NAME

smsyncd — simple mtree sync daemon

SYNOPSIS

smsyncd [**-c** *file*] [**-dfh**] [**-h**] [**-p** *port*]

DESCRIPTION

The **smsyncd** program allows synchronization of a file system hierarchy from a *smsync* client. It reads a list of files with attributes from the client and compares it to the requested hierarchy. Each file that is determined to not match the given specification is requested from the client and subsequently copied into place.

OPTIONS

The following options are available:

-c *file*
: Read the daemon's configuration from *file*. If this flag is not specified, **smsyncd** will try to read from */etc/smsyncd.conf*.

-d
: Increase debugging verbosity.

-f
: Per default, **smsyncd** detaches itself from the terminal and runs as a standard unix daemon. Passing this options causes **smsyncd** to run in the foreground.

-h
: Print the online help and exit successfully.

-p *port*
: This option causes **smsyncd** to bind to the given *port* instead of the standard port 4242.

NOTES

smsyncd expects the input from the client to be in the format generated by *mtree* **-C -kall**. It invokes *mtree* **-U** with this input and then records which files differ. It then sends a list of indeces to the client to request the needed files. For each file in the list, it will expect the client to send it the entire file.

The client and server communicate with each other via the smsync(4) protocol. All communication is encrypted using a one-time session key generated after successful authorization using the pre-shared key. This pre-shared secret is stored in the configuration file. To ensure the confidentiality of the secret, **smsyncd** will refuse to start up if the file does not have the proper permissions.

FILES

`/etc/smsyncd.conf` default configuration file

EXIT STATUS

The **smsyncd** utility exits 0 on success, and >0 if an error occurs.

SEE ALSO

smsync(1), smsync(4), smsyncd.conf(5), mtree(1)

HISTORY

The **smsyncd** program was first assigned as the final project for the class CS765 "Advanced Programming in the UNIX Environment" during the fall semester of 2005 at Stevens Institute of Technology.

CAVEATS

The client the and the server need to have a pre-shared secret in order for the authentication and encryption to work.

BUGS

Well, let's see...