Rocky Enterprise Linux 9.2 Manual Pages on command 'hosts.equiv.5'

$ man hosts.equiv.5

HOSTS.EQUIV(5) Linux Programmer's Manual HOSTS.EQUIV(5)

NAME

hosts.equiv - list of hosts and users that are granted "trusted" r com?

mand access to your system

DESCRIPTION

The file /etc/hosts.equiv allows or denies hosts and users to use the

r-commands (e.g., rlogin, rsh, or rcp) without supplying a password.

The file uses the following format:

+|[-]hostname|+@netgroup|-@netgroup [+|[-]username|+@netgroup|-@net?

group]

The hostname is the name of a host which is logically equivalent to the

local host. Users logged into that host are allowed to access like-

named user accounts on the local host without supplying a password.

The hostname may be (optionally) preceded by a plus (+) sign. If the

plus sign is used alone, it allows any host to access your system. You

can explicitly deny access to a host by preceding the hostname by a mi?

nus (-) sign. Users from that host must always supply additional cre?

dentials, including possibly a password. For security reasons you

should always use the FQDN of the hostname and not the short hostname.

The username entry grants a specific user access to all user accounts (except root) without supplying a password. That means the user is NOT restricted to like-named accounts. The username may be (optionally) preceded by a plus (+) sign. You can also explicitly deny access to a specific user by preceding the username with a minus (-) sign. This says that the user is not trusted no matter what other entries for that host exist.

Netgroups can be specified by preceding the netgroup by an @ sign.

Be extremely careful when using the plus (+) sign. A simple typographical error could result in a standalone plus sign. A standalone plus sign is a wildcard character that means "any host"!

FILES

/etc/hosts.equiv

NOTES

Some systems will honor the contents of this file only when it has owner root and no write permission for anybody else. Some exceptionally paranoid systems even require that there be no other hard links to the file.

Modern systems use the Pluggable Authentication Modules library (PAM). With PAM a standalone plus sign is considered a wildcard character which means "any host" only when the word promiscuous is added to the auth component line in your PAM file for the particular service (e.g., rlogin).

EXAMPLES

Below are some example /etc/host.equiv or ~/.rhosts files.

Allow any user to log in from any host:

```
+
```

Allow any user from host with a matching local account to log in:

```
host
```

Note: the use of +host is never a valid syntax, including attempting to specify that any user from the host is allowed.

Allow any user from host to log in:

host +

Note: this is distinct from the previous example since it does not re?
quire a matching local account.

Allow user from host to log in as any non-root user:

host user

Allow all users with matching local accounts from host to log in except
for baduser:

host -baduser

host

Deny all users from host:

-host

Note: the use of -host -user is never a valid syntax, including at?
tempting to specify that a particular user from the host is not
trusted.

Allow all users with matching local accounts on all hosts in a net?
group:

+@netgroup

Disallow all users on all hosts in a netgroup:

-@netgroup

Allow all users in a netgroup to log in from host as any non-root user:

host +@netgroup

Allow all users with matching local accounts on all hosts in a netgroup
except baduser:

+@netgroup -baduser

+@netgroup

Note: the deny statements must always precede the allow statements be?
cause the file is processed sequentially until the first matching rule
is found.

SEE ALSO

rhosts(5), rlogind(8), rshd(8)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A
description of the project, information about reporting bugs, and the

latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.