SEE TEXT ONLY

FreeBSD is an operating system well-known for its advanced network stack and security. It provides thousands of third-party software packages through the ports collection, making it possible for expert users to have a great firewall running with a minimum of investment. Other users with less expertise or even no experience with FreeBSD, and in some cases more money, prefer to run their security solutions using commercial products.

pfsense® and Security

By James Dekker, Luiz Otavio O. Souza, and Renato Botelho

pfSense® software exists in the middle of these two worlds. pfSense is an open-source firewall/router distribution running on top of FreeBSD. All configuration occurs via an easy-to-use web GUI, and many functions are provided using other open-source applications (e.g., Unbound, Strongswan, OpenVPN, etc.). pfSense software provides users with the main features offered by most commercial firewalls, but with the primary advantages of costing less and being open source. Open-source software can be audited and fixed when its behavior or security is in doubt. The open-source model also allows anyone to fix broken code, while closed source can only be fixed by the vendor.

pfSense software was born in 2004, as a fork of m0n0wall. Since then it has been continuously developed and improved, making it a trusted product used by hundreds of thousands of users around the world. Currently there are over 550,000 active installations of pfSense software.

Netgate is the company behind pfSense and employs core developers, making it possible to have people working on and supporting it every day of the year. Netgate offers 24/7 support and other professional services to help users design their network from scratch, review existing configurations, and migrate environments from most known commercial products to pfSense software to meet the customer's requirements.

Improvements in New Versions

After a long period of being based on FreeBSD 8, pfSense software version 2.2, based on FreeBSD 10.3, was released January 23, 2015. We have continued tracking FreeBSD closely, and as a result, the 2.3 version, released April 12, 2016, was based on FreeBSD 10.3, (released March 28, 2016). pfSense software 2.3 brought a large number of new features and improvements. The most significant changes were a complete rewrite of webGUI using Bootstrap and the underlying system, including the base system and kernel, being converted entirely to FreeBSD pkg.

Having pfSense software use pkg brought three large benefits. First, pfSense software updates were historically distributed as a monolithic image that was extracted on top of the currently installed version. There were no records of any system files. Using FreeBSD pkg made it possible to distribute minor upgrades easily and made the process of testing and validating a new upgrade much faster.

Second, using pkg to upgrade every piece of pfSense software makes it possible to detect whether or not the kernel is being upgraded. With version 2.4, this information is used by pfSense-upgrade to decide if the system needs to be rebooted or not. If the answer is no, it will use reroot as an alternative and the system will restart all services and be back online faster using the same running kernel.

Finally, starting with version 2.1, pfSense used PBIs to distribute additional packages. This was a good choice at the time because each PBI contained all required libraries and other binary dependencies so as to not pollute the pfSense core with additional files. With pkg, all primary and additional packages are built together in the same environment using poudriere. Poudriere takes care that dependencies are tracked correctly and that all unnecessary files are removed when a package is uninstalled, leaving the system clean.

The conversion of the WebGUI to Bootstrap made the user interface more modern, easier, and intuitive, while leaving the code both more readable and easier to extend, resulting in a renewed level of contributions to the project. The number of pull requests submitted to main pfSense repository at github more than doubled from 384 to 955 during 2.3 webGUI conversion time.

We continue to track FreeBSD source closely. Our next major release, pfSense software version 2.4, which is in BETA as we write this article, will be based on FreeBSD 11. With this version, 32-bit architecture and the nanobsd model were discontinued, but we now offer images for ARM platform, more specifically to Netgate SG-1000 (uFW). The old bsdinstaller has been replaced by bsdinstall, the same installer used by FreeBSD, which brings important features such as UEFI and ZFS support.

FreeBSD 11 also brings an updated 802.11 stack, which will bring pfSense software numerous improvements when used as a WiFi client or access point. Appliances offered in the Netgate/pfSense store with native wireless are

fully compatible with FreeBSD 11 and are ideal for applications including research and penetration testing.

During 2.3 and 2.4 development, we drastically reduced the number of patches we carried against the FreeBSD source code to accommodate features required by pfSense users. Changes that were of interest to upstream FreeBSD users were contributed to FreeBSD, while other patches were removed and pfSense was reworked to have the same result without patching FreeBSD. As a result of this work, pfSense software version 2.4 is currently running on top of a FreeBSD tree that is much closer to stock. This yields the advantage that we can now move to a new FreeBSD major release more quickly than ever before. In the past, and under different leadership, the project took two years to move from FreeBSD 8.1 to FreeBSD 8.3 and another 20 months to move from FreeBSD 8.3 to FreeBSD 10.1.

Packages

One of the major benefits of pfSense is the ability to install additional packages that provide extra functionality such as Snort, Suricata, and Squid. In the past, the number of available additional packages became unworkable as the maintainers of some of them lost interest, leaving some packages to stagnate.

During the pfSense 2.3 release process, while we migrated from the old pfSense-packages repository to a FreeBSD ports-based repository, we decided to clean up and remove packages that have been deprecated upstream, no longer have an active maintainer, or were never stable.

Remaining packages were converted to the new Bootstrap model and reworked to become available as a FreeBSD pkg. Following the release of version 2.3, the community came together to migrate and maintain some packages that had been removed. After review, some of these packages were accepted and reintroduced to the list of available packages.

Captive Portal

The Captive Portal function in pfSense software allows securing a network by requiring a username and password (or only a click-through entered on a portal page). If authentication is used, this can be performed with native user management within pfSense or an external authentication server such as an Active Directory, LDAP or RADIUS server.

The pfSense firewall is implemented principally by pf, one of the three packet filter implementations available in FreeBSD. When Captive Portal is enabled, it uses ipfw, another packet filter implementation available in FreeBSD. Until pfSense software 2.3, we had a heavily modified version of ipfw to make it possible for Captive Portal to work with multiple instances. During the move to FreeBSD 11, due to a huge rewrite of ipfw on FreeBSD, we decided it was a good time to remove the big ipfw patch and make Captive Portal work with native ipfw, or at least with just a few modifications.

All Captive Portal pieces that interact with ipfw were reworked, and, as a result, we were able to eliminate one big extra patch that led us closer than stock FreeBSD. Necessary changes done during this work will be reviewed and upstreamed as soon as 2.4 is released.

Security

pfSense software's main goal is to be a security appliance, and because of that, we take it seriously. In 2014, we announced Security Advisories containing all security-related issues that are discovered, along with their corresponding CVE IDs.

A significant number of security flaws related to the webGUI are caused by the use of HTTP GET. To mitigate this attack, vector pfSense software 2.4 has converted many parts of the webGUI to work only via HTTP POST.

Code Review

In 2016, Netgate enlisted the help of Infosec Global to conduct a top-to-bottom, post-commit audit of pfSense software version 2.3.2. Conducting an independent code review helps improve the quality of the product.

For this project, we provided Infosec Global with the Netgate XG-2758 1U Security Gateway Appliance with pfSense software 2.3.2 installed with a default production configuration.

Infosec Global scores threats on a bottom-up percentage scale, with 0% being a perfect score and 100% being most critical. As indicated in the audit report, pfSense 2.3.2 scored an outstanding 1%, which included concerns that were already mitigated during the timeframe of the audit process with the release of pfSense software ver-

sion 2.3.2_p1. Other issues raised do not apply to the firmware version reviewed. The complete report is available for review at https://www.netgate.com/assets/ISG_Netgate_2016.pdf

OpenVPN

With the release of pfSense software version 2.4, OpenVPN has been upgraded to version 2.4.0. That is a significant upgrade that includes support for a number of new features, including support for AEAD ciphers such as AES-GCM. We also added support for Negotiable Crypto Parameters (NCP) to control automatic cipher selection between clients and servers.

Let's Encrypt

In February 2017, a new package called ACME was made available for releases greater than 2.3.2 in Package Manager. This package interfaces with the Let's Encrypt project to handle the certificate generation, validation, and renewal processes.

Let's Encrypt is an open, free, and completely automated Certificate Authority from the non-profit Internet Security Research Group (ISRG). The goal of Let's Encrypt is to encrypt the web by removing the cost barrier and some of the technical barriers that discourage server administrators and organizations from obtaining certificates for use on Internet servers, primarily web servers. Most browsers trust certificates from Let's Encrypt. These certificates can be used for web servers (HTTPS), SMTP servers, IMAP/POP3 servers, and other similar roles that utilize the same type of certificates.

Certificates from Let's Encrypt are domain validated, and this validation ensures that the system requesting the certificate has authority over the server in question. This validation can be performed in a number of ways, such as by proving ownership of the domain's DNS records or hosting a file on a web server for the domain.

By using a certificate from Let's Encrypt for a web server, including the webGUI in pfSense, the browser will trust the certificate and show a green check mark, padlock, or similar indication. The connection will be encrypted without the need for manually trusting an invalid certificate.

Translations

While pfSense has always been a project of contributions from the community, efforts to translate pfSense started many years ago, but were never 100% complete for any language. We were also never able to provide a central place or good toolchain for contributors of translations for pfSense. For pfSense software version 2.4, we decided to bring this subject back and added pfSense software to the Zanata translation website. Once we had announced this setup, it was a short time until we received a full translation into Spanish, Russian, Norwegian, Chinese (Simplified, China), Chinese (Taiwan) and Bosnian, with incredible progress in German and many others. If you want to see pfSense software translated to your native language, join us!

Contributing to Upstream

Our main upstream is FreeBSD. Both FreeBSD source and ports repositories are the bases used to build and run pfSense software. During the past two years, Netgate engineers have contributed 155 commits to the FreeBSD src repository and 35 commits against the FreeBSD ports collection.

But FreeBSD is not our only upstream; we also run other software provided by third-party projects such as Strongswan, OpenVPN, and dpinger. pfSense developers consistently strive to submit fixes for upstream projects every time a bug is found or a new feature is implemented. This is part of an ongoing effort to improve the software for the entire user base.

pfSense Software in the Cloud

With today's cloud platforms, you can easily extend an on-premise IT environment into the cloud in a manner similar to setting up and connecting to a remote branch office. While these services provide excellent tools for connecting VPNs and setting up access lists, they do not provide full visibility and manageability into the endpoints, nor do they secure the connectivity of critical cloud services.

The Netgate-provided images for pfSense software on AWS and Azure deliver advanced routing, firewall, and VPN functionality for your public cloud infrastructure at a lower cost than other solutions. The pre-built pfSense AWS and Azure images have features similar to both the pfSense hardware appliances and the VMware Certified pfSense available from Netgate.

ARM Support on pfSense 2.4

We had just released pfSense 2.3 when the prototypes of our first ARM system landed on our desks. It was then we knew that the ARM support on FreeBSD had evolved so quickly between 10 and 11, to the point where it was extremely difficult to port many of the new features and fixes. It became clear that FreeBSD 11 would be the only reasonable option and we knew that moving on was the best way to provide the optimal ARM experience.

Since then, Netgate has upstreamed all the code developed, tested, and used daily on the SG-1000. A few contributions specific to the SoC used in this platform include:

- numerous platform and Ethernet fixes to increase system stability and performance
- dual Ethernet support
- management of the integrated Ethernet switch

The introduction of a new, less forgiving ARM architecture to our development environment also helped us spot a few misbehaving sections of code that would have passed unnoticed on a more forgiving architecture. The outcome of fixing these is code correctness and better overall portability.

Today, the ARM support in pfSense provides a seamless experience for any user of this newly introduced architecture. Everything was carefully worked out to provide a reliable, smooth, and incredibly intuitive experience, to the point that you will likely forget you are using a tiny ARM device (unless you manage to remember where this little device lives—or hides—in your office).

Thanks to the pfSense pkg support for updates, the updates on the SG-1000 are bliss and the whole system is updated from the WebGUI (FreeBSD kernel and base, pfSense base and packages), which makes the management of these devices terrific.

The SG-1000 has significantly increased the number of FreeBSD ARM devices in the world. As of this writing, we are actively working on our next ARM system, a dual core ARM with more speed and more hardware features.

Since pfSense began as a fork of m0n0wall, it has continued to evolve into a product that drives the wedge deeper and deeper into the enterprise market space, while still meeting and exceeding the needs of the SMB, prosumer, and home user. What once was a project to provide an extensible and easy-to-use software product for an embedded hardware system has grown into a product that will now run on hardware from embedded ARM devices smaller than a credit card to high availability systems that can take up 2U of rackspace and even larger clustered configurations. This highly versatile, increasingly stable, and secure software product would not be possible without the massive support and involvement of both the community and the resources of Netgate. It's only with the combination of these that we are able to continue pushing the envelope in terms of what is possible for not only FreeBSD, but also, as time will tell, a return to our roots in providing an extensible, high-throughput, easy-to-use software product for embedded devices. ●

RENATO BOTELHO started using FreeBSD on version 3.2 after a friend gave him an installation CD and said, "Install it and never look back." His first contributions to FreeBSD ports tree in 2004 led to his becoming a committer the next year. Likewise, his involvement with pfSense in 2009 was soon followed by his employment at Netgate. He lives in a small city in Brazil with his wife, son, daughter, and two dogs. When he is not in front of a computer, he enjoys running, CrossFit, beer, and barbecues—not necessarily in that order.

JAMES DEKKER is a diehard fan of pfSense and a Production Support Analyst for Netgate. He enjoys securing systems and the networks that connect them. He has been working with *nix systems for some time, primarily focused on security, networking and systems administration. He has been using pfSense since version 2.0 and providing support to the community since 2.1. When James is not working or breaking things in his office, he can be found on the beach, fishing, or working on the family ranch. He lives on the Treasure Coast in Florida with his wife, dog, and cat.

LUIZ OTAVIO O. SOUZA is a software engineer at Netgate and a FreeBSD committer.