UUCP over SSL HOWTO

Tuc <tuc@ttsg.com> $Revision: 1.9 $, $Date: 2003/07/18 16:29:54 $

This document describes how to set up UUCP (master and slave) to work over SSL (Secure Socket Layer) using taylor-UUCP and stunnel.

Contents

1 Introduction

Almost all mail transfer protocols have been enctrypted over the last years. One has to be really lazy to use POP3 or IMAP over the internet and thus handling over their login/password to the public.

UUCP seems not to be a state of the art solution for mail-transfer, but there are still a bunch of users, who mostly do UUCP over TCP - which of course uses plaintext passwords. I'm the administrator of a machine with over 300 UUCP accounts and I really want to provide my users the ability to transfer their mail and news batches securely.

2 Software needed

As my test environment consists out of GNU/Linux boxes, there is no guarantee that this may work on other *NIX-like operating systems - although it should work on any platform which supports openssl, stunnel and taylor-uucp.

Taylor-UUCP has proven as the de-facto standard UUCP implementation in the last 6 years or so. It should be included in almost any GNU/Linux distribution - so the following URL is just for reference: `ftp://ftp.gnu.org/gnu/uucp`

Stunnel is a standalone SSL tunneling program which may be used to encapsulate plaintext protocols like POP3 into SSL. Stunnel is also included in major current GNU/Linux distributions. It may be obtained from `http://www.stunnel.org` Both versions 3 and 4 are mentioned.

3 Setup of the UUCP master

The UUCP master is the passive end of a connection. It is the 'server' side waiting for inbound connections.

3.1 Configuration of the UUCP system

You just set up the configuration for the UUCP slave systems as usual. There's no difference between UUCP/serial UUCP/tcp or UUCP/ssl

3.2 Configuration of (x)INETD for UUCP/tcp

Unless we already have UUCP/tcp running, we have to set up our super inet daemon to handle incoming requests.

if You use inetd, the line for UUCP looks like

```
uucp    stream  tcp     nowait  uucp    /usr/sbin/tcpd  /usr/sbin/uucico -l
```

for xinetd based systems, the UUCP section looks like

```
service uucp
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = uucp
        server          = /usr/sbin/uucico
        server_args     = -l
}
```

don't forget to tell your inet daemon about your changes using `killall -HUP inetd` respectively `killall -USR1 xinetd`

3.3 Entry in /etc/services

IANA has now assigned a port number for UUCP over SSL, port 4031. Of course you could choose any other port, too - it would work - but nonetheless, it's always a good idea to follow standards. Just put an apropriate line into /etc/services:

```
suucp           4031/tcp
```

3.4 Configuration of UUCP/ssl

You have to obtain (or create) a certificate to be used with stunnel. Please read `http://www.stunnel.org/faq/certs.html`. The following examples expect your certificate to be in /usr/local/ssl/certs/stunnel.pem

Once the changes to our (x)inetd configuration file are made, it is neccessary to have them reloaded using the `killall -HUP inetd` respectively `killall -USR1 xinetd` command.

3.4.1 Stunnel 3

Next it is neccessary to add another entry to your (x)inetd.conf:

```
suucp   stream  tcp     nowait  uucp    /usr/sbin/tcpd  /usr/sbin/stunnel -p /usr/local/ssl/certs/stunnel.pem -r localhost:uucp

service suucp
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = uucp
        server          = /usr/sbin/stunnel
        server_args     = -p /usr/local/ssl/certs/stunnel.pem -r localhost:uucp
}
```

3.4.2 Stunnel 4

You'll need to create a config file (/usr/local/etc/stunnel.suucp.conf in this example)

```
cert = /usr/local/ssl/certs/stunnel.pem
connect = 540
```

Next it is neccessary to add another entry to your (x)inetd.conf:

```
suucp   stream  tcp     nowait  uucp    /usr/sbin/tcpd  /usr/sbin/stunnel /usr/local/etc/stunnel.suucp.conf

service suucp
{
        socket_type     = stream
        protocol        = tcp
```

```
        wait            = no
        user            = uucp
        server          = /usr/sbin/stunnel
        server_args     = /usr/local/etc/stunnel.suucp.conf
}
```

4 Setup of the UUCP slave

Just configure your UUCP as usual (i.e. if you were using a serial line or plaintext TCP). There are only two minor changes neccessary for using SSL:

4.1 Changes to the ports file

The ports file lists all uucp-usable ports. The ports are usually modem lines. In our special case for UUCP over SSL we use the stdin special port. Just add an entry like

```
port stdin
type stdin
```

4.2 Changes to the sys file

The sys file contains a list of all uucp systems we may want to connect. I'm not describing the usual setup of an uucp system, have a look at some more generic information about UUCP configuration. In order to configure a system for UUCP over SSL just modify the 'port' line of the system you want to connect over SSL to:

```
port stdin
```

4.3 Calling out using Stunnel 3

Let's assume your uplink system is called 'chico' and has the FQDN 'chico.foo.bar'. Instead of using

```
uucico -S chico
```

you type

```
stunnel -c -r chico.foo.bar:4031 -l uucico -- uucico -S chico -D
```

For currently unknown reasons this only works as root.

4.4 Calling out using Stunnel 4

Let's assume your uplink system is called 'chico' and has the FQDN 'chico.foo.ba r'. Create a config file (In this case in /usr/local/etc/stunnel.suucp.conf)

```
cert = /usr/local/etc/ssl/certs/stunnel.pem
client = yes
foreground = no
connect = chico.foo.bar:4031
exec = uucico
execargs = uucico -S chico -D
```

Then invoke stunnel as :

```
stunnel /usr/local/etc/stunnel.suucp.conf
```