NYI

The FreeBSD™ FOUNDATION

FreeBSD and Commercial Workloads: Managed Services at NYI

Executive Summary

This white paper describes the challenges associated with being a managed services provider, which include interacting with the wider Internet, ensuring high availability, recovering from data loss, and compartmentalizing systems and data. To surmount these challenges, this white paper describes several FreeBSD-based solutions: PF, CARP, pfsync, HAProxy, GEOM mirroring, FreeNAS, ZFS, rsync, and jails. Each solution has been battle-tested by NYI, an ISP headquartered in New York, whose customers include Men's Journal, Rolling Stone, and Us Magazine.

What is FreeBSD?

FreeBSD is an advanced computer operating system used to power modern servers, desktops, and embedded platforms. A large community has continually developed it for more than thirty years. Its advanced networking, security, and storage features have made it the platform of choice for numerous embedded (networking and storage) devices and many of today’s busiest web sites.

“We use FreeBSD for all of our internal and customer-facing solutions.”

- NYI

Introduction

This white paper examines FreeBSD's role in commercial workloads. We discuss the challenges associated with running an ISP, specifically managed services, and we present some of the unique solutions that FreeBSD provides. To help present this material we profile NYI, an ISP headquartered in New York. NYI provides colocation, dedicated servers, web and email hosting, managed services, turnkey disaster recovery, and business continuity solutions. It specializes in mission-critical data services for the financial, architectural, fashion, law, life sciences, media, and real estate industries.

Background

Managed services is the practice of outsourcing day-to-day management responsibilities as a method for improving operations.[1] Using managed services in the information technology (IT) sector, organizations can avoid the burden of maintaining equipment and infrastructure, thereby allowing their IT staff to focus on core business tasks instead. Managed services providers essentially offer the following benefits:

Ensure reliability by keeping networks up and running and minimizing downtime. This includes defending against malware.

Stay up to date by keeping hardware and software updated, as well as keeping pace with bandwidth demands.

Challenges and Solutions

There are a number of challenges associated with providing managed services at NYI, including interacting with the wider Internet, ensuring high availability, recovering from data loss, and compartmentalizing systems and data. Each challenge is surmounted with a FreeBSD-based solution.

Outsider Threats (or the Internet at Large)

IT security and malware threats have been steadily increasing. According to Symantec "the release rate of malicious code and other unwanted programs may be exceeding that of legitimate software applications."[2] A firewall is the de facto standard for protecting against threats from the public Internet. However, on most networks the firewall is a

[1] “Managed services,” last modified August 01, 2013, http://en.wikipedia.org/wiki/Managed_services.

[2] “Symantec Internet Security Threat Report: Trends for July-December 07,” April, 2008, p. 29.

single point of failure. When the firewall goes down, access to and from the internal network comes to a halt, effectively creating downtime for your customers. FreeBSD provides three components—PF, CARP, and pfsync—which NYI uses in tandem to keep their systems well protected with zero downtime.

The Packet Filter system (PF) is the firewall. NYI employs at least two firewalls in parallel, where one of the firewalls is the primary and the rest are the backups. All traffic passes through the primary, and if the primary ever fails, a backup will assume the identity of the primary and continue where it left off. Existing connections are preserved and traffic continues as if nothing happened. An additional benefit of this configuration is that it is trivial to do maintenance and upgrades on the firewalls without impacting the network. Simply take the firewalls offline one at a time.

The Common Address Redundancy Protocol (CARP) is what allows a backup firewall to assume the identity of the primary. The primary purpose of CARP is to allow multiple hosts on the same network segment to share an IP address.[3] Within each CARP group, the primary firewall, known as the master, holds the shared IP address. It responds to any traffic or Address Resolution Protocol (ARP) requests directed towards it. The primary firewall regularly sends out CARP advertisements and the backups listen for this advertisement. If the backups don't hear an advertisement from the primary for a set period of time, they will begin sending their own advertisements. The backup that advertises most frequently will become the new primary.

pfsync is the system that synchronizes the firewalls' state tables. This system is how a backup firewall can preserve the connections of the primary, when the primary fails. The primary firewall sends out pfsync messages containing its state information. To ensure that every backup firewall is synchronized, the backup firewalls replicate these messages and send them out too.

[3] "PF: Firewall Redundancy with CARP and pfsync," last modified May 01, 2013, http://www.openbsd.org/faq/pf/carp.html.

Figure 1[4] provides an example of a PF, CARP, and pfsync setup.

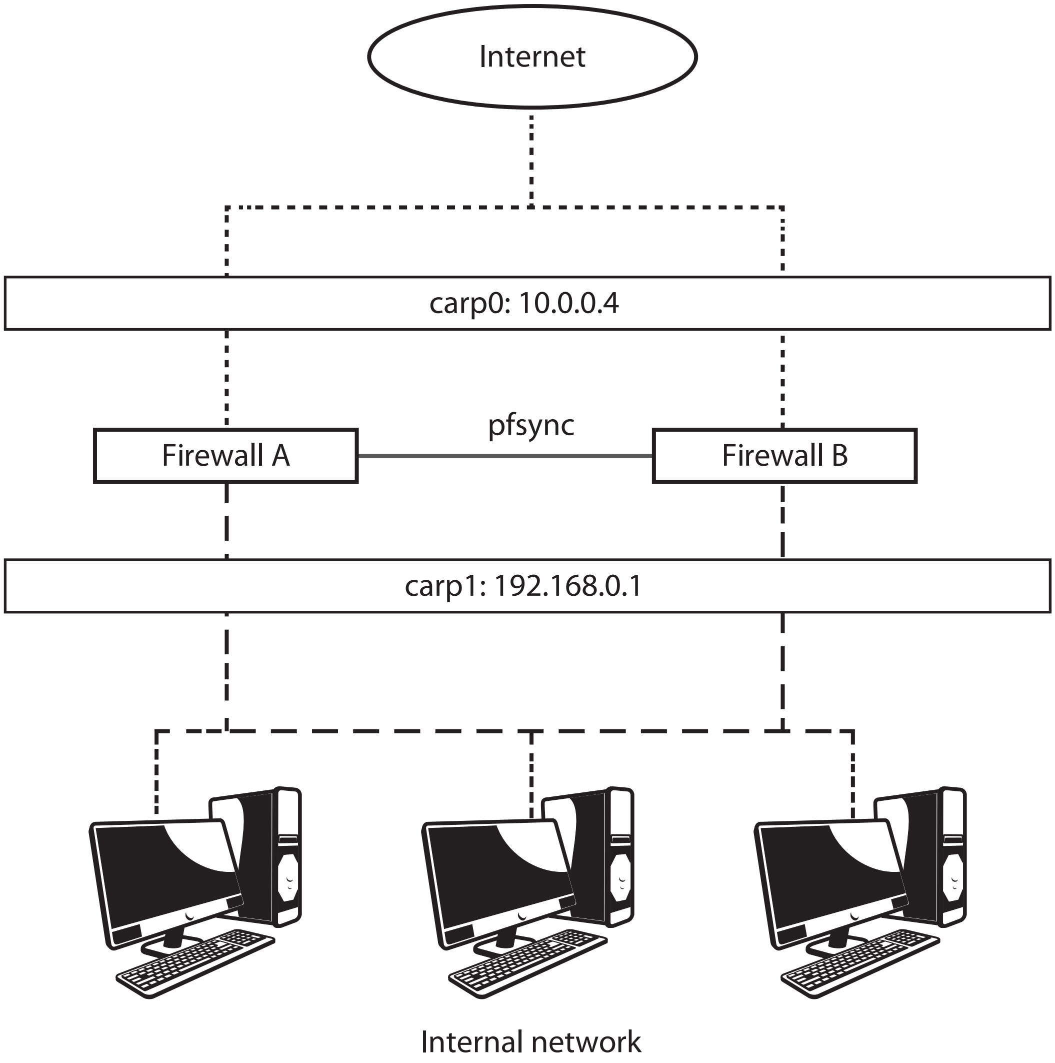

Figure 1: A basic PF, CARP, and pfsync setup.

In Figure 1, the two firewalls (A and B) each have three network interfaces. The interface on the 10.0.0.0/24 subnet connects to the external network (that is, the Internet). The interface on the 192.168.0.0/24 subnet connects to the internal network. Finally, the third interface connects the two firewalls to each other via a crossover cable, which forms a dedicated link for the pfsync messages.

[4] Figure 1 is adapted from "Firewall Failover with pfsync and CARP," accessed August 14, 2013, http://www.countersiege.com/doc/pfsync-carp/.

Figure 2[5] provides the timeline of events in a typical failover scenario.

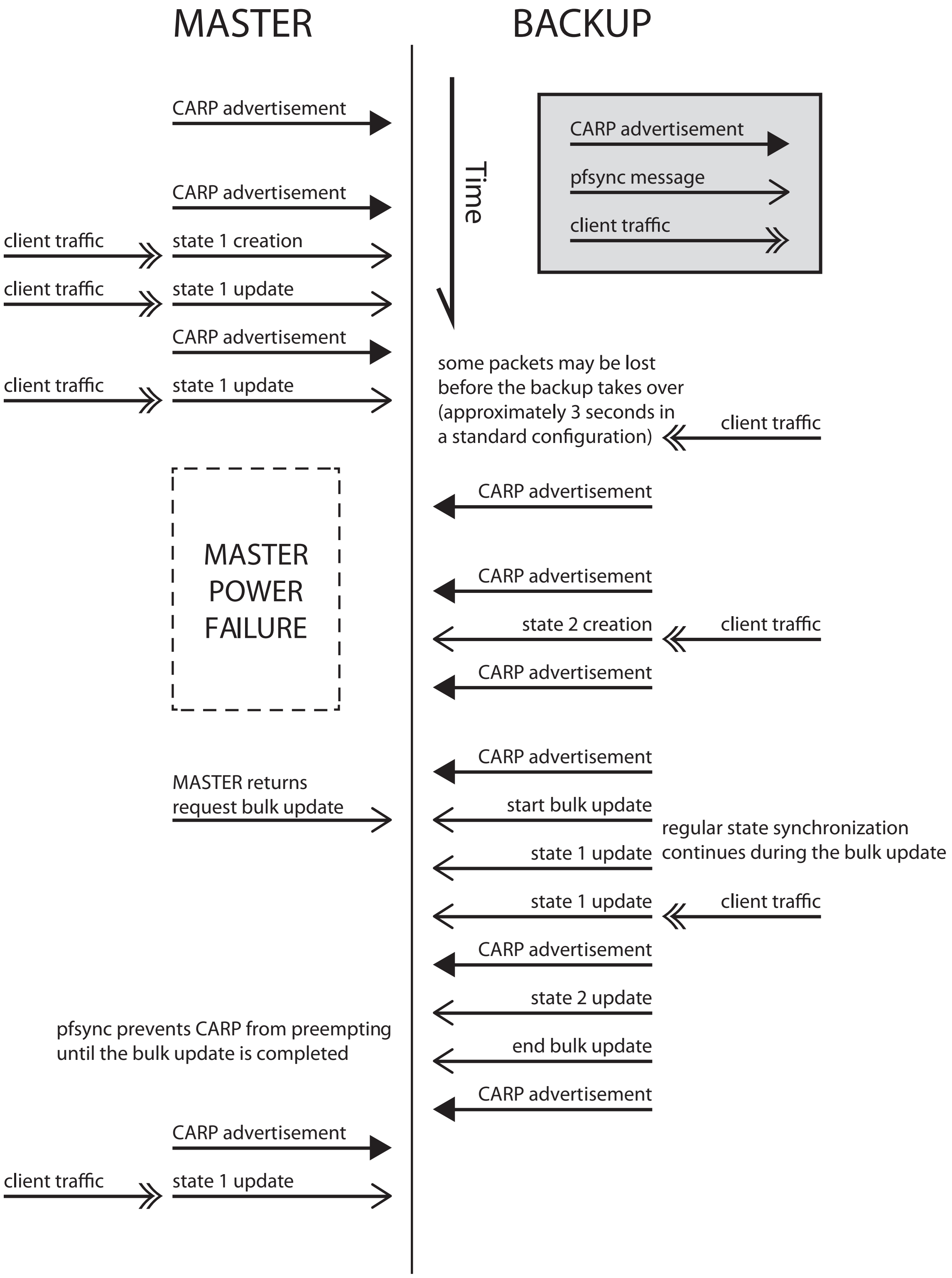

Figure 2: Timeline of events in a typical failover.

[5] Figure 2 is adapted from "Firewall Failover with pfsync and CARP," accessed August 14, 2013, http://www.countersiege.com/doc/pfsync-carp/.

High Availability

Round-the-clock operations have long been a requirement in the ISP industry. With customers spread globally across numerous time zones, any interruption of service, at any time, will affect customers, and a customer who cannot access an online system will inevitably be dissatisfied. To ensure high availability of services, NYI makes use of HAProxy, which is available in the FreeBSD ports tree, with CARP (which was discussed in the previous section).

HAProxy is a TCP/HTTP load balancer, which is used to improve the performance of web sites and web services by spreading the incoming requests across multiple web servers, thereby ensuring that no individual server is overburdened. Figure 3[6] demonstrates how HAProxy works.

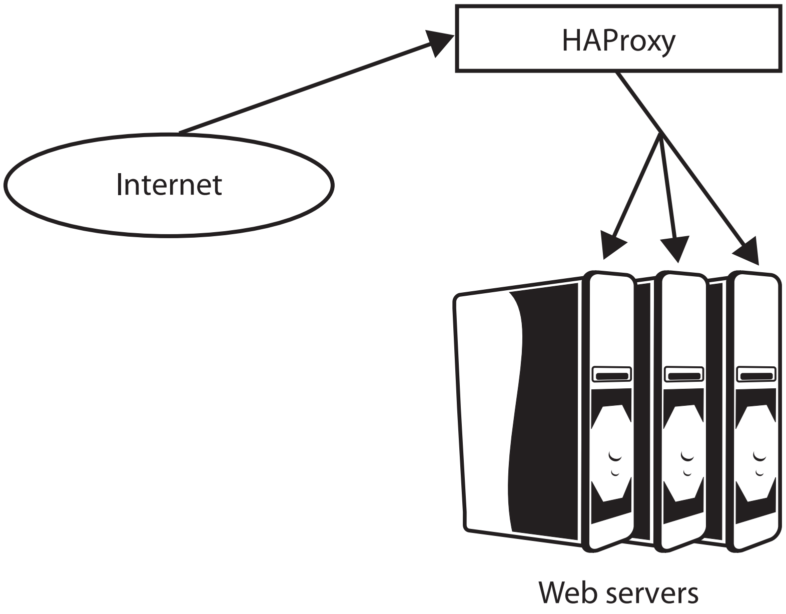

Figure 3: HAProxy in action.

In Figure 3, HAProxy accepts a request from the external network (that is, the Internet) and forwards it to the least used web server within the internal network.

CARP, as mentioned previously, is what allows a backup system to assume the identity of a primary. To ensure high availability for its three largest managed setups (Men's Journal, Rolling Stone, and Us Magazine), NYI employs a pair of machines running HAProxy with CARP. If the primary load balancing machine fails, the backup will assume the identity of the primary and take over.

[6] Figure 3 is adapted from “HAProxy,” accessed August 14, 2013, http://haproxy.1wt.eu/.

From this example, we can see that CARP can provide failover redundancy for systems beyond just firewalls. As another example, NYI employs CARP with some of its managed Internet Protocol Security (IPsec) virtual private networks (VPNs).

Disaster Recovery

Backup and data recovery have long been standard data center disciplines and are equally important for providers of managed services. Any data loss has the potential to significantly impact the profitability of a company. NYI takes advantage of FreeBSD's GEOM mirroring to mitigate this risk.

GEOM mirroring is FreeBSD's way of implementing RAID 1, which creates reliable data storage by generating an exact copy (or mirror) of a data set on two or more disk drives. When a drive fails, the data remains available because it can be provided by the other functioning drives, allowing administrators to replace the failed drive without interrupting their users.

One interesting feature of GEOM mirroring is that it can also be used to quickly clone servers. At NYI the process is as follows:

1. Remove a drive from the mirror.
2. Execute `fsck(8)` on the drive; this checks the consistency and repairs any damaged file systems on the drive.
3. Mount the drive in order to adjust any settings; mounting a drive makes it accessible through the operating system's file system.
4. Adjust settings as needed.
5. Unmount the drive.
6. Put the drive into a new server.

Steps three through five can be omitted if no settings need to be adjusted. In addition to GEOM mirroring, NYI uses FreeNAS, ZFS, and rsync to perform offsite backups in order to mitigate the risk of data loss.

FreeNAS is based on an embedded version of FreeBSD and provides an open source network-attached storage (NAS) solution. NAS systems provide data storage to other devices on a network and communicate in terms of files, rather than in disk blocks.

Figure 4[7] shows an example NAS setup.

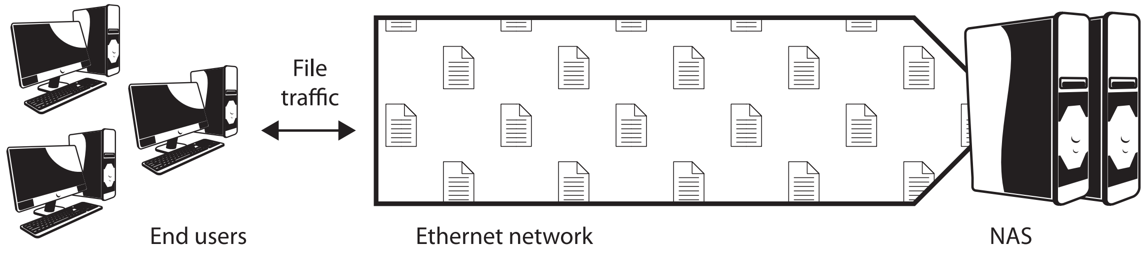

Figure 4: A basic NAS setup.

In Figure 4, the end users read and write files to the NAS system over an Ethernet network. NYI employs multiple FreeNAS machines with over 20 TB (terabytes) of storage to house offsite backups.

ZFS is the file system used by FreeNAS (and optionally by FreeBSD). Its features include support for high storage capacities, protection against data corruption, continuous integrity checking, automatic repair of data, software raid (RAID-Z), instantaneous file system snapshots, and more. In short, ZFS is designed for data integrity from top to bottom, which is desirable when managing backups.

rsync is the network protocol that NYI uses to back up a machine's entire file system offsite to a FreeNAS machine. rsync minimizes data transfer by using delta encoding, which transmits data in the form of differences rather than complete files. After the first full backup, rsync will only transfer the differences between the local and backed-up copy.

Compartmentalization

The desire to establish a clean and clear-cut separation between services, for security purposes, has always been a challenge for system administrators. Traditional Unix systems provide `chroot(2)`; however, `chroot(2)` has a number of limitations (for example, it does not defend against intentional tampering by the root user). FreeBSD has modified and improved on the traditional `chroot(2)` concept with jails.

FreeBSD jails compartmentalize the system. Each jail is a virtual environment running on the host with its own set of files, processes, users, and root user. Unlike a `chroot(2)` environment, which restricts processes to a particular view of the file system, jails

[7] Figure 4 is adapted from "Big data meets big storage," accessed August 14, 2013, http://arstechnica.com/business/2011/05/isilon-overview/2/.

restrict what a process can do in relation to the rest of the system. Jailed processes are sandboxed.[8]

Here is an example usage for jails: A small-scale managed customer of NYI, Expand the Room, requested a staging and production environment for a web site they were developing. The solution was a FreeBSD machine with two jails that were identical in every way except for IP address and hostname.

NYI also uses jails internally for hardware efficient Domain Name System (DNS) servers. For example, a FreeBSD machine may contain a jail for recursive DNS, a second jail for authoritative customer DNS, and a third jail for authoritative NYI DNS. Each of these jailed DNS servers then uses CARP within a cluster of machines to ensure failover redundancy. Figure 5 demonstrates this.

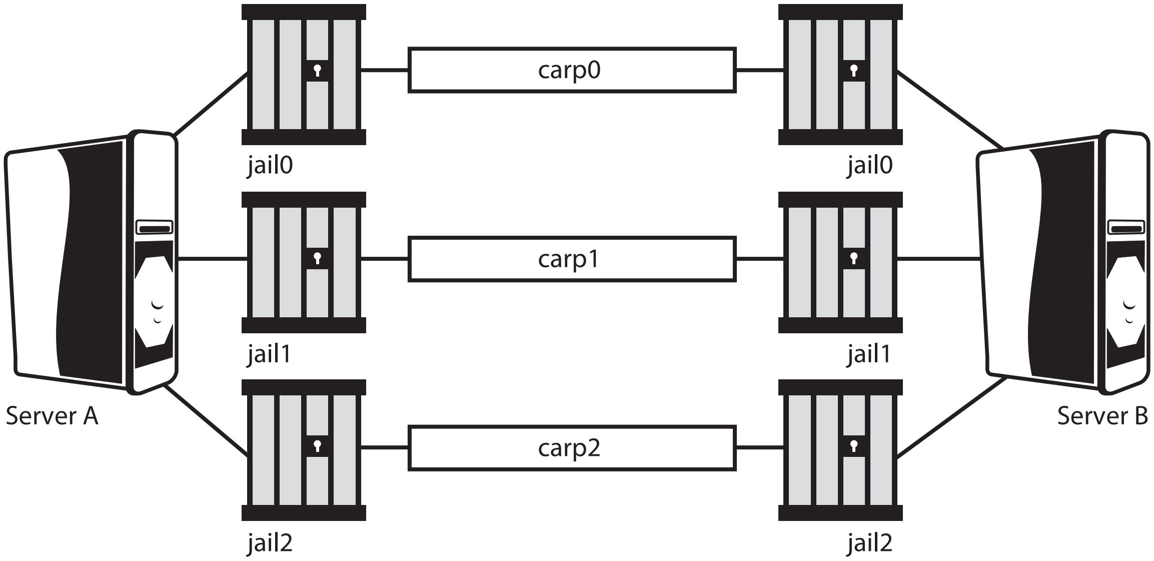

Figure 5: Jailed services with CARP.

In Figure 5, two servers (A and B) are used to provide three distinct services. Each service is contained within its own jail (jail0, jail1, or jail2) and uses CARP to ensure high availability.

Customers

Managing customer expectations is always a challenge. Customers expect and demand things to just work with (near) 100% uptime. This is exacerbated by the fact that managed services providers cannot control the client software that their customers employ,

[8] "FreeBSD jail," last modified June 08, 2013, http://en.wikipedia.org/wiki/FreeBSD_jail.

which makes compatibility an issue. FreeBSD's open source nature helps address this challenge.

For example, one of NYI's largest managed customers used a particularly buggy SSH client, which expected the challenge-response password prompt to be "Password: " (note the space). However, the prompt in FreeBSD is "Password:" (without a space after the colon) and this caused the SSH client to fail at authenticating. Since FreeBSD is open source, NYI could easily patch FreeBSD's SSH server to include a space in the password prompt, allowing their customer to continue using the client of their choice.

Conclusion

In 1996, when NYI was founded, FreeBSD was the only viable open source Unix. Today, FreeBSD continues to drive NYI for the reasons outlined in this white paper and more. Boris Kochergin, NYI's Chief Rigor Officer, had these additional things to say about why NYI uses FreeBSD:

> "FreeBSD has excellent documentation. The FreeBSD Handbook, which covers the day-to-day use of FreeBSD, is clear, concise, and provides an easy means for administrators to learn the system. FreeBSD is open source and the code is well organized, so it's easy and possible to fully understand it. Finally, FreeBSD continues the BSD legacy of empowering the Internet!"[9]

[9] The first widely-used TCP/IP implementation was from BSD.

About NYI

Established in 1996, NYI is headquartered in the heart of Wall Street. Its core services include colocation, dedicated servers, web and email hosting, managed services, turnkey disaster recovery, and business continuity solutions. NYI owns and maintains its own data centers and with its high-bandwidth connectivity partners (Zayo, Verizon Business, Optimum Lightpath, AT&T, Level 3, and GTT), NYI specializes in mission-critical data services for the financial, architectural, fashion, law, life sciences, media, and real estate industries. NYI is SSAE 16 Type II-compliant as well as being PCI and HIPAA compliant. For more information about NYI visit www.nyi.net.

About the FreeBSD Foundation

The FreeBSD Foundation is a 501(c)(3) non-profit organization dedicated to supporting the FreeBSD Project. The Foundation gratefully accepts donations from individuals and businesses, using them to fund projects that further the development of the FreeBSD operating system. In addition, the Foundation can represent the FreeBSD Project in executing contracts, license agreements, and other legal arrangements that require a recognized legal entity. The FreeBSD Foundation is entirely supported by donations. For more information about the Foundation visit www.freebsdfoundation.org.

About Joseph Kong

Joseph Kong is a self-taught computer enthusiast who dabbles in the fields of exploit development, reverse code engineering, rootkit development, and systems programming (FreeBSD, Linux, and Windows). He is the author of the critically acclaimed *Designing BSD Rootkits* and *FreeBSD Device Drivers*. For more information about Joseph Kong visit www.thestackframe.org or follow him on Twitter @JosephJKong.