Skip to main content.

BSD Now

A Weekly BSD Podcast - News, Interviews and Tutorials

Subscribe on Youtube « | Subscribe with iTunes « | RSS: MP3 | Video | HD Video | HD Torrent Feed
Navigation: Home | Episodes | About | Live | Shirt | Contact Us | Tutorials

Chaining SSH connections

2014-06-18

Live demo in BSD Now Episode 042. | Originally written by Lars, with edits by TJ, for bsdnow.tv | Last updated: 2014/06/18

NOTE: the author/maintainer of the tutorial(s) is no longer with the show, so the information below may be outdated or incorrect.

With people running BSD as firewalls and routers, many have sshd running on them. That means that they can be used as a gateway to machines behind them. If it's a one-off occurrence, then the connection information only needs to be specified on the command line. If it is done more frequently, it can be set in the user's or system's ssh_config. Doing so needs only a port open for SSH and no further per-destination modification of the firewall rules.

Passing through a gateway using netcat mode

As of OpenSSH 5.4, a "netcat mode" can connect stdio on the client to a single port forwarded on the server. This can also be used to connect using ssh, but it needs the ProxyCommand option either as a run time parameter or as part of ~/.ssh/config. However, it no longer needs netcat to be installed on the intermediary machine(s). Here is an example of using it in a run time parameter.

```
$ ssh -o ProxyCommand="ssh -W %h:%p jumphost.example.org" server.example.org
```

In that example, authentication will happen twice - first on the jump host, and then on the final host where it will bring up a shell. The syntax is the same if the gateway is identified in the configuration file. ssh expands the full name of the gateway and the destination from the configuration file. The following allows the destination host to be reached by entering ssh server in the terminal:

```
Host server
  Hostname server.example.org
  ProxyCommand ssh jumphost.example.org -W %h:%p
```

The same can be done for SFTP. Here the destination SFTP server can be reached by entering sftp jump and the configuration file takes care of the rest. If there is a mix up with the final host key, then it is necessary to add in HostKeyAlias to explicitly name which key will be used to identify the destination system.

```
Host sftpserver
  HostName sftpserver.example.org
  HostKeyAlias sftpserver.example.org
  ProxyCommand ssh jumphost.example.org -W %h:%p
```

It is possible to add the key for the gateway to the ssh-agent which you have running, or else specify it in the configuration file. The option "User" refers to the user name on the destination. If the user is the same on both the destination and the originating machine, then it does not need to be used. If the user name is different on the gateway, then the -l option can be used in the ProxyCommand option. Here, the user fred on the local machine logs into the gateway as fred2 and into the destination server as fred3.

```
Host server
  HostName server.example.org
  User fred3
  ProxyCommand ssh -l fred2 -i /home/fred/.ssh/rsa_key jumphost.example.org -W %h:%p
```

If both the gateway and destination are using keys, then the option "IdentityFile" is used to point to the destination's private key.

```
Host jump
  HostName server.example.org
  IdentityFile /home/fred/.ssh/rsa_key_2
  ProxyCommand ssh -i /home/fred/.ssh/rsa_key jumphost.example.org -W %h:%p
```

It's also possible to do that more than one layer deep.

Recursively chaining gateways

It is possible to make the configuration more abstract and allow passing through an arbitrary number of gateways. This particular configuration only works if the user name is the same across all hosts involved. There are limitations resulting from using the slash as a separator, as there would be with other symbols. However, it allows use of dirname and hostname to process the host names.

```
Host * / *
  ProxyCommand ssh $(dirname %h) -W $(basename %h):%p
```

Do not put a space between the "* / *" - that's only there because of a technical problem with Markdown not displaying it correctly otherwise. In this way, hosts are separated with a slash (/) and can be arbitrary in number.

```
$ ssh host1/host2/host3/host4
```

If keys are to be used, then agent forwarding can be specified in the command given in the "ProxyCommand" option using -A and first loading the keys into the agent. The following configuration uses sed to allow different port numbers and user names using the plus sign (+) as the delimiter for hosts, a colon (:) for ports and an equal sign (=) for user names. The basic structure is "ssh $() -W $():$()" and where "%h" is substituted for the target host name.

```
Host *+*
  ProxyCommand ssh -v $(echo %h | sed -e 's/+[^+]*$//;
s/\([^+=]*\)=\([^+]*\)$/\2 -l \1/; s/^\([^+:]*\):\([0-9]*\)+/-p \2 \1+/'
) -W $(echo %h | sed -e 's/^.*+//; s/:.*$//;'):$(echo %h | sed -e '
s/^.*+//; /:/!s/^.*/22/; s/^.*://' ;)
```

The port can be left off for the default of 22 or delimited with a colon (:) for non-standard values.

```
$ ssh host1+host2:2022+host3:2224
```

As is, the colons confound sftp, so the above configuration will only work with it using standard ports. If sftp is needed on non-standard ports then another delimiter, such as an underscore (_), can be configured. Any user name except the final one can be specified for a given host using the designated delimiter, in the above it is an equal sign (=). The destination host's user name is specified with -l and all others can be joined to their corresponding host name with the delimiter.

```
$ ssh -l user3 user1=host1+user2=host2+host3
```

If user names are specified, depending on the delimiter, ssh can be unable to match the final host to an IP number and the key fingerprint in known_hosts. In such cases, it will ask for verification each time the connection is established, but this should not be a problem if the equal sign (=) is used.

Latest News

New announcement

2017-05-25

Hi, Mr. Dexter. Also, we understand that Brad Davis thinks there should be more real news....

Two Year Anniversary

2015-08-08

We're quickly approaching our two-year anniversary, which will be on episode 105. To celebrate, we've created a unique t-shirt design, available for purchase until the end of August. Shirts will be shipped out around September 1st. Most of the proceeds will support the show, and specifically allow us to buy...

New discussion segment

2015-01-17

We're thinking about adding a new segment to the show where we discuss a topic that the listeners suggest. It's meant to be informative like a tutorial, but more of a "free discussion" format. If you have any subjects you want us to explore, or even just a good name...

How did you get into BSD?

2014-11-26

We've got a fun idea for the holidays this year: just like we ask during the interviews, we want to hear how all the viewers and listeners first got into BSD. Email us your story, either written or a video version, and we'll read and play some of them for...

1 | 2 | 3 | 4 | 5 | Next >

Episode 228: The Spectre of Meltdown

2018-01-10

Direct Download:HD VideoMP3 AudioTorrent This episode was brought to you by Headlines Meltdown Spectre Official Site Kernel-memory-leaking Intel processor design flaw forces Linux, Windows redesign Intel’s official response The Register mocks intels response with pithy annotations Intel’s Analysis PDF XKCD Response from FreeBSD FreeBSD's patch WIP Why Raspberry Pi isn’t vulnerable to Spectre or Meltdown Xen mitigation patches Overview of affected FreeBSD Platforms/Architectures Groff's response We’ll...

Episode 227: The long core dump

2018-01-03

Direct Download:HD VideoMP3 AudioTorrent This episode was brought to you by Headlines NetBSD 7.1.1 released The NetBSD Project is pleased to announce NetBSD 7.1.1, the first security/critical update of the NetBSD 7.1 release branch. It represents a selected subset of fixes deemed important for security or stability reasons. Complete source and binaries for NetBSD 7.1.1...

Episode 226: SSL: Santa’s Syscall List

2017-12-27

Direct Download:HD VideoMP3 AudioTorrent This episode was brought to you by Headlines FreeBSD Q3 Status Report 2017 FreeBSD Team Reports FreeBSD Release Engineering Team Ports

Collection The FreeBSD Core Team The FreeBSD Foundation Projects FreeBSD CI Kernel Intel 10G iflib Driver Update Intel iWARP Support pNFS Server Plan B Architectures AMD Zen (family 17h) support Userland Programs Updates to GDB Ports FreeBSDDesktop OpenJFX 8 Puppet Documentation Absolute FreeBSD, 3rd Edition Manual Pages Third-Party Projects The nosh Project FreeBSD...

Episode 225: The one true OS

2017-12-20

Direct Download:HD VideoMP3 AudioTorrent This episode was brought to you by Headlines TrueOS stable release 17.12 We are pleased to announce a new release of the 6-month STABLE version of TrueOS! This release cycle focused on lots of cleanup and stabilization of the distinguishing features of TrueOS: OpenRC, boot speed, removable-device...