FreeBSD

Kernel Compilation

For Implementing

Interception (Transparent) Proxy

Prepared by
ViSolve Squid Team
May 21, 2009

Disclaimer :

The information in this document is believed to be accurate in all respects to the best of ViSolve's Knowledge at the time of publication and is subject to change without prior notice.

ViSolve Inc., is not legally liable for any errors or opinions that appear in this document.

Introduction | What is FreeBSD? | Why Interception (Transparent) Proxy?
FreeBSD & Interception Proxy | Steps to recompile Kernel | Conclusion | About ViSolve

Introduction

This document describes about the basic compilation procedure of FreeBSD to implement Interception (Transparent) Proxy. This manual should not be treated as a tutorial but only as a reference to begin with. To configure squid as Transparent Proxy, please refer our technical white paper "Implementing Transparent Caching Using Squid".

What is FreeBSD?

FreeBSD is a member of the Unix family of operating systems that is based on the Berkeley Software Distribution (BSD), which is completely free, and runs on a variety of different types of computers. It runs on processors such as the Pentium that are compatible with Intel's x86 architecture and also on AMD64, Power PC, ARM, IA-64, PC-98 and UltraSPARC processors.

FreeBSD is developed as a complete operating system. The same source code revision tracing tree encompasses the kernel, device drivers and all of the userland utilities, such as the shell, whereas in Linux distributions, the kernel, userland utilities and applications are developed separately and packaged together in different ways by others.

FreeBSD offers advanced networking, performance, security and compatibility features, which makes it an ideal Internet or Intranet server. It provides robust network services under the heaviest loads and uses memory efficiently to maintain good response times for thousands of simultaneous user processes.

FreeBSD is an alternative to Linux that will run Linux applications without the need for any compatibility layer and supports all major X Window desktops, such as KDE and GNOME. It's important to note that FreeBSD and Linux are two different operating systems, developed independently of each other throughout their period of existence.

FreeBSD is released under variety of licenses. The kernel code and most newly created code is released under the two-clause BSD license which allows everyone to use and redistribute FreeBSD as they wish. There are parts released under three and four-clause BSD licenses, as well as the GPL, LGPL, ISC, CDDL and Beerware licenses.

Why Interception (Transparent) Proxy?

The full explanation about the term "Transparent Caching and Transparent Proxying" depends on the context, but we assume the context here as HTTP proxy/caches with transparent hijacking of port 80, which is the default HTTP traffic in the internet.

In this mode, clients are not required to configure their browsers to access the cache, but Squid will transparently pick up the appropriate packets and cache requests. This solves the biggest problem with caching: i.e. getting users to use the cache server.

Advantages of Interception (Transparent) Caching

- Simplified administration - The browser need not be configured to communicate to a cache server.
- Central control - The users can be made to browse via cache.

Disadvantages of Interception (Transparent) Caching

- User control – Interception caching takes control away from the user. Many users have very strong biases about caching and will actually change ISPs to either avoid it or get it.
- Impossibility for Ident string scanning and certain proxy authentication, ending with protocol violation.

FreeBSD and Interception (Transparent) Proxy

A FreeBSD default kernel configuration doesn't support NAT-ing or Port Redirection. To make this possible the kernel has to be recompiled in support of the above for which full FreeBSD source tree is required to build the kernel.

Steps to recompile kernel

Kernel Path : /usr/src/sys/i386/conf

1. Make a copy of the GENERIC configuration file in the same path with a new name.

```
# cp GENERIC SQUID
```

2. Add or enable the following options in the new configuration file. (Example - SQUID)

```
options     IPFIREWALL
options     IPFIREWALL_VERBOSE
options     IPFIREWALL_DEFAULT_TO_ACCEPT
```

Steps to recompile kernel (cont…)

```
options     IPDIVERT
options     IPFIREWALL_FORWARD
options     DUMMYNET
```

3. Change to /usr/src directory

```
# cd /usr/src/
```

4. Compile the Kernel

```
# make buildkernel KERNCONF=SQUID
```

5. Now install the new kernel

```
# make installkernel KERNCONF=SQUID
```

Squid Configuration

1. Download the latest source from squid-cache.org.

2. Extract the source

3. Move to squid source directory

```
# cd  </path/to/squid-source>
```

4. Configure and install the squid

```
# ./configure
# make
# make install
```

Default, squid will be installed in /usr/local/squid

1. Once the installation of kernel is complete, locate the squid configuration file (squid.conf) and configure it in transparent mode.

```
# http_port 3128 transparent
```

2. Creating the cache directory

```
#squid –z
```

3. Start the squid

```
#squid –D
```

4. To forward all the request through 3128, configure IPfirewall for port redirection

```
# ipfw add 100 fwd SQUIDIP,3128 tcp from any to any 80 recv IFACE
```

Troubleshooting

Booting issue

FreeBSD maintains the latest kernel in the default directory location ***/boot/kernel/***, and the previous kernel in the ***/boot/kernel.old/*** directory location. After completing the first successful kernel compilation, the /boot/kernel.old/ directory holds the generic version of the kernel. It's good practice to make a copy of it.

What if the kernel doesn't boot?

If your new kernel does not boot or fails to recognize your devices, do not panic! FreeBSD has a mechanism for recovering from incompatible kernels. When the boot menu appears, choose the kernel you want to boot from at the FreeBSD boot loader. Select the "Escape to a loader prompt" option. When prompted, type ***unload kernel*** and then type ***boot /boot/kernel.old/GENERIC***, or the filename of any other kernel that boots properly. When reconfiguring a kernel, always have a kernel on hand that really works.

Change the default deny rule

Default ipfw rule

```
# 65535 deny ip from any to any
```

The above rule will not allow login access via ssh or FTP. Add the following rules for remote login before recompiling the kernel.

```
# ipfw add 65534 allow ip from any to any
```

Enable the following option before recompiling the kernel

```
# options IPFIREWALL_DEFAULT_TO_ACCEPT
```

Boot with new kernel, the default deny rule will be changed to allow automatically.

```
# 65535 allow ip from any to any
```

To list all the rules

```
# ipfw list
```

To flush all the rules

```
# ipfw flush
```

To enable the ip firewall system startup

```
# vim /etc/rc.conf
  firewall_enable="YES"
```

References

ViSolve Squid Configuration Manual 2.4

ViSolve Squid Configuration Manual 2.6

ViSolve Squid Configuration Manual 2.7

ViSolve Squid Configuration Manual 3.0

Implementing Transparent Caching Using Squid

Conclusion

This document gives a high-level idea of compiling FreeBSD to implement Interception (Transparent) Proxy. The choice and method of implementation as said may vary, although other features present in the implementation may continue or be enhanced but the underlying fundamentals will be the same as those discussed here. And finally the overall success largely depends on the configuration and fine-tuning of both Squid and FreeBSD.

The choice of opting an OS or the type of proxy to be implemented solely depends upon the discretion of the user based on their network setup and business requirement.

About ViSolve

ViSolve is an open source based software consulting and support organization established in 1995. ViSolve has made significant contribution for the open source community in general and squid in particular, by assisting developers to help fix security vulnerabilities, testing, QA and support.

ViSolve is a distinct leader in providing commercial squid support for the past 10years and ranks #1 in almost all search engines in squid related keyword search.

ViSolve is an acknowledged provider of squid configuration manuals and whitepapers on various mode of squid deployment thereby helping millions of online users searching for initial help to deploy Squid. For more details on ViSolve, please visit http://www.visolve.com/index.php