



Channel Bonding Server Installation Manual

Document number : XXX-Axxxx
Issue : 1.0
Date : 01 July 2014
Filename : CHANNELBONDSEVERINSTALLATION_RevA.

NOTICE

This document is the property of Cobham. All rights of copyright are reserved by Cobham. This document must not be reproduced or used for any other purpose than that for which it was intended without written permission. Cobham reserves the right to change any information contained in this document without prior warning. It is the user's responsibility to ensure that the appropriate issue of this document is being used. While Cobham has taken every care in the preparation of this document, Cobham does not accept responsibility for loss or damage alleged to have been suffered arising out of errors or omissions. *CAGE Code SV658.*

CONTROLLED DOCUMENTATION

Notwithstanding the terms of the above Disclaimer, Cobham undertakes in the case of controlled documentation to supply the Customer with all updates or amendments to this document and furthermore undertakes to notify the Customer in advance of material changes to the information contained herein.

DOCUMENT CHANGE HISTORY

ISSUE	DATE	ECP	DESCRIPTION OF CHANGE
	01 Jul 2014		Installation manual for version 1.0 of the bond server

DOCUMENT APPROVAL AND VALIDATION

	NAME	SIGN	DATE
AUTHOR	Nick Spring		01 July 2014
APPROVED			
APPROVED QA			

TABLE OF CONTENTS

	PAGE
1. INSTALLATION	4
1.1 DISTRIBUTIONS	4
1.2 DEPENDENCIES	4
1.3 INSTALLATION PROCESS	4
2. CONFIGURATION.....	4
2.1 /ETC/CBSD.CONF	4
2.1.1 CONTROL_PORT	4
2.1.2 DATA_PORT	5
2.1.3 SUBNET_ID	5
2.2 /ETC/CBSD.D/TUN-UP	5
2.3 /ETC/PPP/CHAP-SECRETS	5
2.4 LINUX FIREWALL RULES.....	5
3. RUNNING.....	6

LIST OF FIGURES

PAGE

No table of figures entries found.

LIST OF TABLES

PAGE

No table of figures entries found.

1. INSTALLATION

1.1 Distributions

Version 1.0 of the bond server is a 32 bit application that has been installed and tested on the following Linux distributions:

- CentOS 6.5-i386

1.2 Dependencies

The bond server application requires the use of the OpenSSL cryptography library during client-server authentication. This library is installed, for example on CentOS 6.5 32 bit system as follows:

```
> sudo yum install openssl-devel
```

1.3 Installation process

Version 1.0 of the channel bonding server is distributed as a tarball, *bondserver.tar*. Once this file is copied onto the server machine, the bond server can be installed by unpacking the tarball as follows:

```
> sudo tar -Pxf bondserver.tar
```

All application, configuration and script files will be unpacked as follows

```
/etc/cbsd.conf  
/etc/cbsd.d/tun-up  
/etc/cbsd.d/tun-down  
/usr/bin/bondserver
```

2. CONFIGURATION

Configuring the bond server application requires editing of the main configuration file and potentially some of the application scripts. Following this, users are then added to the server.

2.1 /etc/cbsd.conf

cbsd.conf is the main application configuration file and can be manually edited using a text editor. This file contains text based configuration items which are formatted as follows:

```
PARAMETER:VALUE
```

Default values for all configuration items are specified in the installed file.

2.1.1 CONTROL_PORT

The CONTROL_PORT configuration item specifies on which TCP port the bond server application shall listen for incoming client control connections. The default value for this item is TCP port 9052. All firewalls protecting the bond server machine must be configured to allow for TCP connections to the application on this specified port. **NOTE: When configuring the server for bonding with Explorer 710 UTs, this value must be 9052.**

2.1.2 DATA_PORT

The DATA_PORT configuration item specifies on which UDP port the bond server application shall receive bonded IP traffic from the UT client. The default value for this item is UDP port 9053. All firewalls protecting the bond server machine must be configured to allow for UDP data to be received by the application on this specified port. **NOTE: When configuring the server for bonding with Explorer 710 UTs, this value must be 9053.**

2.1.3 SUBNET_ID

The bond server application requires a private IP subnet for internal use. This subnet ID must be configured such that it does not conflict with any existing private subnets already in use on the bond server machine as well as the UT equipment. SUBNET_ID is defaulted to 10.0.1.0 and must adhere to "RFC 1918 – Address allocation for private networks". Valid configuration values must be within the IP ranges:

10.0.0.0	-	10.255.255.255	(10/8 prefix)
172.16.0.0	-	172.31.255.255	(172.16/12 prefix)
192.168.0.0	-	192.168.255.255	(192.168/16 prefix)

2.2 /etc/cbsd.d/tun-up

At start up, the bond server will execute the tun-up script to configure an *iptables* masquerading rule. If required, the location of this rule in *iptables* NAT table can be changed by editing this script.

2.3 /etc/ppp/chap-secrets

The bond server application implements a CHAP style authentication scheme. All client username and password credentials must be configured in *chap-secrets*. A sample file would look like:

```
# Secrets for authentication using CHAP
# client      server      secret      IP addresses
user1         *           password1   *
user2         *           password2   *
```

With a wild card (*) value entered for *server* and *IP addresses* fields.

2.4 Linux Firewall rules

Note: The purpose of this section of the document is to inform the installer of the bond server application what minimum firewall configuration is required to allow for channel bonding operation and **NOT** how to securely configure the firewall itself.

The following set of example rules use eth0 as the WAN interface name. These example rules are a starting point for creating a rule that meets security requirements as well as channel bonding requirements.

```
# Allow inbound TCP connections on CONTROL_PORT to the bond server
> iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 9052 -i eth0 -j ACCEPT
```

```
# Allow inbound bonded UDP data to be received on the bond server on port DATA_PORT
> iptables -I INPUT 1 -p udp -m udp --dport 9053 -i eth0 -j ACCEPT
```

```
#Allow outbound bonded packets
> iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -o eth0 -j ACCEPT
```

```
# Allow the bond server NAT to operate correctly
> iptables -I FORWARD 1 -m state --state ESTABLISHED,RELATED -i eth0 -j ACCEPT

# Allow the bond server NAT to operate correctly
> iptables -I FORWARD 1 -m state --state NEW,ESTABLISHED,RELATED -o eth0 -j ACCEPT
```

3. RUNNING

The bond server application must be run as *sudo* and the name of the WAN interface network adaptor specified (-w), for example

```
> sudo bondserver -w eth0
```

The application, by default will run as a daemon, however if required, the application can be run in console mode (-c).

The syslog can be grepped for *BondServer* to extract the bond server application logs.