EDITORIAL

Musings

RIK FARROW

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

Cost externalization is usually applied to the use of the environment as a dumping ground for industrial waste. Instead of having to pay for cleaning up sulfur and nitrogen dioxides from burning coal, companies passed those costs along to humans and the world. Turns out, open source software has a parallel problem.

Many of you would not have been alive to experience the '70s, and too young to be bored by the '80s. I'm not referring to the culture wars but rather to the acid rain that was eating holes in tree leaves, ruining crops, and hurting wildlife [1]. The yellow smog was particularly bad when combined with summer heat and ozone, limiting visibility in big US cities to one hundred meters on some days.

In the 1990s, the US Congress passed laws forcing coal burners to *scrub* their emissions. That made burning coal more expensive, continuing to do so even in 2020, but eventually made the air a lot cleaner [2] by removing nearly half of the sulfur and nitrogen dioxides from coal burning plants' exhausts.

Moving costs off of your company's balance sheet appears to be a fine idea, whether you are communist or capitalist. You can read a short blog entry about cost externalization [3], but all you really need to know is that some companies have artificially lowered their production costs.

I started out by stating that OSS has a similar problem, but I don't mean air pollution. Many companies use OSS without contributing to its maintenance or creation. They have externalized the costs of programming the software they need to run their businesses to OSS developers—people whom they usually do not pay.

Let's take a particularly toxic example: the Equifax hack of 2017 [4]. Equifax was using the Apache Struts software as part of the web front-end they used to make money. Equifax, like the other credit agencies, collected data on individuals and families and sold that to potential creditors. In Equifax's case, attackers took advantage of the Struts framework to invade the Equifax network, and stole data for over 147 million people. The vulnerability they used had been patched in March, while the hack began in July.

I hope that my comparison appears fair to you—that using OSS without contributing to its support is a form of cost externalization. If Equifax had been actively supporting Struts, I believe they would also have been very aware of the vulnerability and would have patched it.

Equifax will be fined and forced to "repay" those whose data had been stolen—to the tune of hundreds of millions of dollars, maybe [5]. Seems like crime does pay, or perhaps externalizing your costs to society works well enough most of the time.

The Lineup

I was inspired to write about cost externalization after reading Dan Geer and George Sieniawski's article about paying for the maintenance of OSS. It appears that externalizing costs extends to even tiny code snippets, such as the 11 lines of JavaScript that millions of programmers had been using.

But that's not where we begin in this issue. An award-winning paper about the reliability of enterprise SSDs forms the basis for the first feature article. Maneas et al. used data provided by NetApp to examine four different failure modes seen in SSDs. Interestingly, the failures over time of enterprise SSDs are very different from those of hard drives.

Jeff LeFevre and Carlos Maltzahn explore a way to leverage Ceph, the distributed storage system, to move computation to the location of data. SkyhookDM takes advantage of Ceph's design to distribute data across Ceph's Object Storage Devices so that work, such as SQL queries, can be executed on the system where data resides instead of having to copy all data to a single server first.

I interviewed Natalie Silvanovich, part of Google's Project Zero team. Natalie, the first woman on the team, talked about the techniques she uses while bug-searching, having another woman join the team, and things you should know or learn about if you want to learn more about finding exploitable bugs.

Dick Sites volunteered to write an article demonstrating some uses of his kernel monitoring software, KUtrace. KUtrace provides much finer observations of CPU activity while the CPU executes kernel code. Sites describes four activities where the kernel is wasting a lot of CPU cycles that likely occur commonly enough to be serious issues.

Marianne Bellotti writes about how misaligned incentives result in bad software design. Most people who have heard of Conway's Law know that it describes how designs mirror organizational structures, but Bellotti uncovers a different facet of the law: programmers are incentivized to *stand out,* and that often results in championing their own additions to code that no one else can support.

Alex Hidalgo considers how best to use service level objectives as a tool in decision-making. Hidalgo expresses concerns about how SLOs are becoming buzzwords, when SLOs and SLIs can be used to create more system reliability.

Steve Ross and Todd Underwood take a look at using ML in support of SRE. Both engineers support ML and have often been asked to use ML as part of the support infrastructure. The authors explain machine learning and then point out serious issues with applying ML to SRE tasks.

Laura Nolan focuses her SRE column on decision-making, making that the theme of this issue. Nolan, however, disparages the current culture of complacency that encourages ignoring potential problems until they become crises. Nolan uses the response to the coronavirus pandemic in process as I write this as an example of the crisis/complacency dynamic.

Dave Josephsen, who lives in self-quarantine by choice, continues his exploration of BPF scripts. In particular, Dave explains the `biolatency` script and how it integrates with the block I/O (the *bio*) portion of the Linux kernel.

I've mentioned Geer and Sieniawski's column already as having inspired my musings about externalization. Their focus is actually on how widely OSS is used by enterprises while few take care to actively support the software they use, as they would with commercial software that they pay for.

Mark Lamourine has a container focus this issue, reviewing books about Docker, microservices, and containers. Mark joined the decision-making crowd with reviews of related books.

Robert G. Ferrell was inspired by the failed foray, related by Ross and Underwood, to have ML take over SRE. Robert spins this a bit differently, as he instead discusses how humans and AI will become competitors.

Peter Norton and Chris McEniry didn't write for this issue.

In the world we live and work in, market forces are supposed to rein in cheaters—those who sell bad products or take advantage of our culture to cut costs. In reality, market forces seem to encourage cheating, even when companies that do this get caught in the process. Oil companies are still supported by tax breaks, while companies like Theranos thrive for a while until the illusion they created via marketing dissipates.

And perhaps we should give Equifax a break. After the attack, people feared that identities would be stolen, bank accounts drained, and false tax forms filed. Instead, it turns out the Chinese hackers were to blame [6]. So instead of being worried about identity theft, we should be worried about Chinese intelligence operatives using our personal data to blackmail us.

Oh, well. I bet the people at Equifax weren't thinking about that when they designed their web front-ends using someone else's code.

References

[1] "Effects of Acid Rain": https://www.epa.gov/acidrain/effects-acid-rain.

[2] N. S. Rastogi, "Whatever Happened to Acid Rain?" *Slate*, August 2009: https://slate.com/technology/2009/08/whatever-happened-to-acid-rain.html.

[3] J. Whitehead and T. Haab, "ECON 101: Negative Externality": https://www.env-econ.net/negative-externality.html.

[4] L. H. Newman, "Equifax Officially Has No Excuse," *WIRED*, September 14, 2017: https://www.wired.com/story/equifax-breach-no-excuse/.

[5] Z. Whittaker, "FTC Slaps Equifax with a Fine of up to $700 Million for 2017 Data Breach," TechCrunch, July 22, 2019: https://techcrunch.com/2019/07/22/equifax-fine-ftc/.

[6] B. Krebs, "U.S. Charges 4 Chinese Military Officers in 2017 Equifax Hack," Krebs on Security, February 10, 2020: https://krebsonsecurity.com/tag/equifax-breach/.