Musings

RIK FARROW

Rik is the editor of *;login:*.
rik@usenix.org

We've got a jam-packed issue this time, so I thought I'd cut right to the chase. Instead of musing, I will just tell you why I picked a particular author, set of authors, or topic for your edification. The reason, in some cases, is to attempt to disabuse you of long-held beliefs.

We begin with Conway et al., who researched the issue of file system fragmentation in Linux [1]. The researchers used a Git workload and showed that all of the popular Linux file systems suffer performance degradation, even on SSDs. After just 100 pulls, hard drive performance was halved, while it took a bit more activity, 800 pulls, to reduce SSD performance by 25%. The authors do have an agenda: their own file system design, BetrFS, doesn't manifest this problem and is faster than other Linux file systems in many cases.

Disk drive manufacturers have been hiding information from us. For many years (decades?), they have converted the logical block address into the physical location of their firmware's choice, so file system designs that attempt to prevent fragmentation really don't have much of a chance. But there's more: hard drives with capacities greater than two terabytes use device managed shingled magnetic recording (SMR). The Aghayev et al. article describes changes they made to ext4 that improve performance not just on SMR drives, but on any hard drive.

Ganesan and company delve into another of our popular myths: that having redundant copies makes our distributed data secure. They researched a number of distributed file systems and databases, induced read or write errors, and discovered some really terrible things. That is, I think it's bad when having multiple copies means that the bad copy gets used to overwrite the good copy, or when a failed read crashes the system. You might want to read this even if you don't use any of the eight distributed systems they tested.

Shvachko and Chen take another look at HDFS. The single-point-of-failure NameNode has long been an issue, and their Giraffa system replaces the NameNode with a distributed naming and block management system. HopsFS, from the Niazi et al. paper at FAST '17 [2], also confronts this issue, although using a different approach.

Programming

Andy Rudoff, who wrote about persistent memory for *;login:* way back in 2013, has written about the PMEM libraries currently available for use with Linux and Windows systems. These libraries focus on using PMEM as memory-mapped files (`mmap()`), but Rudoff also tells us about some other useful libraries and explains how best to use these new devices. Oh, and during Rudoff's FAST '17 tutorial on this topic, he kept waving around an Intel-Micron 3D XPoint device. I actually held this device, and can tell you that **it's real**. PMEM will change the way many systems work in profound ways.

Graeme Jenkinson has written a great article explaining Rust, a programming language with a focus on type safety. Will Rust save the world from buggy code? Probably not, as most people are addicted to whatever they currently use. But Rust is still really worth looking at.

I interviewed Eric Allman, the author of both `syslog` and `sendmail`. Eric has traveled the open source road, a journey more often painful than rewarding for him.

Security

Peck et al. have written what may be the final article in the series on BeyondCorp. BeyondCorp has been a journey away from traditional, trusted, internal networks and into a Zero Trust network design [3]. This article is about the paths taken, ones that couldn't have succeeded without the long process of gaining the trust of the users of Google's networks, learning what could easily be migrated, and how to migrate the more unusual services, over several years.

Hunt et al. have written about the Ryoan sandbox, a system designed to run within Intel SGX enclaves on a distributed system. Their model provides assurance that the expected software is running in the sandbox, that the data sent through the sandbox remains private, and that the sandbox doesn't leak much information through covert channels. You can also learn a lot about how SGX enclaves work by reading their article or their OSDI '16 paper [4].

Kuppaswamy, DeLong, and Kappos challenge people to find flaws in their design, *Uptane*, for providing secure firmware updates for automobiles. Cars are loaded with computers, and many new cars are also network-connected, so having a secure method for installing updates that works both within cars and for car manufacturers is more important than ever.

Radia Perlman reprises her talk at LISA16 about Bitcoin. Radia compares the design and capabilities of Bitcoin to other systems, past and present. Bitcoin design has attracted lots of attention and investors, but is it really any better than other cryptographic systems?

Jos Wetzels spoke at Enigma 2017 about embedded system security. Wetzels researches IoT security issues, and in this article he describes some of the issues facing both researchers and developers of software for embedded systems. In short, things don't look promising, but policy and regulation could set a reasonable baseline for the IoT, just as RoHS [5] already restricts the use of certain hazardous substances in electrical and electronic equipment.

Columns

Dave Beazley explores the new `pathlib` module that appears in Python 3.4 and later. Dave had written about `pathlib` several years ago, and he demonstrates some of the things you can do with that module, as well as things you can't do. Then Dave explains both how `pathlib` improves pathname manipulation, but also problems that arise with incompatibilities between `pathlib` objects and other functions that accept pathnames.

David Blank-Edelman explores Perl modules from the air. David takes us on a tour of some of the modules that come with a stock install of Perl, a very different approach to his usual Perl examples.

Dave Josephsen gets excited about compression. Dave tells us about Gorilla, a time series database that has been open sourced by Facebook and is designed to keep the most recent data in memory. In particular, Dave explains some of the tricks used to compress datestamps in time series.

We have a new columnist this issue. Jeanne Schock, who has worked as a system administrator and now focuses on change, incident, and problem management, has written about root causes and their relations to problems. Seems obvious, right? Well, read on, because it's not that obvious.

Dan Geer and Jon Callas have written about the impact of revealing a nation-state's exploit toolkit [6]. You'll have to read their column, as it's an interesting exercise in game theory.

Robert Ferrell considers how modern backup systems *should* work, then takes pokes at software subscriptions and advertising that targets your dreams. You might think you know what that means, having watched ads with people driving fancy cars sitting next to the mate of their dreams. But that's not what Robert means.

Sometimes it seems to me that things are changing much too quickly to keep up with. Then I notice that Tim Feldman wrote about SMR drives in 2013 [7], just as Andy Rudoff was writing about PMEM the same year. Four years later, and we are just now seeing the effects of the concepts discussed back then, and while there are millions of SMR drives in use, there aren't any 3D XPoint cards available on the open market. As William Gibson quipped in 1993, "The future is already here—it's just not evenly distributed."

References

[1] A. Conway, A. Bakshi, Y. Jiao, Y. Zhan, R. Johnson, B. C. Kuszmaul, and M. Farach-Colton, "File Systems Fated for Senescence? Nonsense, Says Science!" in *Proceedings of the 15th USENIX Conference on File and Storage Technologies (FAST '17):* https://www.usenix.org/system/files/conference/fast17/fast17-conway.pdf.

[2] S. Niazi, M. Ismail, S. Haridi, J. Dowling, S. Grohsschmiedt, and M. Ronström, "HopsFS: Scaling Hierarchical File System Metadata Using NewSQL Databases," in *Proceedings of the 15th USENIX Conference on File and Storage Technologies (FAST '17):* https://www.usenix.org/system/files/conference/fast17/fast17-niazi.pdf.

[3] D. Barth and E. Gilman, "Zero Trust Networks: Building Trusted Systems in Untrusted Networks," SREcon17 Americas: https://www.usenix.org/conference/srecon17americas/program/presentation/barth.

[4] T. Hunt, Z. Zhu, Y. Xu, S. Peter, and E. Witchel, "Ryoan: A Distributed Sandbox for Untrusted Computation on Secret Data," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16):* https://www.usenix.org/conference/osdi16/technical-sessions/presentation/hunt.

[5] Wikipedia, "Restriction of Hazardous Substances Directive," last modified on March 24, 2017: https://en.wikipedia.org/wiki/Restriction_of_Hazardous_Substances_Directive.

[6] I. Thomson, "That CIA Exploit List in Full: The Good, the Bad, and the Very Ugly," *The Register*, March 4, 2017: https://www.theregister.co.uk/2017/03/08/cia_exploit_list_in_full/.

[7] T. Feldman and G. Gibson, "Shingled Magnetic Recording: Areal Density Increase Requires New Data Management," *;login:*, vol. 38, no. 3 (June 2013), pp. 22–30: https://www.usenix.org/system/files/login/issues/1306_login_online.pdf.

Letter to the Editor

Hi folks,

I just read the interview with Amit in *;login:* while I'm on the road to AsiaCCS. Great interview, and it's nice to see renewed interest in security for embedded devices.

Tock sounds interesting and I'll definitively check it out.

The fun part is that at AsiaCCS I'll present our work on enforcing memory safety for TinyOS [1]. We have worked on porting a CCured-like type system to nesC and enforce memory safety for a set of embedded devices at low overhead.

Embedded devices have unique advantages such as mostly static allocation, a well known stack depth, and a bunch of other interesting features that can be used to enforce strong protections, mostly statically, only falling back to a runtime check when absolutely required. In addition, there's usually a single task and dedicated resources, so we can leverage all available slack for security mechanisms.

Coincidentally, we also have an upcoming paper at Oakland [2] on protecting embedded devices using a privilege overlay. Embedded devices often run bare-metal. Our idea was to deprivilege all instructions and then, based on a static analysis, enable privileges on only a few locations and instructions. The MPU allows a dynamic configuration of these privilege overlays and enables quick switches.

It's amazing to see the renewed interest in protecting embedded systems, and I'd love to talk as we're continuing to work in that area!

Cheers,
Mathias Payer

References

[1] https://nebelwelt.net/publications/files/17AsiaCCS2.pdf.

[2] http://nebelwelt.net/publications/files/17Oakland.pdf.

ENIGMA

A USENIX CONFERENCE

JAN 16–18, 2018

SANTA CLARA, CA, USA

CALL FOR PARTICIPATION NOW OPEN!

PROGRAM CO-CHAIRS

Bryan Payne,
Netflix

Franziska Roesner,
University of Washington

SECURITY AND PRIVACY IDEAS THAT MATTER

Enigma centers on a single track of engaging talks covering a wide range of topics in security and privacy. Our goal is to clearly explain emerging threats and defenses in the growing intersection of society and technology, and to foster an intelligent and informed conversation within the community and the world. We view diversity as a key enabler for this goal and actively work to ensure that the Enigma community encourages and welcomes participation from all employment sectors, racial and ethnic backgrounds, nationalities, and genders.

Enigma is committed to fostering an open, collaborative, and respectful environment. Enigma and USENIX are also dedicated to open science and open conversations, and will make all talk media freely available on the USENIX web site.

See the complete CFP at www.usenix.org/enigma2018/cfp

usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION