read: all) of the packages that make up an ECSC firewall are open-source and freely available this document will only cover specific firewall related details, plus anything helpful to troubleshooting. It is not designed to be a beginners guide and so expects the reader to have a good understanding of the underlying technology. Wherever possible other documentation will be referenced; we suggest you indulge in further reading.

This document may also cover peripheral systems, such as Microsoft Windows, Macromedia Coldfusion and Sophos Anti-virus with regard to making them operate smoothly alongside, or with, ECSC firewalls.

Layout and Style

The document is mastered in XML using the Docbook style definitions. The source should look beautiful, be readable and understandable.

The use of the English language should aim to impress even Oscar Wilde; oozing grammatical prowess and charm. Use of character case, anacronyms, spacing, punctuation, words and bad analogies should be consistent throughout.

Whilst marginally informal at times, this document should be serious. The information presented should be seen as reliable and correct and, if possible, should actually be reliable and correct.

Chapter 2. Upgrading Software

ECSC firewalls use the RedHat Package Management system (RPM) to manage the various versions of software installed and keep track of upgrades.

Lots of things can go wrong whilst upgrading RPMs on a remote firewall:

- Loss of services;
- Loss of remote admin access (usually ssh);
- Machine not booting after upgrades (not often detected until months later);
- RPM database corruption;
- Loss of your job.

APT

The APT utilties help keep RPMs up-to-date by comparing the local list of installed RPMs and versions with a remote, central database.

This central database, or APT repository, and RPM archive is accessible via HTTP at http://updates.ecsc.co.uk/apt. The APT utilities use standard HTTP to access these files.

Upgrading Gnu C Libraries (glibc, glibc-common)

The `glibc` and `glibc-common` RPMs should be installed with care. An incomplete C library upgrade can cause major upset. Whilst most currently active programs may continue normally they will be in an unpredictable state.

To minimise the chances of something going terribly wrong do not upgrade other RPMs in the same transaction. For example, the following command will download and upgrade the glibc, glibc-common and tuxracer RPMs:

```
apt-get install tuxracer glibc glibc-common
```

If for some reason, the tuxracer rpm fails in an unpredicted manner after the

```
rpm -e ...
```

stage, then you'll be left with glibc and glibc-common no longer installed.

> **Note:** The likelyhood of a necessary tuxracer upgrade is quite low.

After a glibc upgrade you should restart currently running programs as they will still be using the old libraries. You can see which processes will need restarting like this:

```
fuser -v /lib/i686/libc-*.so
```

It is particularly important that you restart `sshd` as we have experienced login problems after glibc upgrades. See FIXME: "safely upgrading and restarting sshd".

Recovering from a Borked C Library Upgrade

If something goes wrong upgrading these RPMs and libraries are missing, no dynamically linked programs will be able to execute until things are fixed, and fixing problems like these isn't a fun process. Remember, you can't start any new programs unless they are statically linked, so no bash, no rpm tools, no vi, and suprisingly not even cat or echo! (talk about bloat!) Whilst some currently active programs may continue normally they will be in an unpredictable state.

1. Boot disk;
2. mount hard disk;
3. rpm -r /mnt/harddisk/ -Uvh RedHat/RPMS/glibc*.rpm;
4. sync;
5. reboot.

Chapter 3. Configuring VPNs

IPSEC

Restarting IPSEC

```
ipsec auto --replace conn_name
ipsec auto --add whatever
ipsec auto --rereadsecrets  - make pluto re-read secrets
```

Services over VPN

Novell Netware

Logging In

From the Novell Knowledge-base[1]: "The best way to log into the server you want via IP only is to right click on the red N in the System Tray and select "NetWare Login. . . .". Then hit the advanced tab and enter the IP address of the server you want to log into in the "Server:" field. As long as your user has rights, and the server you want to log into has its routing path correct, you should be able to log into that server."

Notes

1. http://support.novell.com/cgi-bin/search/searchtid.cgi?/10057455.htm

Chapter 4. Configuring PPP Dial-in

For a more comprehensive PPP-HowTo see: tldp.org[1]

Identifying the Modem port

External

Once logged in to the client machine, you should `grep` through`/var/log/messages` to search for a `tty` string. If you find something similar to this:

```
ttyS00 at 0x03f8 (irq = 4) is a 16550A
```

This means there is one com port, and the mode will be attached to `/dev/ttyS0`

> **Note:** If there are more than one com ports, you will have to find out which com port the modem is attached to (com 0, 1, 2). This number will be similar to the device in `/dev`, id est: `com 0` = `/dev/ttyS0`; `com 1` = `/dev/ttyS1`

Internal

With an internal modem, you may find the correct modem port from output of `lspci -vv`.

```
02:09.0 Communication controller: Lucent Microelectronics Venus Modem (V90,56KFlex)
        Subsystem: Action Tec Electronics Inc: Unknown device 0480
        Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr-
Stepping- SERR+ FastB2B-
        Status: Cap+ 66Mhz- UDF- FastB2B+ ParErr- DEVSEL=medium TAbort-
TAbort- MAbort- SERR- PERR-
        Latency: 0 (63000ns min, 3500ns max)
        Interrupt: pin A routed to IRQ 21
        Region 0: Memory at f9fffc00 (32-bit, non-prefetchable) [size=256]
        Region 1: I/O ports at dc00 [size=256]
        Region 2: I/O ports at d800 [size=256]
        Region 3: I/O ports at d4f8 [size=8]
        Capabilities: [f8] Power Management version 2
                Flags: PMEClk- DSI+ D1- D2+ AuxCurrent=0mA PME(D0-,D1-
,D2+,D3hot+,D3cold-)
                Status: D0 PME-Enable- DSel=0 DScale=0 PME-
```

You will notice the `IRQ 21` line. You now need to iterate through `/dev/ttyS*` devices with setserial until you find a matching IRQ, as seen below:

```
# setserial /dev/ttyS4
/dev/ttyS4, UART: 16550A, Port: 0xdc00, IRQ: 21
```

Naming Convention

From here on I will refer to the modem port as `ttyS0` for reference. You should replace this with the port found by the above investigation.

Inittab

The /etc/inittab file needs to have an extra 'tty' added for the modem so that dial-in access will be initialised upon a reboot.

There will be several lines in this file for the standard getty's, which will look like this:

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

You must add the line 7:2345:respawn:/sbin/mgetty ttyS0 to the end of this section, so the section looks like this:

```
6:2345:respawn:/sbin/mingetty tty6
7:2345:respawn:/sbin/mgetty ttyS0
```

You will then need to restart init by executing the command: kill -HUP 1.

PPP Configuration

There are three configuration files that will need to be edited to set up the PPP connection.

Options

The /etc/ppp/options file contains the dial-in configuration.

This file needs to contain the following information:

```
auth +chap -pap crtscts proxyarp lock
ms-dns 123.456.789.123
name dialin
10.254.100.1:10.254.100.1
```

You will need to change the ms-dns 123.456.789.123 IP address to point to the correct DNS server (normally the IP address of eth0, if a DNS server is running on the host).

You may also need to change the 10.254.100.1:10.254.100.1 address range to ensure you do not have ssh host file conflicts when dialing in.

Chap Secrets

The Chap Secrets file contains our username and password. You will need the program genpass, to create a cryptographically secure password.

This file should contain information like below:

```
"pppusername" "dialin" "123456" "10.254.100.2"
```

You will need to change the "pppusername" and "123456" sections, to provide a good username (normally "companynameppp") and a secure password. You may also want to change the "10.254.100.2" IP address to use a different IP address, but this is not required by default.

Please note the following characters cannot be used in ppp passwords:

- ’ ’ - space
- ’ ’ - tab
- ’#’ - pound sign
- ’@’ - at sign
- ’\’ - backslash (next character is interpreted as function ie: \n = newline)
- ’"’ - double quote

ttyS0 Options

The `/etc/ppp/options.ttyS0` file will contain the IP address range which the host will use as local PPP addresses. This file will need to look like this:

```
10.254.100.1:10.254.100.1
```

You may want to change the address range if this conflicts with an address already in use, or if you require something different for another reason. However, the default is commonly sufficient.

Mgetty Configuration

The file `/etc/mgetty+sendfax/login.config` will need to contain the following information:

```
*       -       -       /usr/sbin/pppd
```

Firewall Rules

Firewall Interfaces

The `/etc/fw.if.conf` will need to contain an interface group for the ppp device. Add the following line to this file:

```
interface group modem has ppp0
```

Firewall rules

The `/etc/firewall.conf` will need to contain a rule allowing SSH access over the PPP connection. Add the following line to an appropriate section of this file:

```
accept input dport tcp.ssh from modem:@10.254.100.2
```

If you changed the leased IP address in `/etc/mgetty+sendfax/chap-secrets`, you will need to replace the `modem:10.254.100.2` address with the correct ip address.

You will then need to restart the firewall by issuing the command `service firewall restart`.

Dialing in

To test dial-in access you will need a preconfigured modem working on your machine.

WvDial Configuration

You will need to add the PPP dialin information to `/etc/wvdial.conf`. This file should already contain some information similar to:

```
[Modem0]
Modem = /dev/ttyS1
Baud = 57600
SetVolume = 0
Dial Command = ATDT
Init1 = ATZ
FlowControl = NOFLOW
```

The following information needs to be added:

```
[Dialer companynameppp]
Area Code = 0123
Username = companyppp
Password = 123456
Phone = 123456
Dial Prefix = 9
Inherits = Modem0
```

You will need to set the `Username` and `Password` values to those defined in the server config.

The `Dialer companynameppp`, `Area Code` and `Phone` lines will also need to be configured to reflect the company name and the phone number to dial in to the modem.

Dialing

To dial-in to the server using PPP, you will need to issue the command `wvdial companyppp`. You will then see a screen similar to this:

```
[root@ecsc root]# wvdial companyppp
--> WvDial: Internet dialer version 1.41
--> Initializing modem.
--> Sending: ATZ
ATZ
OK
--> Modem initialized.
--> Sending: ATDT 9,01274736223
--> Waiting for carrier.
ATDT 9,01274736223
CONNECT 21600/LAP-M
--> Carrier detected.  Starting PPP immediately.
--> Starting pppd
```

Once connected, a ppp0 interface will be shown if you issue the command `ifconfig`. This will look similar to this:

```
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.254.100.2  P-t-P:10.254.100.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:2 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:72 (72.0 b)  TX bytes:78 (78.0 b)
```

The `P-t-P:10.254.100.1` line shows the address you should connect to by SSH'ing to `root@10.254.100.1`

Troubleshooting

Cannot open /dev/ttyS0

If when trying to dial-in to a server, you receive the error:

```
--> WvDial: Internet dialer version 1.41
--> Cannot open /dev/ttyS1: Input/output error
```

Then your modem is not set up, or is not configured correctly. You may be trying to use a PCMCIA modem without the PCMCIA cardmgr services running.

LCP timeout

If you cannot connect to a server when dialing in, check your `/var/log/messages` to see if there is any debugging information. If it contains information similar to below:

```
pppd 2.4.1 started by root, uid 0
Using interface ppp0
Connect: ppp0 <--> /dev/ttyS1
/etc/hotplug/net.agent: assuming ppp0 is already up
LCP: timeout sending Config-Requests
Connection terminated.
Receive serial link is not 8-bit clean:
Problem: all had bit 7 set to 0
Exit.
```

Or, on the server you see information similar to this in `/var/log/messages`:

```
pppd 2.4.1 started by root, uid 0
Using interface ppp0
Connect: ppp0 <--> /dev/pts/0
```

You need to add the word `local` to the `/etc/ppp/options` on the server so it looks like this:

```
auth +chap -pap crtscts proxyarp lock local
```

FIXME: Why is this?

Notes

1. http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/PPP-HOWTO.html

Chapter 5. Configuring Mail

Virtual Mail

This section covers the virtual mail system. The virtual mail system provides IMAP and POP3 and mostly involves qmail and vmailmgr. Other aspects of Qmail are covered elsewhere in this document.

Installation

RPMs Needed:

- qmail
- vmailmgr
- vmailmgr-daemon
- ucspi-unix
- ucspi-tcp
- vmmi

For IMAP support, these extra RPMs are needed:

- courier-imap
- vmailmgr-courier-imap

Configuring

Files which need to be configured from the default install:

- `/etc/vmailmgr/socket-file`
- `/var/qmail/control/defaultdomain`
- `/var/qmail/control/virtualdomains`
- `/var/qmail/control/rcpthosts`
- `/var/qmail/control/checkpassword`
- `/etc/httpd/conf/httpd.conf`
- `/etc/squid/squid.conf`

`/etc/vmailmgr/socket-file` should contain the full path to the unix-style socket which vmailmgrd listens on. The path should be set to `/tmp/.vmailmgrd`.

`/var/qmail/control/defaultdomain` and `/var/qmail/control/rcpthosts` should contain the client's mail domain(s). e.g.

```
example.co.uk
example.com
```

`/var/qmail/control/virtualdomains` should contain the client's mail domain(s), then the virtual mail user, separated by a colon. e.g.

```
example.co.uk:vmail
example.com:vmail
```

`/var/qmail/control/checkpassword` should contain the password checker for virtual mail:

```
checkvpw
```

As of apache-1.3.23-ECSC4, and web_filter-2.2.0-ECSC2, commented-out lines exist to enable the vmmi package to run on port 7000; and as of webfwadmin-1.6.1-ECSC7, the mail button on the webadmin will push the client to a SSL secured port 7000 for vmail configuration. Several files need to be checked and/or changed thusly.

`/etc/httpd/conf/httpd.conf` contains the apache webserver configuration. There are two sections commented out (from lines 134 -> 140, and 275 -> 291) containing Directory and VirtualHost directives to enable `/var/www/html/vmail` on port 7000 with SSL. You will need to uncomment this first block; changing the "allowed from" line to allow access to the management interface from the client's LAN (or specific IPs). i.e.

```
allow from 192.168.0.0/24
```

.

You will also need to uncomment the large second block at the end of the file and change the addresses on the "Listen" and "VirtualHost" lines to reflect the IP address of the administrative interface (normally the firewall's internal address on the client's LAN).

You will need to configure squid to allow SSL through to port 7000 using lines similar to the following, which should be added to `/etc/squid/squid.conf`

```
acl vmail_port port 7000
acl vmail_users src 192.168.1.0/255.255.255.0
http_access allow CONNECT vmail_users vmail_port
```

Running and Testing

A virtual mail administrator must be added and a password assigned to that account.

> **Note:** The user added must be identical to the virtual mailuser added to `/var/qmail/control/virtualdomains`

Perform the following commands to setup the user:

```
adduser vmail
passwd vmail
su vmail
vadduser postmaster
```

To start vmailmgr link the vmailmgrd directory to the svcscan monitored /service directory as follows:

```
cd /service
ln -s /var/vmailmgrd
```

Checking the output of

```
ps -aufx
```

should show vmailmgrd running under /service, and the output of

```
netstat -an
```

should show a listening socket on `/tmp/.vmailmgrd`.

To restart apache and squid after re-configuring the configuration files, issue the following commands.

```
service httpd restart
service squid restart
```

The final test is to open a browser and connect to the firewall using https on port 7000.

Adding IMAP Mail Support

To enable IMAP support in Vmailmgr you must change the courier-imap authorisation program and set it as the default authorisation program. First perform the following command:

```
ln -s ‘which authvmailmgr‘ /usr/lib/courier-imap/libexec/authlib/authvmailmgr
```

Then modify the ‘AUTHMODULES‘ statement in `/usr/lib/courier-imap/etc/imapd.config` and add

```
authvmailmgr
```

as the first authentication module.

Extra features

Auto append @domain.com to incoming mails (envnoathost)

To allow the sending of e-mail without specifying the domain, you need to tell qmail which domain to append when receiving such mails.

Put the domain name you want to use in the qmail control file `/var/qmail/control/envnoathost` and restart qmail

Chapter 6. Configuring the Apache Web Server

Secure Virtual Site Configuration

This section will help you add virtual web sites on Apache under Linux, in a secure manner.

Creating the Web Root

Create a new user

Create a new user for the web root, this example will use the user 'fred'. Lock his password using:

```
passwd -l fred
```

. su to the new user Fred, and in his home dir, create a directory for the web files to go in. For example: `/home/fred/www.testsite.com`

Permissions

Fred needs full access to his files, apache needs read access to the web root (and execute for directories of course), and everybody else needs diddly squat. NOTE: Apache will also need execute access for Fred's home dir as it tends to use stat to work out its path (apparently).

If you su'ed to Fred before creating the webroot, it should already belong to him and have the correct default permissions for himself. Set up the other permissions using the chmod and chgrp commands:

```
chgrp apache /home/fred/
chmod 0710 /home/fred
chgrp apache /home/fred/www.testsite.com
chmod 0750 /home/fred/www.testsite.com
```

You should also set the web root directory to be setgid, ensuring all files created in there will belong to apache, thus reducing the user's temptation to give the world read (and usually write) access to their valuable secret proprietary perl scripts.

```
chmod 2750 /home/fred/www.testsite.com
```

Creating the Web Logs Directory

Each web hosting user needs access to their own web access and error logs. This is a delicate process where permissions are concerned.

Creating the Directory

We now place logfiles in a seperate location, and symlink them into the user's home dir. We'll need to make a directory to put the logs in. For this example, we'll use `/var/log/httpd/wwwlogs/fred`.

Permissions

Incorrect log dir permissions can open Apache up to a denial of service attack, and as the logs are created and written to by an Apache process running as root, lots of other nasty attacks. The user should have only READ access to the directory:

```
chown root.fred /var/log/httpd/wwwlogs/fred
chmod 0750 /var/log/httpd/wwwlogs/fred
```

httpd.conf

Globally, things such as handlers, php, Aliases and UserDir should not be enabled. You should do this at the virtual host level. If Fred doesn't use php scripts then he doesn't need the

```
AddType application/x-httpd-php
.php4 .php3 .phtml .php
```

line covering him globally.

Configuring the VirtualHost

Set up the apache VirtualHost directive in the httpd.conf, for example:

```
<VirtualHost>
ServerAdmin fred@testsite.com
DocumentRoot /home/fred/www.testsite.com
ServerName www.testsite.com
ErrorLog /var/log/httpd/wwwlogs/fred/wwwerror.log
CustomLog /var/log/httpd/wwwlogs/fred/www.testsite.com.log combined
</VirtualHost>
```

Chapter 7. Configuring IIS Protection and SSL acceleration

Backend Services

Microsoft Outlook Web Access

The Microsoft Outlook Web Access system (OWA) can run directly from an Exchange server or through an IIS virtual host. Either way, there are a number of problems you may run into.

SSL Acceleration

SSL Accelerating an OWA service requires a little tweak with Apache. You'll need the specially patched mod_proxy and the following line in your apache configuration:

```
ProxyRequestHeader set Front-End-Https On
```

This sets a special HTTP header in the backend request to the Outlook Web host. Now the Outlook Web service knows there is an SSL accelerator in front of it and rewrites URLs with `https://` rather than `http://`.

The SEARCH http method and Squid

When accessing an OWA service, Internet Explorer makes use of a non-standard HTTP method named "SEARCH". You will need to set your squid caches to accept and allow this method else the the inbox will just show `Loading...` and never return (or complain, bleh).

You can do this with the following line in your squid config (Squid 2.4):

```
extension_methods request SEARCH
```

The Apache Limit* configuration

Apaches Limit* directives can cause problems with OWA as it tends to make large and sometimes unusual-looking requests. The ECSC Webfwadmin's default Apache settings are currently:

```
LimitRequestBody 2048
LimitRequestFieldsize 4096
LimitRequestLine 2048
LimitRequestFields 30
```

Requests generated by OWA can be as big as any e-mail sent by a user, so 2k is clearly not enough for all-singing all-dancing animated html ActiveX e-mails. You can experiement with these settings but it is currently recommended to comment them out.