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

Reverse SSH tunneling

2014-08-27

Live demo in BSD Now Episode 052. | Originally written by TJ for bsdnow.tv | Last updated: 2014/08/27

NOTE: the author/maintainer of the tutorial(s) is no longer with the show, so the information below may be outdated or incorrect.

We've done a number of SSH tutorials in the past, but most of them rely on the fact that you have a certain level of control on the network. In some cases, you need to be able to access a system that's behind a firewall. This guide will show you how to do just that - reversing the connection and accessing an internal system from the outside. The only requirement in this case is that the firewall allows outbound SSH traffic. You'll have to have access to the machine behind the firewall at some point for this to work. The -R switch will play a key role in this (very short) tutorial. Since we'll be reversing the connection, be sure your client system has a publicly-accessible sshd running. On the system behind the firewall, run the following:

```
$ ssh -fN -R 9000:localhost:22 user@clientip
```

Replace "clientip" with the IP of your system and "22" with the port on which you run sshd. It's recommended to run that command in tmux so it doesn't get lost. You might want to consider running sshd on port 443, so it looks similar to normal SSL traffic. See our stunnel tutorial for more ideas there. Now move back to the client system, and we'll make the reverse connection like so:

```
$ ssh -p 9000 user@127.0.0.1
```

While it may appear to be connecting on the loopback device, it's actually using the already-established connection made by the internal machine. You'll need to use the username and password/key that you normally would for the internal system. Some recommended settings to have in **the client's** /etc/ssh/sshd_config:

```
ClientAliveInterval 300
TCPKeepAlive yes
```

With these, the internal system will send a packet to the client every five minutes to keep the connection from dying due to inactivity. One problem with this setup is, of course, if the first connection dies. Another possible issue is if your client's IP changes. While not much can be done about the first one (aside from maybe a cron job to try and re-establish the connection), there is a good way to handle the second situation. If you use something like SSH chaining, you can leave the internal system connected to a dedicated server whose IP doesn't ever change. From there, connect to the server, then to localhost.

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...