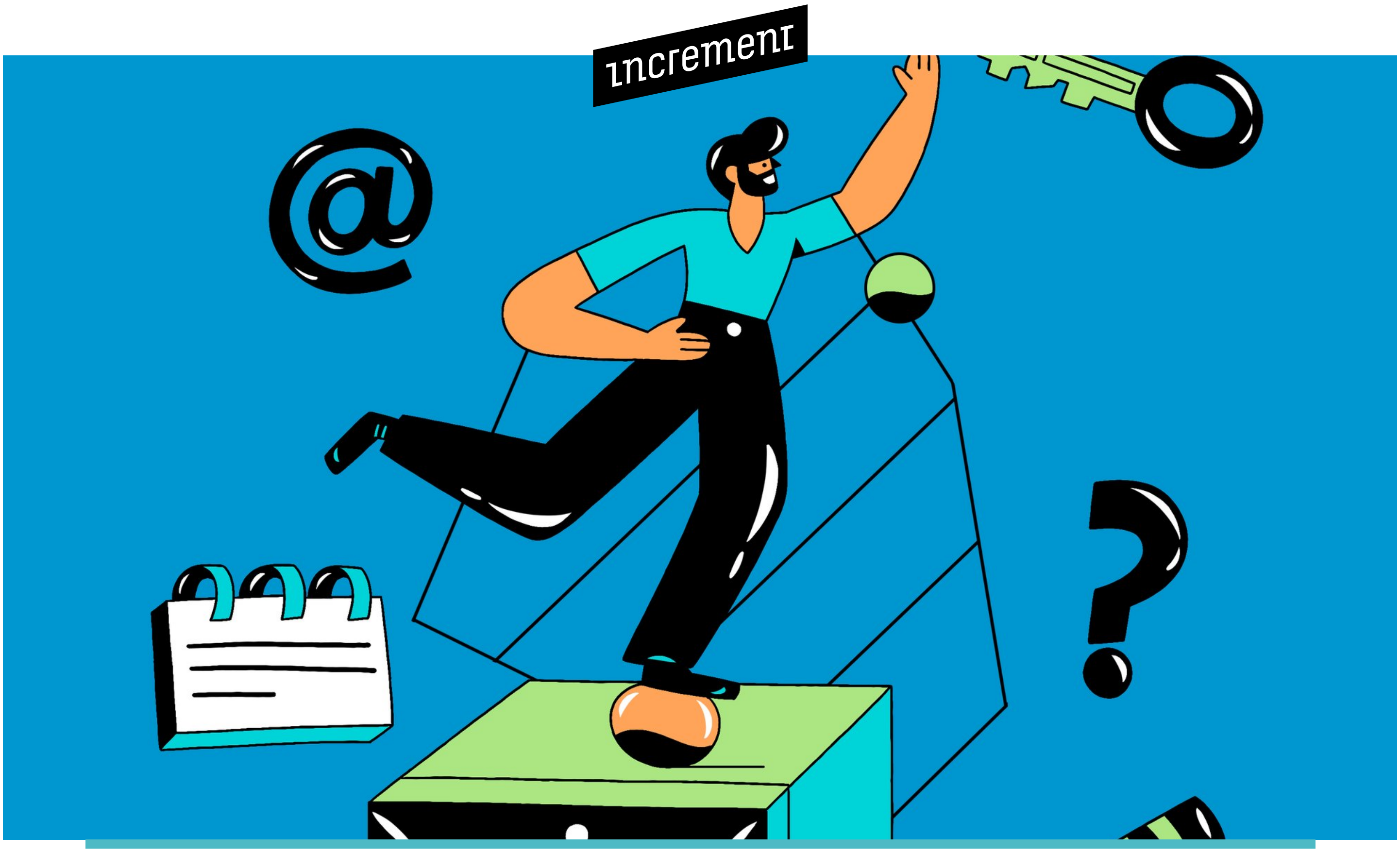
increment



EMILY CAIN

Why don’t we follow password security best practices?

From broken password change pages to conflicting best practices, users face a host of challenges when making password decisions. How can we remove the obstacles and help them use the systems we build securely?

PART OF

ISSUE 7
OCT 2018
Security

For a lot of people, following password security best practices is like flossing our teeth. We know what we *should* do, and on the occasions the topic comes up we feel anxious and guilty, but most of the time we simply don’t think about it. There are a number of issues that prevent people from following best practices, including poor usability of individual sites’ prevent people from following best practices, including poor usability of individual sites login interfaces, contradictory advice from experts, habits left over from pre-digital systems, and the overwhelming vastness of our modern digital lives.

Many of the reasons for our insecure password practices have nothing to do with general competence with technology, so even those of us who create software for a living can make these mistakes. In July 2018, a malicious update made it into the npm package ESlint, which was then automatically downloaded by many of the web developers using npm to manage their dependencies. The postmortem revealed that the developer responsible for the offending code had been hacked after reusing their password across several sites.

As a technology educator, I create instructional materials on how to use application monitoring software; many of my colleagues and much of my audience create software for a living (and I used to as well). In this piece, I'll explore the lessons I learned from setting up a password manager, an experience that has broader implications for the ways we build and talk about secure software. I hope to ground these lessons in empathy for users who, like me, might have known they weren't doing their best to secure their passwords, but felt overwhelmed by the prospect of changing their methods. I'll further explore the mistakes technical people can make by sharing the experience of my friend Nic, a software developer whose well-intentioned password decisions wound up compromising his bank password and resulted in $1,000 stolen from his account.

> People use technology to do the things they need to do; if your best practices make it more difficult to do those things, people won't follow them.

There are several understandable reasons why people don't pick strong passwords and store them consistently. For one, a lot of the advice out there is contradictory; people are generally advised to use either words interspersed with special characters ("p@ssw0rd") or long strings of dictionary words ("correcthorsebatterystaple"). They're asked for real information about their lives as a password reset mechanism — mother's maiden name, birthday, first pet — even as social media tracks and displays that information for the whole world to see. Then they're faced with login systems that enforce rules that contradict these approaches (banning dictionary words, enforcing maximum lengths, and so on).

To quote the Twitter account @SwiftOnSecurity, "If you don't make a system usable and [illegible] secure, the user will make it usable and insecure.” People use technology to do the things they need to do; if your best practices make it more difficult to do those things, people won’t follow them. Throwing contradictory advice at people and blaming them for failing to follow it is a major usability problem. How can we give better advice and make sure people are equipped with the tools and knowledge they need to follow it?

Password antipatterns

In software, we use the term “antipattern” to describe a practice that arises in response to a common problem, but causes its own set of problems. It allows technologists to understand the reasons behind the deficient practice and look for better responses.

Password reuse (as seen in the npm package debacle) is one common security antipattern. Writing down passwords by hand is another. Searching “password notebook” on Amazon yields thousands of results for paper notebooks dedicated to storing passwords. After the disastrous false missile warning in Hawaii in January 2018, a photo circulated online of the state’s Emergency Management Agency office, showing a password written down on a sticky note on one of the computer monitors. While recording passwords in some way can be necessary for people to remember them all, if someone finds your password notebook or spots your sticky note, then they’ve got your passwords. And if you lose the notebook or forget to take it with you when you need it, it’s useless.

Instead of writing their passwords down, many people heed the common security advice of creating high-entropy passwords, and use a personal algorithm to generate and remember them. This sort of algorithm may be a holdover from pre-digital forms of authentication—when you have to remember a handful of combination locks or keypads, it’s not such a bad idea to use a birthday, an address, or the year of some significant event. But there are two problems with this strategy in the internet age. One is the high number of accounts we have—an average of 27 per person, according to one 2016 survey. Most of these will have password length and complexity requirements, meaning that both the number of passwords and the difficulty of remembering an individual password are exponentially higher. The other problem is that, depending on the type of attack, password cracking tools can attempt anywhere from thousands to billions of possible passwords per second. In contrast, guessing a lock’s combination takes a few seconds per attempt. (I’ve opened a keypad lock by observing that certain numbers were worn down and guessing their order; I wouldn’t have been able to do this if the keypad had been new and I’d had to guess from every possible permutation of all 10 digits.)

While a personal algorithm theoretically avoids password reuse, it still gets people into trouble reuse, it still gets people into trouble.

While a personal algorithm theoretically avoids password reuse, it still gets people into trouble. My friend Nic told me how that strategy led to his bank account being hacked and his money being stolen: He'd been "using the same password for the past 12 years in different combinations." He'd combine old addresses, personal trivia, and numbers, using the same formula across accounts. "I was incrementing the numbers to change the password," he said. "I was fooling myself into thinking that was secure." If an attacker learns one permutation of your password formula, there's a good chance they can guess the others.

Former National Institute of Standards and Technology manager Bill Burr, who wrote the 2003 document that encouraged many of the well-known password creation practices in use today, has since said he regrets that advice. It's led to confusion and frustration as people are advised (and sometimes required by password validation rules) to create hard-to-remember passwords interspersed with numbers and special characters. And because everyone has been following the same advice to make their passwords more secure, the types of passwords that advice tends to produce are well known, and therefore less secure.

Some people have proposed alternate rules to generate secure and easy-to-remember passwords, like the xkcd comic advising a combination of four dictionary words. But I'm inclined to agree with Docker security lead Diogo Monica, who advises against relying on these tricks and algorithms (emphasis mine):

> **There is no need [to teach] users how to choose good passwords.** Everyone knows what a good password looks like, we just can't memorize unique, strong passwords for every single online service out there. With the advent of password managers, the large majority of all passwords should just be randomly generated, and replaced with a single password that provides access to all the others.

Password managers

For most people, a password manager solves these problems. It provides good-enough security, and is convenient enough that people will actually use it. Password managers enable us to achieve several goals:

Generating passwords that consist of random strings of letters and numbers, which are unlikely to be guessed by brute-force attacks, attacks using dictionary words, or attacks using previously compromised passwords.

Storing a unique password for each account, so that if one account becomes compromised, the damage is contained.

Encrypting the individual passwords using a master password, so that—unlike a plain text file or handwritten note—the data will be useless to anyone who doesn't have the master password.

Ensuring passwords are always on hand and easy to access, via a browser plugin or smartphone app.

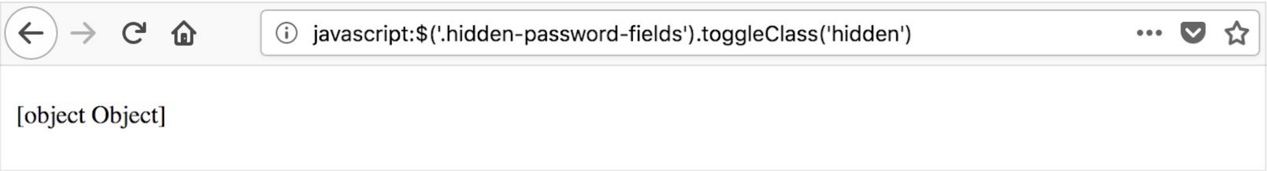
For me, setting up a personal password manager fell into the category of things I knew I "should" do. I'd remember it whenever I read about a password breach in the news, then I'd shove it back into the guilt-tinged procrastination pile. Why hadn't I done it yet? Would it be difficult? Would it be able to fully replace the clunky system I'd become accustomed to over the years? (I finally enabled two-factor authentication on my Gmail account a couple of years ago; I was using that account as a makeshift password manager, resetting my passwords on other accounts every time I needed to log into them.)

Spring cleaning for passwords

After talking to Nic, I began the process of finding and changing my passwords. I started by opening up my security preferences in Firefox to view my saved logins. This turned out to be an overwhelming experience, similar to going through boxes of old things I'd forgotten about. Sorting through the artifacts of my digital life was like finding a time capsule of all the years I'd spent online, and I found myself falling down unexpected rabbit holes. It didn't help that some of the websites' login or password change features were broken, leading to further distractions, or that some site interfaces bombarded me with lists of mostly irrelevant notifications, making it difficult to quickly complete a single task and move on.

I first logged into LinkedIn, where I was met with a broken post-login page. Then I found myself distracted and overwhelmed by months-old messages. I started responding to them all, catching myself just in time to remember that I was trying to accomplish a specific task.

Another troublesome account was the BoingBoing store *(oh, hey, I never started that course bundle I bought — FOCUS!)*, which featured a perfectly normal account management page with a completely nonfunctional password change link:



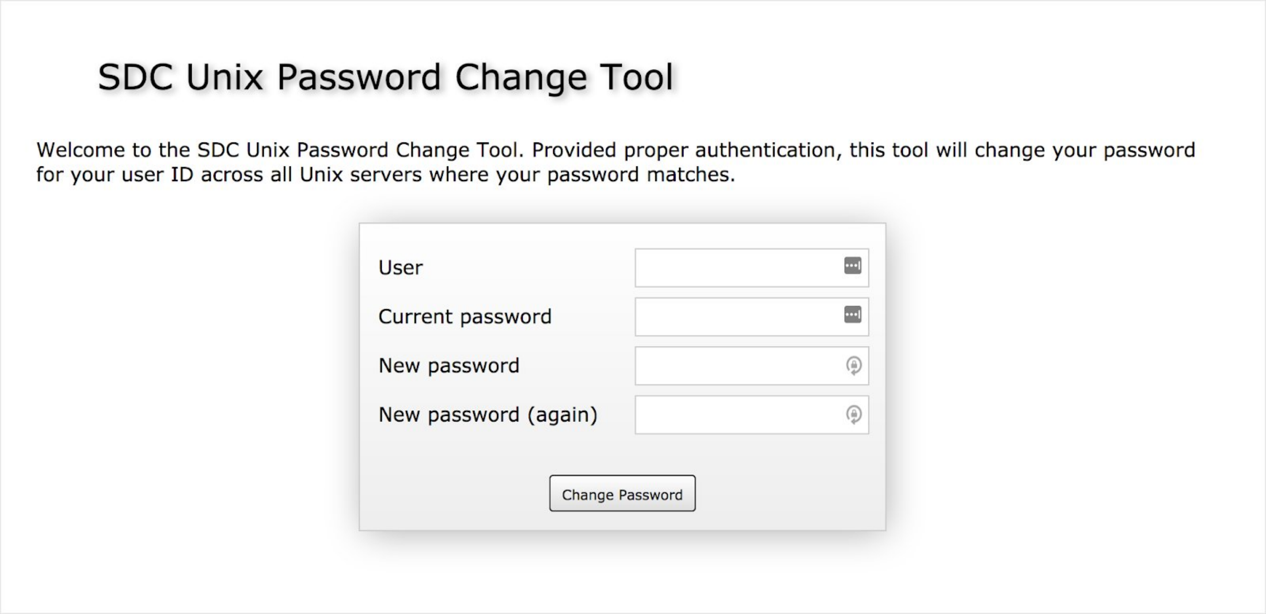
```
javascript:$('.hidden-password-fields').toggleClass('hidden')
```

```
[object Object]
```

When I contacted Support about this, their advice was to “use Chrome.”

Perhaps the most inscrutable behavior came from my state’s Department of Human Services site. I’d dealt with the bureaucratic maze of social services a few years ago after losing my job, and their online services proffered a similarly discombobulating experience. I went to the URL Firefox had saved for me, and clicked a link called “password change utility.”

This seemed straightforward enough:



SDC Unix Password Change Tool

Welcome to the SDC Unix Password Change Tool. Provided proper authentication, this tool will change your password for your user ID across all Unix servers where your password matches.

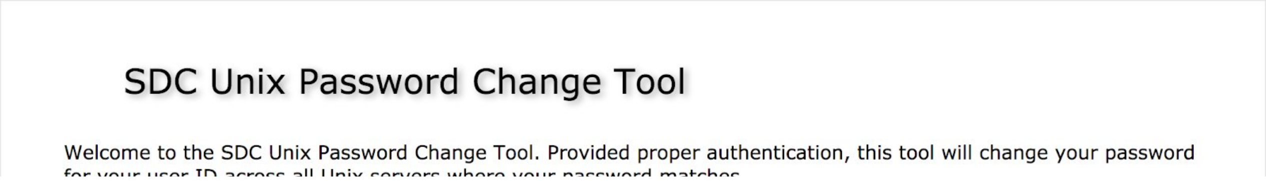
User

Current password

New password

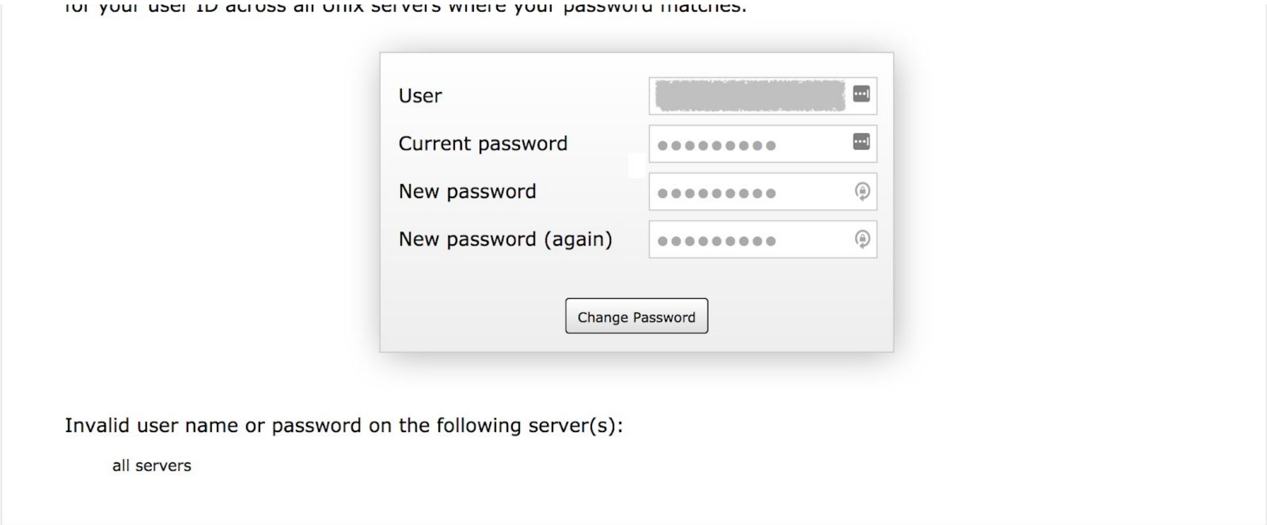
New password (again)

...or not:



SDC Unix Password Change Tool

Welcome to the SDC Unix Password Change Tool. Provided proper authentication, this tool will change your password for your user ID across all Unix servers where your password matches.



for your user ID across all Unix servers where your password matches.

User [redacted]

Current password ●●●●●●●●●

New password ●●●●●●●●●

New password (again) ●●●●●●●●●

Invalid user name or password on the following server(s):

all servers

My past experiences with these particular state agencies did not leave me confident that there would be any point in attempting to contact a technical support person. I gave up.

Even the websites that didn't present any particular errors served as a drain on my time and attention. Going through this process with usability in mind gave me a lot more sympathy for anyone who sees setting up a password manager as an insurmountable chore. While the effort was ultimately worth it, I can see why someone who's not familiar with the benefits of password managers would get overwhelmed and simply give up.

All in all, I put 16 logins in my password manager, which is accessible from my laptop and my smartphone. I'm sure there are other accounts I've forgotten, but when I go to use them in six months or a year, I won't have to repeat the cycle of indefinitely resetting my passwords. I was lucky that only three passwords were troublesome to reset, and that two of those problems were easily resolved. It took some time, but the peace of mind I get from knowing that I now carry my passwords in my pocket—and that they're harder to crack—is worth it.

Takeaways for technologists

This process was hard enough for me, someone who is familiar and confident enough with technology to find workarounds for buggy websites. It's on us, the people who create the software systems people rely on every day, to remove any obstacles we can to using these systems securely. This includes creating password change interfaces that work, that are easy to use and easy to find. It also means making sure we give people straightforward, usable advice—which, to me, means recommending password managers to just about everyone.

And, perhaps most importantly, it means approaching users and their behavior with empathy, working to understand their needs and meet them where they are.

ABOUT THE AUTHOR

Emily Cain is a programmer, writer, and software educator. She creates instructional videos and articles on software monitoring for New Relic University. Her other projects include a Twitter bot that creates fictional medications, a Glitch project to teach children to make websites using dog memes, and writing on sites such as the O'Reilly Media blog and dev.to.

@data_bae

ARTWORK BY

Valeria Alvarez

behance.net/valerialvarez

TOPICS

Essays & Opinion

CONTINUE READING

2 | Cloud

DAVID J. LUMB

The U.S. Government's long