

FreeBSD Administration - Support #541

Install Snort, Barnyard2, PulledPork, and Snorby With Nginx on FreeBSD

01/24/2015 05:04 PM - Daniel Curtis

Status:	Closed	Start date:	01/24/2015
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	100%
Category:	Intrusion Detection/Prevention	Estimated time:	1.50 hour
Target version:	FreeBSD 9	Spent time:	14.00 hours

Description

Like many paranoid network & system administrators, I have need of more than just an antivirus and firewall on each networked device. So I have decided to install a Snort machine and log the information to a remote MariaDB server. This is a simple guide to set up a Snort machine on a FreeBSD 9.2 system.

Prepare the system

- Update the system

```
pkg update && pkg upgrade  
portsnap fetch extract
```

- Install portmaster:

```
cd /usr/ports/ports-mgmt/portmaster  
make install clean  
pkg2ng
```

Install Snort

- Install Snort

```
portmaster security/snort security/barnyard2 security/pulledpork
```

NOTE: Enable [X]MYSQL during the config of security/barnyard2

- Create the following directories:

```
mkdir -p /usr/local/etc/snort/so_rules  
mkdir -p /usr/local/etc/snort/rules/iplists  
mkdir -p /var/log/barnyard2
```

- Then create a few blank files:

```
touch /usr/local/etc/snort/rules/snort.rules  
touch /usr/local/etc/snort/rules/local.rules  
touch /usr/local/etc/snort/rules/white_list.rules  
touch /usr/local/etc/snort/rules/black_list.rules  
touch /var/log/snort/barnyard2.waldo
```

Configure Snort

- Edit the snort config file:

```
vi /usr/local/etc/snort/snort.conf
```

- And modify the following parameters:

```
ipvar HOME_NET 192.168.1.0/24
ipvar EXTERNAL_NET any

var RULE_PATH /usr/local/etc/snort/rules
var SO_RULE_PATH /usr/local/etc/snort/so_rules
var PREPROC_RULE_PATH /usr/local/etc/snort/preproc_rules
var WHITE_LIST_PATH /usr/local/etc/snort/rules
var BLACK_LIST_PATH /usr/local/etc/snort/rules

dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/
dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so
#dynamicdetection directory /usr/local/lib/snort/dynamicrules

output unified2: filename snortunified2.log, limit 128

## Comment out every $RULE_PATH line
#include $RULE_PATH

## Add definition for aggregate snort.rules file
include $RULE_PATH/snort.rules
```

- (Optional) Remove all commented lines from snort config:

```
grep '^#[^#]' /usr/local/etc/snort/snort.conf > /usr/local/etc/snort/temp.conf
mv -f /usr/local/etc/snort/temp.conf /usr/local/etc/snort/snort.conf
```

Configure Pulledpork

- Create and edit a Pulledpork config file:

```
cp /usr/local/etc/pulledpork/pulledpork.conf.sample /usr/local/etc/pulledpork/pulledpork.conf
vi /usr/local/etc/pulledpork/pulledpork.conf
```

- And modify the following, making sure to replace <oinkcode> with your actual oinkcode.

```
rule_url=https://www.snort.org/reg-rules/|snortrules-snapshot.tar.gz|<oinkcode>

rule_url=https://s3.amazonaws.com/snort-org/www/rules/community/|community-rules.tar.gz|Community

rule_url=http://labs.snort.org/feeds/ip-filter.blf|IPBLACKLIST|open

rule_url=https://www.snort.org/reg-rules/|opensource.gz|<oinkcode>

rule_url=https://rules.emergingthreatspro.com/|emerging.rules.tar.gz|open

rule_path=/usr/local/etc/snort/rules/snort.rules

out_path=/usr/local/etc/snort/rules/

so_rule_path=/usr/local/etc/snort/so_rules/

distro=FreeBSD-9.0
```

```
black_list=/usr/local/etc/snort/rules/iplists/default.blacklist
```

- Update Snort rules using Pulledpork:

```
pulledpork.pl -c /usr/local/etc/pulledpork/pulledpork.conf
```

- If you have such an error while issuing the command with the -vv parameter:

```
500 Can't connect to www.snort.org:443 (Crypt-SSLeay can't verify hostnames
```

- Then add this environment variable:

```
bash
export HTTPS_CA_DIR=/usr/share/ca-certificates/
pulledpork.pl -c /usr/local/etc/pulledpork/pulledpork.conf
```

- And add following line to /etc/crontab (the example automatically checks for the presence of new rules every 12 hours):

```
echo '## Update Snort rules' >> /etc/crontab
echo '5 */12 * * * /usr/bin/perl /usr/local/bin/pulledpork.pl -c /usr/local/etc/pulledpork/pulledpork.conf' >> /etc/crontab
```

- Restart cron:

```
service cron restart
```

Configure Barnyard2

- Edit the barnyard2 config file:

```
vi /usr/local/etc/barnyard2.conf
```

- And modify the following:

```
config hostname: snort.example.com
output database: log, mysql, user=snorby password=SuperSecretPassword dbname=snorby host=localhost
```

Install Snorby

Snorby is a web frontend for the Snort IDS, and this is a simple guide on installing it on FreeBSD 9.2. This guide only sets up Snorby, as my setup has the Snort agent on remote machine, sending its data to a different remote database.

- Install a few prerequisite packages:

```
portmaster shells/bash ftp/wget textproc/flex devel/pcre net/libdnet textproc/libxml2 textproc/libxslt graphics/ImageMagick devel/lwp www/p5-LWP-UserAgent-WithCache security/p5-Crypt-SSLeay www/p5-LWP-Protocol-https lang/ruby21 devel/ruby-gems converters/wkhtmltopdf devel/readline
```

- Fix Bash:

```
ln -s /usr/local/bin/bash /bin/bash
```

NOTE: This is required later by snorby, an error will occur otherwise.

- Install some prerequisite gems:

```
portmaster print/rubygem-prawn devel/rubygem-thor devel/rubygem-i18n sysutils/rubygem-bundler  
devel/rubygem-tzinfo devel/rubygem-builder databases/rubygem-memcache-client www/rubygem-rack  
www/rubygem-rack-test www/rubygem-erubis mail/rubygem-mail textproc/rubygem-text databases/ruby  
gem-sqlite3 devel/rubygem-rake databases/rubygem-mysql www/rubygem-rack-mount www/rubygem-ra  
ils
```

- Now create a snorby user:

```
pw add user -n snorby -d /usr/local/www/snorby -m -s /usr/local/bin/bash -c "Snorby"
```

- Get Snorby from the download section or use the latest edge release via git.

```
cd /usr/local/www  
git clone git://github.com/Snorby/snorby.git
```

- Install RVM:

```
su - snorby  
curl -L https://get.rvm.io | bash  
source /usr/local/www/snorby/.rvm/scripts/rvm
```

- Install Ruby 1.9.3

```
rvm install 1.9.3  
rvm use 1.9.3
```

- Install Passenger inside the RVM environment:

```
gem install passenger
```

- Install bundler inside the RVM environment:

```
gem install bundler
```

- Create a database config file:

```
cp config/database.example.yml config/database.yml
```

- Change the database, host, user, and password accordingly

- Create and edit the Snorby config:

```
cp config/snорby_config.yml.example config/snорby_config.yml  
vi config/snорby_config.yml
```

- And add or modify the following

```
# Change the production configuration for your environment.  
production:  
domain: snорby.example.com  
wkhtmltopdf: /usr/local/bin/wkhtmltopdf  
mailer_sender: 'snорby@snорby.org'  
geoip_uri: "http://geolite.maxmind.com/download/geoip/database/GeoLiteCountry/GeoIP.dat.gz"  
"  
rules:  
- "/usr/local/etc/snort/rules"  
authentication_mode: database
```

- Install Gem Dependencies

```
RAILS_ENV=production bundle install --path vendor/bundle
```

- Install the railties gem using the system libraries:

```
gem install railties -- --use-system-libraries
```

- Run the Snorby Setup

```
RAILS_ENV=production bundle exec rake snорby:setup
```

- Restart the snорby worker:

```
RAILS_ENV=production bundle exec rails r Snорby::Worker.stop  
RAILS_ENV=production bundle exec rails r Snорby::Worker.start
```

- Exit the snорby user environment:

```
exit
```

Snorbyfix Script

- Create the snorbyfix script:

```
vi /usr/local/bin/snорbyfix.sh
```

- And add the following:

```
#!/bin/sh  
# Snorby Worker script  
su - snорby -c 'RAILS_ENV=production rails r Snорby::Worker.restart'
```

- Create a cronjob to run the snorbyfix script every hour:

```
echo '## Fix snorby worker' >> /etc/crontab
echo '* * * * * snorby /usr/local/bin/snorbyfix.sh' >> /etc/crontab
```

- Make the script executable:

```
chmod +x /usr/local/bin/snorbyfix.sh
```

- Restart the cron service:

```
service cron restart
```

Install Nginx

- Install Nginx with Passenger

```
portmaster www/nginx
```

NOTE: Make sure to enable [X]PASSENGER when running make config

- Install the Passenger gem:

```
portmaster www/rubygem-passenger
```

NOTE: Make sure to enable (*) NGINX when running make config

Configure Nginx

- Create a configuration directory to make managing individual server blocks easier:

```
mkdir /usr/local/etc/nginx/conf.d
```

- Configuring Nginx and Passenger, edit the /usr/local/etc/nginx/nginx.conf file:

```
vi /usr/local/etc/nginx/nginx.conf
```

- And add/modify the following

```
user    www    www;
worker_processes  4;
error_log  /var/log/nginx/error.log notice;
pid      /var/run/nginx.pid;

events {
    worker_connections  1024;
}

http {
    passenger_root /usr/local/lib/ruby/gems/2.0/gems/passenger-4.0.58;
    passenger_ruby /usr/local/bin/ruby;
    passenger_max_pool_size 15;
    passenger_pool_idle_time 300;
    #passenger_spawn_method direct; # Uncomment on Ruby 1.8 for ENC to work

    include mime.types;
```

```

default_type application/octet-stream;
sendfile on;
tcp_nopush on;
keepalive_timeout 65;
tcp_nodelay on;

# Load config files from the /etc/nginx/conf.d directory
include /usr/local/etc/nginx/conf.d/*.conf;
}

```

NOTE: The above configuration will set the ruby used by passenger to the system default ruby.

- And add a default site configuration in /usr/local/etc/nginx/conf.d/default.conf:

```

server {
    listen 80 default;
    server_name _;

    index index.html index.php;
    root /usr/local/www;

    # IP and IP ranges which should get access
    allow 10.0.0.0/24;
    allow 10.1.0.1;
    # all else will be denied
    deny all;

    # basic HTTP auth
    auth_basic "Restricted";
    auth_basic_user_file htpasswd;

    location ~ \.cgi$ {
        try_files $uri =404;
        include fastcgi_params;
        fastcgi_pass unix:/var/run/fcgiwrap/fcgiwrap.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param REMOTE_USER $remote_user;
    }

    location ~ \.php$ {
        try_files $uri =404;
        include fastcgi_params;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}

```

- Find the exact ruby version that snorby will use:

```

su - snorby
passenger-config --ruby-command

```

- Example output:

```

passenger-config was invoked through the following Ruby interpreter:
  Command: /usr/local/www/snorby/.rvm/gems/ruby-1.9.3-p551/wrappers/ruby
  Version: ruby 1.9.3p551 (2014-11-13 revision 48407) [x86_64-freebsd9.3]
  To use in Apache: PassengerRuby /usr/local/www/snorby/.rvm/gems/ruby-1.9.3-p551/wrappers/ruby
  To use in Nginx : passenger_ruby /usr/local/www/snorby/.rvm/gems/ruby-1.9.3-p551/wrappers/ruby
  To use with Standalone: /usr/local/www/snorby/.rvm/gems/ruby-1.9.3-p551/wrappers/ruby /usr/local/www/snorby/.rvm/gems/ruby-1.9.3-p551/gems/passenger-4.0.58/bin/passenger start

```

- And create a server block for snorby

```
vi /usr/local/etc/nginx/conf.d/snorby.conf
```

- And add the following:

```
server {  
    listen      80;  
    server_name snorby.example.com;  
  
    passenger_enabled on;  
    passenger_ruby /usr/local/www/snorby/.rvm/gems/ruby-1.9.3-p551/wrappers/ruby;  
    passenger_user      snorby;  
    passenger_group     snorby;  
  
    access_log /var/log/nginx/snorby.log;  
    root /usr/local/www/snorby/public;  
}
```

- Create the log directory to prevent issues on startup:

```
mkdir /var/log/nginx
```

- Restart nginx

```
service nginx restart
```

Log into Snorby

- The server name set up in this example is <http://snorby.example.com>
- The default username is snorby@snorby.org
- The default password is **snorby**

Resources

- <https://github.com/Snorby/snorby>
- <https://github.com/shirkdog/snorby-bsd/blob/master/snorbyInstall.sh>

History

#1 - 01/24/2015 05:11 PM - Daniel Curtis

- Description updated

#2 - 01/24/2015 05:32 PM - Daniel Curtis

- Description updated

#3 - 01/24/2015 07:10 PM - Daniel Curtis

- Description updated

#4 - 01/27/2015 02:25 PM - Daniel Curtis

- Description updated

#5 - 01/27/2015 02:28 PM - Daniel Curtis

Since my database on a remote host, I needed to make sure the barnyard2 mysql schema was created:

```
# Copyright (C) 2000-2002 Carnegie Mellon University
#
# Maintainer: Roman Danyliw <rdd@cert.org>, <roman@danyliw.com>
#
# Original Author(s): Jed Pickel <jed@pickel.net> (2000-2001)
# Roman Danyliw <rdd@cert.org>
# Todd Schrubb <tls@cert.org>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License Version 2 as
# published by the Free Software Foundation. You may not use, modify or
# distribute this program under any other version of the GNU General
# Public License.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
CREATE TABLE `schema` ( vseq INT UNSIGNED NOT NULL,
ctime DATETIME NOT NULL,
PRIMARY KEY (vseq));
INSERT INTO `schema` (vseq, ctime) VALUES ('107', now());
CREATE TABLE event ( sid INT UNSIGNED NOT NULL,
cid INT UNSIGNED NOT NULL,
signature INT UNSIGNED NOT NULL,
timestamp DATETIME NOT NULL,
PRIMARY KEY (sid,cid),
INDEX sig (signature),
INDEX time (timestamp));
CREATE TABLE signature ( sig_id INT UNSIGNED NOT NULL AUTO_INCREMENT,
sig_name VARCHAR(255) NOT NULL,
sig_class_id INT UNSIGNED NOT NULL,
sig_priority INT UNSIGNED,
sig_rev INT UNSIGNED,
sig_sid INT UNSIGNED,
sig_gid INT UNSIGNED,
PRIMARY KEY (sig_id),
INDEX sign_idx (sig_name(20)),
INDEX sig_class_id_idx (sig_class_id));
CREATE TABLE sig_reference (sig_id INT UNSIGNED NOT NULL,
ref_seq INT UNSIGNED NOT NULL,
ref_id INT UNSIGNED NOT NULL,
PRIMARY KEY(sig_id, ref_seq));
CREATE TABLE reference ( ref_id INT UNSIGNED NOT NULL AUTO_INCREMENT,
ref_system_id INT UNSIGNED NOT NULL,
ref_tag TEXT NOT NULL,
PRIMARY KEY (ref_id));
CREATE TABLE reference_system ( ref_system_id INT UNSIGNED NOT NULL AUTO_INCREMENT,
ref_system_name VARCHAR(20),
PRIMARY KEY (ref_system_id));
CREATE TABLE sig_class ( sig_class_id INT UNSIGNED NOT NULL AUTO_INCREMENT,
sig_class_name VARCHAR(60) NOT NULL,
PRIMARY KEY (sig_class_id),
INDEX (sig_class_id),
INDEX (sig_class_name));
# store info about the sensor supplying data
CREATE TABLE sensor ( sid INT UNSIGNED NOT NULL AUTO_INCREMENT,
hostname TEXT,
interface TEXT,
filter TEXT,
detail TINYINT,
encoding TINYINT,
last_cid INT UNSIGNED NOT NULL,
PRIMARY KEY (sid));
# All of the fields of an ip header
CREATE TABLE iphdr ( sid INT UNSIGNED NOT NULL,
cid INT UNSIGNED NOT NULL,
ip_src INT UNSIGNED NOT NULL,
```

```

ip_dst INT UNSIGNED NOT NULL,
ip_ver TINYINT UNSIGNED,
ip_hlen TINYINT UNSIGNED,
ip_tos TINYINT UNSIGNED,
ip_len SMALLINT UNSIGNED,
ip_id SMALLINT UNSIGNED,
ip_flags TINYINT UNSIGNED,
ip_off SMALLINT UNSIGNED,
ip_ttl TINYINT UNSIGNED,
ip_proto TINYINT UNSIGNED NOT NULL,
ip_csum SMALLINT UNSIGNED,
PRIMARY KEY (sid,cid),
INDEX ip_src (ip_src),
INDEX ip_dst (ip_dst));
# All of the fields of a tcp header
CREATE TABLE tcphdr( sid INT UNSIGNED NOT NULL,
cid INT UNSIGNED NOT NULL,
tcp_sport SMALLINT UNSIGNED NOT NULL,
tcp_dport SMALLINT UNSIGNED NOT NULL,
tcp_seq INT UNSIGNED,
tcp_ack INT UNSIGNED,
tcp_off TINYINT UNSIGNED,
tcp_res TINYINT UNSIGNED,
tcp_flags TINYINT UNSIGNED NOT NULL,
tcp_win SMALLINT UNSIGNED,
tcp_csum SMALLINT UNSIGNED,
tcp_urp SMALLINT UNSIGNED,
PRIMARY KEY (sid,cid),
INDEX tcp_sport (tcp_sport),
INDEX tcp_dport (tcp_dport),
INDEX tcp_flags (tcp_flags));
# All of the fields of a udp header
CREATE TABLE udphdr( sid INT UNSIGNED NOT NULL,
cid INT UNSIGNED NOT NULL,
udp_sport SMALLINT UNSIGNED NOT NULL,
udp_dport SMALLINT UNSIGNED NOT NULL,
udp_len SMALLINT UNSIGNED,
udp_csum SMALLINT UNSIGNED,
PRIMARY KEY (sid,cid),
INDEX udp_sport (udp_sport),
INDEX udp_dport (udp_dport));
# All of the fields of an icmp header
CREATE TABLE icmphdr( sid INT UNSIGNED NOT NULL,
cid INT UNSIGNED NOT NULL,
icmp_type TINYINT UNSIGNED NOT NULL,
icmp_code TINYINT UNSIGNED NOT NULL,
icmp_csum SMALLINT UNSIGNED,
icmp_id SMALLINT UNSIGNED,
icmp_seq SMALLINT UNSIGNED,
PRIMARY KEY (sid,cid),
INDEX icmp_type (icmp_type));
# Protocol options
CREATE TABLE opt ( sid INT UNSIGNED NOT NULL,
cid INT UNSIGNED NOT NULL,
optid INT UNSIGNED NOT NULL,
opt_proto TINYINT UNSIGNED NOT NULL,
opt_code TINYINT UNSIGNED NOT NULL,
opt_len SMALLINT,
opt_data TEXT,
PRIMARY KEY (sid,cid,optid));
# Packet payload
CREATE TABLE data ( sid INT UNSIGNED NOT NULL,
cid INT UNSIGNED NOT NULL,
data_payload TEXT,
PRIMARY KEY (sid,cid));
# encoding is a lookup table for storing encoding types
CREATE TABLE encoding(encoding_type TINYINT UNSIGNED NOT NULL,
encoding_text TEXT NOT NULL,
PRIMARY KEY (encoding_type));
INSERT INTO encoding (encoding_type, encoding_text) VALUES (0, 'hex');
INSERT INTO encoding (encoding_type, encoding_text) VALUES (1, 'base64');
INSERT INTO encoding (encoding_type, encoding_text) VALUES (2, 'ascii');
# detail is a lookup table for storing different detail levels
CREATE TABLE detail (detail_type TINYINT UNSIGNED NOT NULL,
detail_text TEXT NOT NULL,

```

```
PRIMARY KEY (detail_type));  
INSERT INTO detail (detail_type, detail_text) VALUES (0, 'fast');  
INSERT INTO detail (detail_type, detail_text) VALUES (1, 'full');  
# be sure to also use the snortdb-extra tables if you want  
# mappings for tcp flags, protocols, and ports
```

Resources

- https://github.com/firnsy/barnyard2/blob/master/schemas/create_mysql

#6 - 01/27/2015 02:52 PM - Daniel Curtis

- *Description updated*
- *Status changed from New to In Progress*

#7 - 01/27/2015 04:22 PM - Daniel Curtis

- *Description updated*

#8 - 01/27/2015 10:05 PM - Daniel Curtis

- *Subject changed from Installing Snorby With Nginx on FreeBSD to Installing Snort, Barnyard2, PulledPork, and Snorby With Nginx on FreeBSD*
- *Description updated*

#9 - 01/31/2015 10:36 AM - Daniel Curtis

- *Description updated*
- *% Done changed from 0 to 50*

#10 - 01/31/2015 02:48 PM - Daniel Curtis

- *Description updated*

#11 - 01/31/2015 09:33 PM - Daniel Curtis

- *Description updated*

#12 - 02/01/2015 07:33 PM - Daniel Curtis

- *Description updated*
- *% Done changed from 50 to 90*

#13 - 02/02/2015 09:03 PM - Daniel Curtis

- *Description updated*

#14 - 02/02/2015 09:04 PM - Daniel Curtis

- *Description updated*

#15 - 02/02/2015 09:16 PM - Daniel Curtis

- *Description updated*

#16 - 02/02/2015 09:30 PM - Daniel Curtis

- *Description updated*
- *Status changed from In Progress to Resolved*
- *% Done changed from 90 to 100*

#17 - 02/03/2015 02:59 PM - Daniel Curtis

- *Description updated*

#18 - 02/03/2015 03:01 PM - Daniel Curtis

- *Description updated*

#19 - 02/03/2015 03:26 PM - Daniel Curtis

- *Description updated*

#20 - 02/03/2015 04:17 PM - Daniel Curtis

- *Description updated*

#21 - 02/04/2015 04:01 PM - Daniel Curtis

- *Description updated*

#22 - 02/06/2015 04:05 PM - Daniel Curtis

- *Description updated*

#23 - 02/06/2015 04:22 PM - Daniel Curtis

- *Description updated*

#24 - 02/07/2015 11:30 AM - Daniel Curtis

- *Description updated*

#25 - 02/07/2015 11:56 AM - Daniel Curtis

- *Description updated*

#26 - 02/08/2015 11:56 AM - Daniel Curtis

- *Description updated*

#27 - 02/10/2015 11:25 AM - Daniel Curtis

- *Description updated*

#28 - 02/10/2015 11:28 AM - Daniel Curtis

- *Description updated*

#29 - 02/14/2015 10:47 AM - Daniel Curtis

- *Target version set to FreeBSD 9*

#30 - 02/14/2015 12:04 PM - Daniel Curtis

- *Category set to Intrusion Detection/Prevention*

#31 - 02/22/2015 05:21 PM - Daniel Curtis

- *Status changed from Resolved to Closed*

#32 - 02/25/2015 04:58 PM - Daniel Curtis

- *Description updated*

#33 - 03/25/2015 09:01 PM - Daniel Curtis

- *Subject changed from Installing Snort, Barnyard2, PulledPork, and Snorby With Nginx on FreeBSD to Install Snort, Barnyard2, PulledPork, and Snorby With Nginx on FreeBSD*

- *Description updated*

- *Status changed from Closed to Resolved*

#34 - 03/26/2015 08:21 AM - Daniel Curtis

- *Description updated*

#35 - 03/26/2015 08:24 AM - Daniel Curtis

- *Description updated*

#36 - 03/31/2015 09:32 PM - Daniel Curtis

- *Description updated*

#37 - 04/05/2015 07:55 PM - Daniel Curtis

- *Description updated*

#38 - 04/05/2015 08:02 PM - Daniel Curtis

- *Description updated*

#39 - 04/06/2015 11:59 AM - Daniel Curtis

- *Description updated*

#40 - 04/06/2015 02:33 PM - Daniel Curtis

- *Description updated*

#41 - 04/09/2015 03:41 PM - Daniel Curtis

- *Description updated*

#42 - 04/09/2015 04:08 PM - Daniel Curtis

- *Description updated*

#43 - 04/09/2015 04:16 PM - Daniel Curtis

- *Description updated*

#44 - 04/11/2015 01:16 PM - Daniel Curtis

- *Status changed from Resolved to Closed*