NAME

ssh-agent - OpenSSH authentication agent

SYNOPSIS

ssh-agent [**-c** | **-s**] [**-Ddx**] [**-a** *bind_address*] [**-E** *fingerprint_hash*] [**-O** *option*] [**-P** *allowed_providers*] [**-t** *life*]
ssh-agent [**-a** *bind_address*] [**-E** *fingerprint_hash*] [**-O** *option*] [**-P** *allowed_providers*] [**-t** *life*] *command* [*arg ...*]
ssh-agent [**-c** | **-s**] **-k**

DESCRIPTION

ssh-agent is a program to hold private keys used for public key authentication. Through use of environment variables the agent can be located and automatically used for authentication when logging in to other machines using ssh(1).

The options are as follows:

-a *bind_address*
Bind the agent to the UNIX-domain socket *bind_address*. The default is *$TMPDIR/ssh-XXXXXXXXXX/agent.<ppid>*.

-c Generate C-shell commands on stdout. This is the default if SHELL looks like it's a csh style of shell.

-D Foreground mode. When this option is specified, **ssh-agent** will not fork.

-d Debug mode. When this option is specified, **ssh-agent** will not fork and will write debug information to standard error.

-E *fingerprint_hash*
Specifies the hash algorithm used when displaying key fingerprints. Valid options are: "md5" and "sha256". The default is "sha256".

-k Kill the current agent (given by the SSH_AGENT_PID environment variable).

-O *option*
Specify an option when starting **ssh-agent**. Currently two options are supported: **allow-remote-pkcs11** and **no-restrict-websafe**.

The **allow-remote-pkcs11** option allows clients of a forwarded **ssh-agent** to load PKCS#11 or

FIDO provider libraries. By default only local clients may perform this operation. Note that signalling that an **ssh-agent** client is remote is performed by ssh(1), and use of other tools to forward access to the agent socket may circumvent this restriction.

The **no-restrict-websafe** option instructs **ssh-agent** to permit signatures using FIDO keys that might be web authentication requests. By default, **ssh-agent** refuses signature requests for FIDO keys where the key application string does not start with "ssh:" and when the data to be signed does not appear to be a ssh(1) user authentication request or a ssh-keygen(1) signature. The default behaviour prevents forwarded access to a FIDO key from also implicitly forwarding the ability to authenticate to websites.

-P *allowed_providers*
Specify a pattern-list of acceptable paths for PKCS#11 provider and FIDO authenticator middleware shared libraries that may be used with the **-S** or **-s** options to ssh-add(1). Libraries that do not match the pattern list will be refused. See PATTERNS in ssh_config(5) for a description of pattern-list syntax. The default list is "usr/lib*/*,/usr/local/lib*/*".

-s Generate Bourne shell commands on stdout. This is the default if SHELL does not look like it’s a csh style of shell.

-t *life* Set a default value for the maximum lifetime of identities added to the agent. The lifetime may be specified in seconds or in a time format specified in sshd_config(5). A lifetime specified for an identity with ssh-add(1) overrides this value. Without this option the default maximum lifetime is forever.

-x Exit after the last client has disconnected.

command [*arg* ...]
If a command (and optional arguments) is given, this is executed as a subprocess of the agent. The agent exits automatically when the command given on the command line terminates.

There are two main ways to get an agent set up. The first is at the start of an X session, where all other windows or programs are started as children of the **ssh-agent** program. The agent starts a command under which its environment variables are exported, for example **ssh-agent xterm &**. When the command terminates, so does the agent.

The second method is used for a login session. When **ssh-agent** is started, it prints the shell commands required to set its environment variables, which in turn can be evaluated in the calling shell, for example **eval ‘ssh-agent -s‘**.

In both cases, ssh(1) looks at these environment variables and uses them to establish a connection to the agent.

The agent initially does not have any private keys. Keys are added using ssh-add(1) or by ssh(1) when **AddKeysToAgent** is set in ssh_config(5). Multiple identities may be stored in **ssh-agent** concurrently and ssh(1) will automatically use them if present. ssh-add(1) is also used to remove keys from **ssh-agent** and to query the keys that are held in one.

Connections to **ssh-agent** may be forwarded from further remote hosts using the **-A** option to ssh(1) (but see the caveats documented therein), avoiding the need for authentication data to be stored on other machines. Authentication passphrases and private keys never go over the network: the connection to the agent is forwarded over SSH remote connections and the result is returned to the requester, allowing the user access to their identities anywhere in the network in a secure fashion.

ENVIRONMENT

SSH_AGENT_PID When **ssh-agent** starts, it stores the name of the agent's process ID (PID) in this variable.

SSH_AUTH_SOCK

When **ssh-agent** starts, it creates a UNIX-domain socket and stores its pathname in this variable. It is accessible only to the current user, but is easily abused by root or another instance of the same user.

FILES

$TMPDIR/ssh-XXXXXXXXXX/agent.<ppid>

UNIX-domain sockets used to contain the connection to the authentication agent. These sockets should only be readable by the owner. The sockets should get automatically removed when the agent exits.

SEE ALSO

ssh(1), ssh-add(1), ssh-keygen(1), ssh_config(5), sshd(8)

AUTHORS

OpenSSH is a derivative of the original and free ssh 1.2.12 release by Tatu Ylonen. Aaron Campbell, Bob Beck, Markus Friedl, Niels Provos, Theo de Raadt and Dug Song removed many bugs, re-added newer features and created OpenSSH. Markus Friedl contributed the support for SSH protocol versions 1.5 and 2.0.