NAME

monkeysphere - Monkeysphere client user interface

SYNOPSIS

monkeysphere *subcommand* **[***args***]**

DESCRIPTION

Monkeysphere is a framework to leverage the OpenPGP web of trust for OpenSSH and TLS key-based authentication. OpenPGP keys are tracked via GnuPG, and added to the authorized_keys and known_hosts files used by OpenSSH for connection authentication. Monkeysphere can also be used by a validation agent to validate TLS connections (e.g. https).

monkeysphere is the Monkeysphere client utility.

SUBCOMMANDS

monkeysphere takes various subcommands:

update−known_hosts [HOST]...

Update the known_hosts file. For each specified host, gpg will be queried for a key associated with the host URI (see HOST IDENTIFICATION in **monkeysphere(7)),** optionally querying a keyserver. If an acceptable key is found for the host (see KEY ACCEPTABILITY in **monkeysphere(7)),** the key is added to the user's known_hosts file. If a key is found but is unacceptable for the host, any matching keys are removed from the user's known_hosts file. If no gpg key is found for the host, nothing is done. If no hosts are specified, all hosts listed in the known_hosts file will be processed. This subcommand will exit with a status of 0 if at least one acceptable key was found for a specified host, 1 if no matching keys were found at all, and 2 if matching keys were found but none were acceptable. 'k' may be used in place of 'update−known_hosts'.

update−authorized_keys

Update the authorized_keys file for the user executing the command (see MONKEYSPHERE_AUTHORIZED_KEYS in ENVIRONMENT, below). First all monkeysphere keys are cleared from the authorized_keys file. Then, for each user ID in the user's authorized_user_ids file, gpg will be queried for keys associated with that user ID, optionally querying a keyserver. If an acceptable key is found (see KEY ACCEPTABILITY in **monkeysphere**(7)), the key is added to the user's authorized_keys file. If a key is found but is unacceptable for the user ID, any matching keys are removed from the user's authorized_keys file. If no gpg key is found for the user ID, nothing is done. This subcommand will exit with a status of 0 if at least one acceptable key was found for a user ID, 1 if no matching keys were found at all, and 2 if matching keys were found but none were acceptable. 'a' may be used in place of 'update−authorized_keys'.

gen−subkey [KEYID]

Generate an authentication subkey for a private key in your GnuPG keyring. KEYID is the key ID for the primary key for which the subkey with "authentication" capability will be generated. If no key ID is specified, but only one key exists in the secret keyring, that key will be used. The length of the generated key can be specified with the '−−length' or '−l' option. 'g' may be used in place of 'gen−subkey'.

ssh−proxycommand [--no-connect] HOST [PORT]

An ssh ProxyCommand that can be used to trigger a monkeysphere update of the ssh known_hosts file for a host that is being connected to with ssh. This works by updating the known_hosts file for the host first, before an attempted connection to the host is made. Once the known_hosts file has been updated, a TCP connection to the host is made by exec'ing netcat(1). Regular ssh communication is then done over this netcat TCP connection (see ProxyCommand in ssh_config(5) for more info).

This command is meant to be run as the ssh "ProxyCommand". This can either be done by

specifying the proxy command on the command line:

ssh −o ProxyCommand="monkeysphere ssh−proxycommand %h %p" ...

or by adding the following line to your ˜/.ssh/config script:

ProxyCommand monkeysphere ssh−proxycommand %h %p

The script can easily be incorporated into other ProxyCommand scripts by calling it with the "−−no−connect" option, i.e.:

monkeysphere ssh−proxycommand −−no−connect $HOST $PORT

This will run everything except the final exec of netcat to make the TCP connection to the host. In this way this command can be added to another proxy command that does other stuff, and then makes the connection to the host itself. For example, in ˜/.ssh/config:

ProxyCommand sh -c 'monkeysphere ssh-proxycommand --no-connect %h %p ; ssh -W %h:%p jumphost.example.net'

KEYSERVER CHECKING: The proxy command has a fairly nuanced policy for when keyservers are queried when processing a host. If the host userID is not found in either the user's keyring or in the known_hosts file, then the keyserver is queried for the host userID. If the host userID is found in the user's keyring, then the keyserver is not checked. This assumes that the keyring is kept up-to-date, in a cronjob or the like, so that revocations are properly handled. If the host userID is not found in the user's keyring, but the host is listed in the known_hosts file, then the keyserver is not checked. This last policy might change in the future, possibly by adding a deferred check, so that hosts that go from non-monkeysphere-enabled to monkeysphere-enabled will be properly checked.

Setting the CHECK_KEYSERVER variable in the config file or the MONKEYSPHERE_CHECK_KEYSERVER environment variable to either 'true' or 'false' will override the keyserver-checking policy defined above and either always or never check the keyserver for host key updates.

subkey−to−ssh−agent [ssh−add arguments]
Push all authentication-capable subkeys in your GnuPG secret keyring into your running ssh-agent. Additional arguments are passed through to **ssh−add**(1). For example, to remove the authentication subkeys, pass an additional '−d' argument. To require confirmation on each use of the key, pass '−c'. The MONKEYSPHERE_SUBKEYS_FOR_AGENT environment can be used to specify the full fingerprints of specific keys to add to the agent (space separated), instead of adding them all. 's' may be used in place of 'subkey−to−ssh−agent'.

keys−for−userid USERID
Output to stdout all acceptable keys for a given user ID. 'u' may be used in place of 'keys−for−userid'.

sshfprs−for−userid USERID
Output the ssh fingerprints of acceptable keys for a given user ID.

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'.

ENVIRONMENT

The following environment variables will override those specified in the monkeysphere.conf configuration 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_GNUPGHOME, GNUPGHOME
: GnuPG home directory. (~/.gnupg)

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

MONKEYSPHERE_CHECK_KEYSERVER
: Whether or not to check keyserver when making gpg queries. (true)

MONKEYSPHERE_KNOWN_HOSTS
: Path to ssh known_hosts file. (~/.ssh/known_hosts)

MONKEYSPHERE_HASH_KNOWN_HOSTS
: Whether or not to hash to the known_hosts file entries. (false)

MONKEYSPHERE_AUTHORIZED_KEYS
: Path to ssh authorized_keys file. (~/.ssh/authorized_keys)

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

MONKEYSPHERE_STRICT_MODES
: If set to ‘false’, ignore too-loose permissions on known_hosts, authorized_keys, and authorized_user_ids files. NOTE: setting this to false may expose you to abuse by other users on the system. (true)

MONKEYSPHERE_SUBKEYS_FOR_AGENT
: A space-separated list of authentication-capable subkeys to add to the ssh agent with subkey-to-ssh-agent.

FILES

~/.monkeysphere/monkeysphere.conf
: User monkeysphere config file.

/etc/monkeysphere/monkeysphere.conf
: System-wide monkeysphere config file.

~/.monkeysphere/authorized_user_ids
: A list of OpenPGP user IDs, one per line. OpenPGP keys with an exactly-matching User ID (calculated valid by the designated identity certifiers), will have any valid authorization-capable keys or subkeys added to the given user’s authorized_keys file.

AUTHOR

Written by: Jameson Rollins <jrollins@finestructure.net>, Daniel Kahn Gillmor <dkg@fifthhorseman.net>

SEE ALSO

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