BSDCan—The Technical BSD Conference

Ottawa, Canada
May 18–19, 2007

USENIX is a Gold Level Sponsor of this event.

■ ***Open Source Security Lessons***

Wietse Venema

Summarized by Julian C. Dunn (jdunn@aquezada.com)

Wietse Venema is perhaps most well-known as the author of the Postfix mail transport agent, which is arguably just as popular as Sendmail, the granddaddy of all mail transport agents. He began his talk with an amusing interlude showing how difficult it is to quantify the popularity of Postfix using Google Trends, given that the term "Postfix" has multiple definitions.

Before Postfix, Venema was instrumental in creating SATAN (Security Administrator's Tool for Analyzing Networks) and, prior to that, the widely deployed TCP Wrappers, which continues to be shipped with many open-source operating systems. He shared some of the lessons learned from his time spent in the open-source security space, not only in terms of the technology but in terms of publicity as well. For example, Venema showed several media quotations prior to the release of SATAN that claimed the tool would cause widespread destruction on the Internet. Of course, when the tool was released, no appreciable increase or decrease in system compromises was recorded, according to the various CERT organizations Venema surveyed.

Venema has had much better media relations around the release of Postfix, claiming that a single *New York Times* article heralding the release of the "IBM Secure Mailer" project (as Postfix was then known) single-handedly changed IBM's attitude toward open source, eventually convincing the company to become involved in many open-source communities such as the Apache Foundation and Linux kernel development. Venema concluded his talk by humorously alluding to the fact that even though his original motivation in writing Postfix was to provide an alternative to the complexity of Sendmail, the number of lines of code in Postfix has now exceeded that in Sendmail. However, he now considers the feature set of Postfix to be more or less complete.

■ ***Recent Improvements to the FreeBSD Ports Monitoring System***

Mark Linimon

Summarized by Bill Moran (wmoran@potentialtech.com)

Mark Linimon provided some demonstrations of his ongoing work to tame the FreeBSD ports system. The FreeBSD ports system provides a convenient method for building and installing third-party software on FreeBSD, and it currently includes over 17,000 applications. Mark has done a great deal of work creating a reporting mechanism that summarizes much of the development of the ports system so that problems can be more easily discovered and addressed. The results of his efforts can be seen at http://portsmon.freebsd.org/.

■ ***Network Stack Virtualization for FreeBSD 7.0***

Marko Zec

Summarized by Bill Moran (wmoran@potentialtech.com)

Marko Zec (http://www.tel.fer.hr/zec/) demonstrated his work virtualizing the FreeBSD network stack. By abstracting the vnet structure an additional layer, Marko was able to create completely independent networking environments within a single FreeBSD instance, each with its own IP information and routing table, thereby providing an excellent opportunity to use FreeBSD as a network research platform or to improve FreeBSD's existing jail system. A live CD is available for download from http://www.tel.fer.hr/imunes/, and Marko is working to get his improvements merged into the mainline FreeBSD source tree.

■ ***Varnish HTTP Accelerator***

Poul-Henning Kamp

Summarized by Julian C. Dunn (jdunn@aquezada.com)

Poul-Henning Kamp is a FreeBSD kernel developer who has worked on a multitude of both kernel-space and "userland" applications ranging from disk encryption to embedded systems. Lately, he has been working on the Varnish HTTP Accelerator project (http://varnish.projects.linpro.no/), which aims to provide inbound HTTP acceleration for busy Web sites such as VG (http://vg.no), the Web site for a popular Norwegian newspaper.

Kamp began by explaining why Squid, the classic HTTP caching solution, is programmed poorly. He outlined the methods in which it "fights the kernel" by trying to explic-

itly separate memory and disk storage. He denounced this methodology, saying that because the kernel provides virtual memory services there is no need for user applications to do this work. Doing so results in excessive system calls, which lowers performance. In contrast, Varnish simply allocates large objects in virtual memory and lets the kernel manage memory in the optimal way.

Varnish also maximizes performance in many other ways by using careful programming tactics, for example, by avoiding expensive text-processing operations if they can be avoided. In addition, Varnish's configuration language (VCL) is preprocessed and compiled into binary, then dynamically loaded for speed. Multiple configurations can be loaded concurrently and an interactive command-line interface (CLI) manager can switch configurations, in addition to doing other cache operations such as purging objects, retrieving cache statistics, and so on.

Future work on Varnish will see features such as edge side includes and URL rewriting added. Kamp hopes to eventually see the project moved under the auspices of the Apache Software Foundation, as there would be a natural synergy with the Apache HTTP Daemon.

■ ***FreeBSD Security Features***

Robert Watson

Summarized by Bill Moran (wmoran@potentialtech.com)

Robert Watson (http://www.watson.org/~robert/) gave an excellent overview of his ongoing work extending the FreeBSD security model, first providing an overview of ACLs (access control lists). ACLs offer an extremely flexible method of describing permissions on filesystem objects. Unfortunately, the two leading systems of ACLs, POSIX and NT, are not compatible. FreeBSD supports POSIX ACLs, but there is interest in supporting NT ACLs, since NFSv4 uses them. Robert also described the powerful security auditing tools introduced in FreeBSD 6.2. These tools are required by Orange Book and other evaluations and provide a method for fine-grained monitoring of systems. FreeBSD's audit tools are based on Solaris and Mac OS, but these tools can be extended with the concept of audit pipes, which allow the administrator to create multiple filters of audit events. Finally, Robert covered mandatory access controls (MACs), which supplement discretionary access controls (such as filesystem permissions). If you've worked with SE Linux, the MAC framework will feel familiar.

■ ***Portsnap***

Colin Percival

Summarized by Bill Moran (wmoran@potentialtech.com)

Colin Percival described the road he took to developing portsnap, an updating tool specifically designed for the FreeBSD ports tree. Because of his role as FreeBSD security officer, Colin started writing portsnap to make the distribution of port updates more secure, but he also managed to significantly improve the speed and bandwidth usage by writing a customized compression program called bsdiff that is aware of byte substitutions. I was also interested to hear Colin describe existing technologies, such as HTTP pipelining and DNS SRV records, that are largely unused but could solve many problems plaguing the Internet.

■ ***How Open Source Projects Survive Poisonous People***

Brian Fitzpatrick and Ben Collins-Sussman

Summarized by Bill Moran (wmoran@potentialtech.com)

Being a member of several groups (not all of them open source software groups), I decided to attend the lecture by Ben Collins-Sussman and Brian Fitzpatrick on how groups can survive poisonous people. Ben and Brian took turns covering various aspects of four tenets: comprehension, fortification, identification, and disinfection. I found their insights enlightening, but the highlight was when they asked the room if anyone knew what "bikeshed" referred to, only to find that not only did everyone in the room know, but the man who popularized the phrase, Poul-Henning Kamp, was sitting in the back of the room.

■ ***Failover and Load Balancing with pfSense***

Scott Ullrich and Chris Buechler

Summarized by Chris Buechler (cbuechler@gmail.com)

I was one of the presenters for this session and a co-founder of this project. pfSense is a FreeBSD-based firewall distribution using OpenBSD's pf packet filter, with a Web interface for configuring all aspects of the system. This presentation focused on the failover and load balancing functionality available in the system.

Five main topics were covered: CARP, multi-WAN failover, policy-based routing and failover, DNS failover, and incoming and outgoing load balancing.

CARP allows for seamless hardware failover, to accommodate hardware failure, or firewall maintenance and upgrades without loss of connectivity. Typical CARP configurations and deployments were discussed.

Multi-WAN failover allows the use of multiple Internet connections, and upon failure of a connection, the remaining WAN connection(s) can be automatically used to maintain connectivity. Common deployment scenarios were illustrated and discussed.

Policy-based routing and failover enables routing of traffic based on IP protocol, TCP or UDP port, and source or destination IP, among other possibilities. Upon failure of the preferred routing destination, backup destinations can be utilized. Generally this is configured in combination with multi-WAN. Common configurations were given.

DNS failover combines with the previously mentioned functionality to update your public DNS records upon failure of a WAN connection. This enables the multi-WAN functionality to be used for inbound access from the Internet, with automated failover.

Incoming and outgoing load balancing combines with multi-WAN and policy-based routing to allow multiple Internet connections to be load-balanced for outgoing traffic, or for inbound traffic from the Internet, it allows for load balancing between multiple servers (for example, a Web server farm). Some of the deployments in production today were illustrated and discussed.

At the end, we logged into a production firewall cluster and showed how this functionality is configured and works in a real-world installation.

Overall, the feedback we received was mostly positive, but in hindsight we tried to pack entirely too much into the allotted time frame. We also assumed that those attending a presentation on some of the advanced pfSense functionality would know about the basic functionality, which was mostly correct, but there were a decent number of people who weren't familiar with the project at all. Because of time constraints, we could only give a high-level overview of the previously mentioned functionality, and we couldn't leave users with specific information on how to configure the various deployments discussed. In hindsight this presentation would have been much more useful to our attendees if it were one of the longer tutorials rather than a one-hour presentation. In the future, we'll need to watch our scope or go for a longer session.

■ ***UTORvpn: A Cross-Platform OpenSource SSL VPN Implementation***

Russell Sutherland

Summarized by Chris Buechler (cbuechler@gmail.com)

This session was presented by Russell Sutherland, a network engineer at the University of Toronto. He began by going over the various common types of VPN implementations in production environments today. The problem with most VPN solutions in wide deployment today is cost or lack of cross-platform support. The University of Toronto needed a VPN solution that worked with numerous platforms and would scale to thousands of users, but didn't cost a fortune. Enter OpenVPN.

OpenVPN is an open source SSL VPN solution that has an open source client available on numerous operating systems, including Windows 2000/XP and newer, FreeBSD, NetBSD, OpenBSD, Mac OS X, Linux, and Solaris. As such, it met their requirements for cross-platform compatibility. Its authentication and authorization capabilities also were able to tie into the university's existing Kerberos authentication and LDAP authorization systems.

Russell logged into the university's Web interface where users can sign up for OpenVPN access, to show how they have automated the build of the Windows installer on the FreeBSD server using NSIS, so each user has a customized Windows installer available with the appropriate certificates and configuration built in. He also showed the management interface and the type of reporting and statistics they gather from the log files, and how they manage the certificates, all via a custom-written Web interface.

I'm already a happy OpenVPN user, but this gave me some ideas on how to get more out of it. Russell did an excellent job of introducing people to OpenVPN and showing how it can be used in large deployments.