



new faces

of FreeBSD

BY DRU LAVIGNE

This column aims to shine a spotlight on contributors who recently received their commit bit and to introduce them to the FreeBSD community. This month, the spotlight is on **Eric Badger** and **Toomas Soome**, who became src committers in July and August 2016.

Tell us a bit about yourself, your background, and your interests.



ERIC

Eric: I'm a software engineer at Dell Compellent, based in Minnesota, USA. I work on the Platform OS team in charge of the FreeBSD operating system running on Dell Storage controllers. I've been doing this for a little more than two years. I've learned a ton about FreeBSD working at Dell, partly from solving problems that arise in the OS and partly from the various smart people that I get to work with.

I'm interested in lots of things; nearly anything low level-ish can capture my interest. The things I've actually had time to work on have been mostly oriented towards networking, debugging tools (like gdb or DTrace), and storage devices. I just bought a BeagleBone Black for playing around with FreeBSD on something other than amd64, and have had lots of fun doing that.

Outside of computers, I like to play music (I like most kinds and have played with bands here and there) and study languages (Spanish is my principal interest at the moment).

Toomas: I was introduced to Unix as a student at the University of Tartu in Estonia, back in about 1992. We had a mix of different systems back then: SCO, SunOS, Irix, later Ultrix, NetBSD, FreeBSD, and eventually also Linux, just to name a few. At that time I was working for the university as a system administrator and that time meant a lot of compiling, software integration, and so on as all the (free) software was distributed as source.



TOOMAS

For a fun fact, we actually started with www using the CERN sources for both the server and browser.

Later on I was working as pre-sales engineer for a local Sun Microsystems reseller, which meant less compiling but still a lot of integration, architecting, and deep diving into almost every layer of IT infrastructure.

For other interests, snowboarding in winter and skydiving in summer. Although, due to time issues, those activities have been somewhat diminished.

How did you first learn about FreeBSD and what about FreeBSD interested you?

Eric: I encountered FreeBSD at my first computer job at the Supercomputing Institute while a student at the University of Minnesota. My first boss was the network admin, and she used FreeBSD for network management tools and jump boxes onto the switch management network. I wanted to learn as much as possible about how things worked, so I added a FreeBSD box to my home network of Linux boxes (mainly old computers from friends and family). That first FreeBSD machine was my DHCP server and also ran tftpd and NFS for serving up files to PXE-booted machines.

The division between base and ports definitely stood out to me. I ran mostly Debian and CentOS then, which meant that sometimes third-party software wasn't available in the repos, or wasn't available at the version I wanted. FreeBSD ports meant I could have relatively bleeding edge software on a relatively stable base. Since working with FreeBSD as a developer, I've come to appreciate the fact that kernel and userland all live in the same tree. This often makes understanding and working with

things that cross the boundary easier. I've also come to appreciate the first class treatment of source code. FreeBSD makes it easy to get the system source and rebuild parts of it on your system. Ports make it easy to hack on third party software that needs a little help to build on FreeBSD.

Toomas: Cannot remember the year—it was a long time ago when I was still with the university, but we had a few instances of FreeBSD, and NetBSD. But BSD-based systems have always been interesting due to their roots and now even more as one of the Unix brand providing free software and really supporting technology exchange.

How did you end up becoming a src committer and which part of src do you work on?

Eric: I submitted a couple of patches that had been accepted to the tree and started a few discussions, mainly on -current. I think I earned my stars mainly by debugging and characterizing tricky problems, however, rather than a large number of patch submissions.

Mostly I've touched code under sys/kern at this point. I wouldn't say I've developed any strong specialty yet, though I have a growing list of things that I'd like to get into a committable state, including improved support for FreeBSD on KVM, some SR-IOV work, and a few other things.

Toomas: The roots are in a different operating system, due to my own roots to technologies related to Sun Microsystems. I was poking around with Illumos-based systems and eventually found myself investigating ways to improve boot loader and related topics. While working with different options, I found myself investigating the FreeBSD loader and ended up porting it to Illumos. That work has reached the stage where we are very close to implementing an actual integration. Quite obviously, Illumos has its own needs for its boot loader and so gradually those needs got addressed and some changes and improvements were implemented in the port. Once you have done some changes, you soon find that it is reasonable to offer some of those back upstream. Eventually, I guess, people with commit rights got the idea to invite me to the ranks of FreeBSD developers, which was quite a nice surprise for me.

So the area I'm working on is the boot loader and related topics. Some results are already integrated into FreeBSD, hopefully for more good than

damage, but there are still many issues to investigate regarding how is the best way to improve the boot loader.

How has your experience been since joining the FreeBSD Project? Do you have any advice for readers who may be interested in also becoming a src committer?

Eric: My experience has been good. It's kind of like having a second job, and in that respect it can be demanding. I've only a couple of commits at this point, but still spend a lot of time playing with things and trying to understand them. But it feels like time well spent, since I'm usually having fun and have the opportunity to contribute to a major open-source operating system.

My advice to a potential future committer: if you find something that doesn't seem to be working correctly in FreeBSD, don't stop until you understand why. I find that there are few better ways of getting to understand something than debugging a problem you've personally experienced. Working with a company that uses FreeBSD also goes a long way. At Dell Compellent, there are hundreds of FreeBSD machines being pounded on by our test group every day, and any OS issues come through my group. That has provided a tremendous learning opportunity.

Toomas: I have only a positive experience. The most valuable part is really about information exchange, to have more ways to learn how the different components are related, how teams are related, and what are the driving forces behind a decision. Most of the success of the particular project is really not about how well some piece of code is written, as there is always space for improvements, but how well it will fit in with the rest of the system and if you can somehow manage to bring in some feature at the right time to provide the most value for the system or other teams or people.

As for advice, you may have all the dreams and training, but before the door opens and you actually make your step through, nothing will matter. Once you have done the first step, and can see the training you have received is actually working and giving results, then you are probably sold and the hard work towards the goals you have set will really start. So, make that first step and check out if you like what you are experiencing. ●