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

Protecting traffic with a BSD-based VPN

2014-08-13

Live demo in BSD Now Episode 050. | Originally written by Adam McDougall, with minor edits by TJ, for bsdnow.tv | Last updated: 2014/09/15

NOTE: the author/maintainer of the tutorial(s) is no longer with the show, so the information below may be outdated or incorrect.

When you're on an untrusted network, the last thing you want is someone logging all your traffic. While doing most of your work over SSH is probably the best solution, sometimes

you want to encapsulate all your traffic in a tunnel. OpenSSH can do that with some extra work, but OpenVPN is tool made for this exact purpose. OpenVPN works on all the BSDs. We'll be using FreeBSD in this tutorial, but you can adapt the file locations and configuration to other BSDs very easily. Install OpenVPN from ports or packages and let's get started. First, we'll add OpenVPN to the startup items **on the client and server**. Run the following on both systems:

```
# sysrc openvpn_enable="YES"
# sysrc openvpn_if="tun"
```

We'll also configure proper logging on **both systems**.

```
# vi /etc/syslog.conf
```

Make the end of /etc/syslog.conf look like this:

```
!openvpn
*.* /var/log/openvpn.log
!*
```

The last !* should already be there.

```
# touch /var/log/openvpn.log
# service syslogd reload
```

Setup log rotation so we don't run out of disk space:

```
# cat << END >> /etc/newsyslog.conf
/var/log/openvpn.log                    600  30    *    @T00  ZC
END
```

Now we will create the certs, so move over to the server system. The OpenVPN port will install easy-rsa, which is a tool to generate certificates. We will make our own working copy of the easy-rsa directory in a secure place:

```
# cp -r /usr/local/share/easy-rsa /root/easy-rsa
# cd /root/easy-rsa
```

Edit variables in the "vars" file to set defaults for the certificates. Most of what you want to change is at the bottom. If you aren't sure what to put, put anything; these can be overridden when certificates are created. "KEY_SIZE" and "KEY_CN" are the most useful ones to set, as "KEY_CN" is used as the default server hostname for certificates. They cannot be blank. Delete the duplicate "KEY_EMAIL" line if it exists.

```
# export KEY_SIZE=2048
# export KEY_COUNTRY="US"
# export KEY_PROVINCE="YourState"
# export KEY_CITY="YourCity"
# export KEY_ORG="YourOrganizationName"
# export KEY_EMAIL="me@myhost.mydomain"
# export KEY_CN="openvpn-server"
# export KEY_NAME=changeme
# export KEY_OU=changeme
```

Prepare a Bourne shell environment for certificate creation and source the vars file:

```
# sh
# . ./vars
```

You must run clean-all the first time before creating certificates. If you already have some, it will wipe them out.

```
# ./clean-all
```

Generate the CA certificate that will be used to sign the others. Just accept defaults unless you want to override them.

```
# ./build-ca
```

Generate the Server certificate using a hostname CN 'openvpn-server'. It's convenient to use the server's hostname for the CN, but not required. Again, just accept defaults unless you want to override them.

```
# ./build-key-server openvpn-server
```

Generate DH keys:

```
# ./build-dh
```

Generate a client certificate for each client, in this case 'openvpn-client'. It's convenient to use the client's hostname for the CN, but not required. It's only used as an identifier.

```
# ./build-key openvpn-client
```

Permanently store the server keys in a secure place:

```
# mkdir -p /usr/local/etc/openvpn/keys
# chmod 700 /usr/local/etc/openvpn/ /usr/local/etc/openvpn/keys
# cp /root/easy-rsa/keys/ca.crt /usr/local/etc/openvpn/keys/
# cp /root/easy-rsa/keys/dh*.pem /usr/local/etc/openvpn/keys/
# cp -p /root/easy-rsa/keys/openvpn-server.crt /usr/local/etc/openvpn/keys/
# cp -p /root/easy-rsa/keys/openvpn-server.key /usr/local/etc/openvpn/keys/
```

On each client, create a secure place for config and keys:

```
# mkdir -p /usr/local/etc/openvpn/keys
# chmod 700 /usr/local/etc/openvpn/ /usr/local/etc/openvpn/keys
```

Securely copy the following client certificate and key files from openvpn-server:

```
/root/easy-rsa/keys/ca.crt
/root/easy-rsa/keys/openvpn-client.crt
/root/easy-rsa/keys/openvpn-client.key
```

to openvpn-client in /usr/local/etc/openvpn/keys. On each client, ensure secure permissions on client key file:

```
# chmod 600 /usr/local/etc/openvpn/keys/*.key
```

Next, we'll edit the server configuration for openvpn.conf:

```
# cp /usr/local/share/examples/openvpn/sample-config-files/server.conf \
   /usr/local/etc/openvpn/openvpn.conf
# chmod 600 /usr/local/etc/openvpn/openvpn.conf
# vi /usr/local/etc/openvpn/openvpn.conf
```

Replace:

```
ca ca.crt
cert server.crt
key server.key  # This file should be kept secret
```

With:

```
ca /usr/local/etc/openvpn/keys/ca.crt
cert /usr/local/etc/openvpn/keys/openvpn-server.crt
key /usr/local/etc/openvpn/keys/openvpn-server.key
```

And then replace:

```
dh dh1024.pem
```

With:

```
dh /usr/local/etc/openvpn/keys/dh2048.pem
```

Finally, replace:

```
comp-lzo
```

With:

```
comp-lzo no
push "comp-lzo no"
```

LZO compression was disabled because of possible chosen-plaintext attacks. Move over to the client, and edit the configuration for openvpn.conf:

```
# cp /usr/local/share/examples/openvpn/sample-config-files/client.conf \
  /usr/local/etc/openvpn/openvpn.conf
# chmod 600 /usr/local/etc/openvpn/openvpn.conf
# vi /usr/local/etc/openvpn/openvpn.conf
```

Replace:

```
remote my-server-1 1194
```

With:

```
remote openvpn-server 1194
topology subnet
```

Use your real server name or IP for 'openvpn-server'. Next, replace:

```
ca ca.crt
cert client.crt
key client.key  # This file should be kept secret
```

With:

```
ca /usr/local/etc/openvpn/keys/ca.crt
cert /usr/local/etc/openvpn/keys/openvpn-client.crt
key /usr/local/etc/openvpn/keys/openvpn-client.key
```

Finally, replace:

```
comp-lzo
```

With:

```
comp-lzo no
```

Now we can start the connections. On the server, run:

```
# service openvpn start
```

Check ifconfig and the log - a new tun interface should appear. Do the same on the client:

```
# service openvpn start
```

The client should be able to ping 10.8.0.1. You may want to move /root/easy-rsa off the server to a safe place to reduce the risk of all keys being copied by an attacker. It is self-contained and you can still use it to generate more keys elsewhere. If you want, you can use the 'cipher' option to choose stronger encryption and 'auth' plus 'tls-auth' to enable HMAC on packets. If you enable

```
push "redirect-gateway def1 bypass-dhcp"
```

on the server, connecting clients will temporarily override the default gateway and setup a new default gateway pointing at the VPN server through the tunnel. The VPN server would need to be setup as a router (gateway_enable=YES in rc.conf) and NAT to properly route replies to external network traffic. It is also possible to route traffic to/from the client without NAT using a more involved configuration. If you restart the server, clients will reconnect after the timeout defined in the client configuration. See the 'openvpn' manpage for descriptions of configuration file parameters.

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...