## NAME

ssh-keyscan — gather SSH public keys from servers

## SYNOPSIS

**ssh-keyscan** [-46cDHqv] [-f *file*] [-O *option*] [-p *port*] [-T *timeout*] [-t *type*] [*host* | *addrlist namelist*]

## DESCRIPTION

**ssh-keyscan** is a utility for gathering the public SSH host keys of a number of hosts. It was designed to aid in building and verifying *ssh_known_hosts* files, the format of which is documented in *sshd*(8). **ssh-keyscan** provides a minimal interface suitable for use by shell and perl scripts.

**ssh-keyscan** uses non-blocking socket I/O to contact as many hosts as possible in parallel, so it is very efficient. The keys from a domain of 1,000 hosts can be collected in tens of seconds, even when some of those hosts are down or do not run *sshd*(8). For scanning, one does not need login access to the machines that are being scanned, nor does the scanning process involve any encryption.

Hosts to be scanned may be specified by hostname, address or by CIDR network range (e.g. 192.168.16/28). If a network range is specified, then all addresses in that range will be scanned.

The options are as follows:

`-4` Force **ssh-keyscan** to use IPv4 addresses only.

`-6` Force **ssh-keyscan** to use IPv6 addresses only.

`-c` Request certificates from target hosts instead of plain keys.

`-D` Print keys found as SSHFP DNS records. The default is to print keys in a format usable as an *ssh*(1) *known_hosts* file.

`-f` *file*
: Read hosts or "addrlist namelist" pairs from *file*, one per line. If '-' is supplied instead of a filename, **ssh-keyscan** will read from the standard input. Names read from a file must start with an address, hostname or CIDR network range to be scanned. Addresses and hostnames may optionally be followed by comma-separated name or address aliases that will be copied to the output. For example:

```
192.168.11.0/24
10.20.1.1
happy.example.org
10.0.0.1,sad.example.org
```

`-H` Hash all hostnames and addresses in the output. Hashed names may be used normally by *ssh*(1) and *sshd*(8), but they do not reveal identifying information should the file's contents be disclosed.

`-O` *option*
: Specify a key/value option. At present, only a single option is supported:

  `hashalg=`*algorithm*
  : Selects a hash algorithm to use when printing SSHFP records using the `-D` flag. Valid algorithms are "sha1" and "sha256". The default is to print both.

`-p` *port*
: Connect to *port* on the remote host.

`-q` Quiet mode: do not print server host name and banners in comments.

`-T` *timeout*
: Set the timeout for connection attempts. If *timeout* seconds have elapsed since a connection was initiated to a host or since the last time anything was read from that host, the connection is closed and the host in question considered unavailable. The default is 5 seconds.

`-t` *type*
: Specify the type of the key to fetch from the scanned hosts. The possible values are "ecdsa", "ed25519", "ecdsa-sk", "ed25519-sk", or "rsa". Multiple values may be specified by separating them with commas. The default is to fetch all the above key types.

`-v`
: Verbose mode: print debugging messages about progress.

If an ssh_known_hosts file is constructed using **ssh-keyscan** without verifying the keys, users will be vulnerable to *man in the middle* attacks. On the other hand, if the security model allows such a risk, **ssh-keyscan** can help in the detection of tampered keyfiles or man in the middle attacks which have begun after the ssh_known_hosts file was created.

## FILES

*/etc/ssh/ssh_known_hosts*

## EXAMPLES

Print the RSA host key for machine *hostname*:

```
$ ssh-keyscan -t rsa hostname
```

Search a network range, printing all supported key types:

```
$ ssh-keyscan 192.168.0.64/25
```

Find all hosts from the file *ssh_hosts* which have new or different keys from those in the sorted file *ssh_known_hosts*:

```
$ ssh-keyscan -t rsa,ecdsa,ed25519 -f ssh_hosts | \
        sort -u - ssh_known_hosts | diff ssh_known_hosts -
```

## SEE ALSO

*ssh*(1), *sshd*(8)

*Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints*, RFC 4255, 2006.

## AUTHORS

David Mazieres <*dm@lcs.mit.edu*> wrote the initial version, and Wayne Davison <*wayned@users.sourceforge.net*> added support for protocol version 2.