NAME

ssh-agent — authentication agent

SYNOPSIS

ssh-agent [**-c** | **-s**] [**-Dd**] [**-a** *bind_address*] [**-E** *fingerprint_hash*]
[**-P** *pkcs11_whitelist*] [**-t** *life*] [*command* [*arg ...*]]
ssh-agent [**-c** | **-s**] **-k**

DESCRIPTION

ssh-agent is a program to hold private keys used for public key authentication (RSA, DSA, ECDSA, Ed25519). **ssh-agent** is usually started in the beginning of an X-session or a login session, and all other windows or programs are started as clients to the ssh-agent program. 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 agent initially does not have any private keys. Keys are added using ssh(1) (see **AddKeysToAgent** in ssh_config(5) for details) or ssh-add(1). 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.

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).

-P *pkcs11_whitelist*
: Specify a pattern-list of acceptable paths for PKCS#11 shared libraries that may be added using the **-s** option to ssh-add(1). The default is to allow loading PKCS#11 libraries from "/usr/lib/*,/usr/local/lib/*". PKCS#11 libraries that do not match the whitelist will be refused. See PATTERNS in ssh_config(5) for a description of pattern-list syntax.

-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.

If a command line is given, this is executed as a subprocess of the agent. When the command dies, so does the agent.

The idea is that the agent is run in the user's local PC, laptop, or terminal. Authentication data need not be stored on any other machine, and authentication passphrases never go over the network. However, the connection to the agent is forwarded over SSH remote logins, and the user can thus use the privileges given by the identities anywhere in the network in a secure way.

There are two main ways to get an agent set up: The first is that the agent starts a new subcommand into which some environment variables are exported, eg **ssh-agent xterm &**. The second is that the agent prints the needed shell commands (either sh(1) or csh(1) syntax can be generated) which can be evaluated in the calling shell, eg **eval 'ssh-agent -s'** for Bourne-type shells such as sh(1) or ksh(1) and **eval 'ssh-agent -c'** for csh(1) and deri vatives.

Later ssh(1) looks at these variables and uses them to establish a connection to the agent.

The agent will never send a private key over its request channel. Instead, operations that require a private key will be performed by the agent, and the result will be returned to the requester. This way, private keys are not exposed to clients using the agent.

A UNIX-domain socket is created and the name of this socket is stored in the SSH_AUTH_SOCK environment variable. The socket is made accessible only to the current user. This method is easily abused by root or another instance of the same user.

The SSH_AGENT_PID environment variable holds the agent's process ID.

The agent exits automatically when the command given on the command line terminates.

In Debian, **ssh-agent** is installed with the set-group-id bit set, to prevent ptrace(2) attacks retrieving private key material. This has the side-effect of causing the run-time linker to remove certain environment variables which might have security implications for set-id programs, including LD_PRELOAD, LD_LIBRARY_PATH, and TMPDIR. If you need to set any of these environment variables, you will need to do so in the program executed by ssh-agent.

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), 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.