NAME

monkeysphere−host − Monkeysphere host key administration tool.

SYNOPSIS

monkeysphere−host *subcommand* **[***args***]**

DESCRIPTION

Monkeysphere is a framework to leverage the OpenPGP web of trust for SSH and TLS key−based authentication.

monkeysphere−host stores and manages OpenPGP certificates for various services offered by the host.

Most subcommands take a KEYID argument, which identifies (by OpenPGP key ID (e.g. 0xDEADBEEF) or full OpenPGP fingerprint) which certificate is to be operated upon. If only one certificate is currently managed by **monkeysphere−host**, the KEYID argument may be omitted, and **monkeysphere−host** will operate on it.

SUBCOMMANDS

monkeysphere−host takes various subcommands:

import−key FILE SCHEME://HOSTNAME[:PORT]

Import an SSH host secret key from file FILE. If FILE is '−', then the key will be imported from stdin, and must be an RSA key in PEM−encoded format. SCHEME://HOSTNAME[:PORT] is used to specify the scheme (e.g. ssh or https), fully−qualified hostname (and port) used in the user ID of the new OpenPGP key (e.g. ssh://example.net or https://www.example.net). If PORT is not specified, then no port is added to the user ID, which means the default port for that service (e.g. 22 for ssh) is assumed. 'i' may be used in place of 'import−key'.

show−keys [KEYID ...]

Output information about the OpenPGP certificate(s) for services offered by the host, including their KEYIDs. If no KEYID is specified (or if the special string '−−all' is used), output information about all certificates managed by **monkeysphere−host**. 's' may be used in place of 'show−keys'.

set−expire EXPIRE [KEYID]

Extend the validity of the OpenPGP certificate specified until EXPIRE from the present. Expiration is specified as with GnuPG (measured from today's date):

0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years

'e' may be used in place of 'set−expire'.

add−servicename SCHEME://HOSTNAME[:PORT] [KEYID]

Add a service−specific user ID to the specified certificate. For example, the operator of 'https://example.net' may wish to add an additional servicename of 'https://www.example.net' to the certificate corresponding to the secret key used by the TLS−enabled web server. 'add−name' or 'n+' may be used in place of 'add−servicename'.

revoke−servicename SCHEME://HOSTNAME[:PORT] [KEYID]

Revoke a service−specific user ID from the specified certificate. 'revoke−name' or 'n−' may be used in place of 'revoke−servicename'.

add−revoker REVOKER_KEYID|FILE [KEYID]

Add a revoker to the specified OpenPGP certificate. The revoker can be specified by their own REVOKER_KEYID (in which case it will be loaded from an OpenPGP keyserver), or by

specifying a path to a file containing the revoker's OpenPGP certificate, or by specifying '−' to load from stdin. 'r+' may be be used in place of 'add−revoker'.

revoke−key [KEYID]
Generate (with the option to publish) a revocation certificate for given OpenPGP certificate. If such a certificate is published, the given key will be permanently revoked, and will no longer be accepted by monkeysphere−enabled clients. This subcommand will ask you a series of questions, and then generate a key revocation certificate, sending it to stdout. You might want to store these certificates safely offline, to publish in case of compromise). If you explicitly tell it to publish the revocation certificate immediately, it will send it to the public keyservers. PUBLISH THESE CERTIFICATES ONLY IF YOU ARE SURE THE CORRESPONDING KEY WILL NEVER BE RE−USED!

publish−keys [KEYID ...]
Publish the specified OpenPGP certificates to the public keyservers. If the special string '−−all' is specified, all of the host's OpenPGP certificates will be published. 'p' may be used in place of 'publish−keys'. NOTE: that there is no way to remove a key from the public keyservers once it is published!

version
Show the monkeysphere version number. 'v' may be used in place of 'version'.

help Output a brief usage summary. 'h' or '?' may be used in place of 'help'.

diagnostics
Review the state of the monkeysphere server host key and report on suggested changes. Among other checks, this includes making sure there is a valid host key, that the key is not expired, that the sshd configuration points to the right place, etc. 'd' may be used in place of 'diagnostics'.

SETUP SSH SERVER CERTIFICATES

To enable users to verify your SSH host's key via the monkeysphere, an OpenPGP certificate must be made out of the host's RSA ssh key, and the certificate must be published to the Web of Trust. Certificate publication is not done by default. The first step is to import the host's ssh key into a monkeysphere−style OpenPGP certificate. This is done with the import−key command. For example:

monkeysphere−host import−key /etc/ssh/ssh_host_rsa_key ssh://host.example.org

On most systems, sshd's RSA secret key is stored at /etc/ssh/ssh_host_rsa_key.

See PUBLISHING AND CERTIFYING MONKEYSPHERE SERVICE CERTIFICATES for how to make sure your users can verify the ssh service offered by your host once the key is imported into **monkeysphere−host**.

SETUP WEB SERVER CERTIFICATES

You can set up your HTTPS−capable web server so that your users can verify it via the monkeysphere, without changing your server's software at all. You just need access to a (PEM−encoded) version of the server's RSA secret key (most secret keys are already stored PEM−encoded). The first step is to import the web server's key into a monkeysphere−style OpenPGP certificate. This is done with the import−key command. For example:

monkeysphere−host import−key /etc/ssl/private/host.example.net−key.pem https://host.example.net

If you don't know where the web server's key is stored on your machine, consult the configuration files for your web server. Debian−based systems using the 'ssl−cert' packages often have a default self−signed certificate stored in '/etc/ssl/private/ssl−cert−snakeoil.key' ; if you're using that key, your users are getting browser warnings about it. You can keep using the same key, but help them use the OpenPGP WoT to verify that it does belong to your web server by using something like:

monkeysphere−host import−key /etc/ssl/private/ssl−cert−snakeoil.key https://$(hostname −−fqdn)

If you offer multiple HTTPS websites using the same secret key, you should add the additional website names with the ‘add−servicename’ subcommand.

See PUBLISHING AND CERTIFYING MONKEYSPHERE SERVICE CERTIFICATES (the next section) for how to make sure your users can verify the https service offered by your host once the key is imported and any extra site names have been added. Note that you can add or remove additional servicenames at any time, but you’ll need to certify any new ones separately.

PUBLISHING AND CERTIFYING MONKEYSPHERE SERVICE CERTIFICATES

Once the host key has been imported, the corresponding certificate must be published to the Web of Trust so that users can retrieve the cert when connecting to the host. The host certificates are published to the keyserver with the publish−key command:

$ monkeysphere−host publish−key −−all

In order for users accessing the system to be able to identify the host’s service via the monkeysphere, at least one person (e.g. a server admin) will need to sign the host’s certificate. This is done using standard OpenPGP keysigning techniques. Usually: pull the host’s OpenPGP certificate from the keyserver, verify and sign it, and then re−publish your signature. More than one person can certify any certificate. Please see https://web.monkeysphere.info/doc/host−keys/ for more information and details. Once an admin’s signature is published, users accessing the host can use the certificate to validate the host’s key without having to manually check the host key’s fingerprint (in the case of ssh) or without seeing a nasty "security warning" in their browsers (in the case of https).

SECURITY CONSIDERATIONS

Note that **monkeysphere−host** currently caches a copy of all imported secret keys (stored in OpenPGP form for future manipulation) in /var/lib/monkeysphere/host/. Cleartext backups of files in this directory could expose secret key material if not handled sensitively.

ENVIRONMENT

The following environment variables will override those specified in the config file (defaults in parentheses):

MONKEYSPHERE_LOG_LEVEL
: Set the log level. Can be SILENT, ERROR, INFO, VERBOSE, DEBUG, in increasing order of verbosity. (INFO)

MONKEYSPHERE_KEYSERVER
: OpenPGP keyserver to use. (pool.sks−keyservers.net)

MONKEYSPHERE_PROMPT
: If set to ‘false’, never prompt the user for confirmation. (true)

FILES

/etc/monkeysphere/monkeysphere−host.conf
: System monkeysphere−host config file.

/var/lib/monkeysphere/host_keys.pub.pgp
: A world−readable copy of the host’s OpenPGP certificates in ASCII armored format. This includes the certificates (including the public keys, servicename−based User IDs, and most recent relevant self−signatures) corresponding to every key used by Monkeysphere−enabled services on the host.

/var/lib/monkeysphere/host/
: A locked directory (readable only by the superuser) containing copies of all imported secret keys (this is the host's GNUPGHOME directory).

/etc/monkeysphere/monkeysphere−host−x509−anchors.crt or /etc/monkeysphere/monkeysphere−x509−anchors.crt
: If monkeysphere-host is configured to query an hkps keyserver for publish-keys, it will use the PEM-encoded X.509 Certificate Authority certificates in this file to validate any X.509 certificates used by the keyserver. If the monkeysphere-host-x509 file is present, the monkeysphere-x509 file will be ignored.

AUTHOR

This man page was written by: Jameson Rollins <jrollins@finestructure.net>, Daniel Kahn Gillmor <dkg@fifthhorseman.net>, Matthew Goins <mjgoins@openflows.com>

SEE ALSO

monkeysphere(1), **monkeysphere**(7), **gpg**(1), **monkeysphere−authentication**(8), **ssh**(1), **sshd**(8)