1$_Y0ur_P@$$w0rd_Cl3v3r?

Scott Reichelt

Writer's Comment: Studying computer science has given me an interesting insight into the universal experience of what is called "computer rage." We all have laptops, smartphones, and internet-enabled toasters. And we all, occasionally, do insane things—like curse at our electronics. When I find myself in close proximity to someone on the verge of this type of outburst, I get an odd look: an eye that oscillates between accusatory and "Help Me!" Unfortunately, my understanding of the internet's extremely complicated infrastructure only serves to make my own "computer rage" slightly more informed than the average user's. When my UWP 104E professor asked me to write a paper that "communicates a technical topic to the general public," I knew this was my opportunity to convey the experience of "computer rage" from the computer scientist's perspective—knowing in excruciating detail what the problem is, but being powerless to do anything about it. When the inevitable moment comes that you enter your password incorrectly, I hope my paper helps you direct your rage at those responsible.

Instructor's Comment: In my Writing in the Professions: Science course, one assignment requires the writer to explain a highly technical subject so that any nonscientist can understand. The assignment is structured so that the result will be, essentially, science journalism. In addition to the challenges of just explaining the basic subject, the writer needs to introduce us to one or more experts on that subject, then quote them directly, introducing all sources without the usual citations and references page (which are rare in real-world journal-

ism). It is all quite difficult. Scott Reichelt, our winner, writes about some of the comical absurdities of the digital password—the ideal topic, really, because while it has an extraordinarily technical side, the essay also concerns itself with a phenomenon that we all struggle with daily, and that is in fact a spectacular annoyance. After I read the essay, many months ago—and ever since—I have found myself second guessing all the absurd passwords I am required to construct and remember. See if the same thing does not happen to you. I suspect it will.

—Scott Herring, University Writing Program

Like so many other tales of intrigue, the question of whether your password is truly clever begins deep within the labyrinthian bowels of a government bureaucracy. Here, we have an unassuming middle-management engineer—Bill Burr. In 2003 the National Institute of Standards and Technology (NIST) produced a document authored by Burr—"NIST Special Publication 800-63"—that has shaped passwords worldwide to this day. This document is the origin of ubiquitous password rules, now deeply entrenched in our collective consciousness, like "a minimum of eight characters," or "must include at least one capital letter," and the uninformative "cannot be in a dictionary." Even seemingly reasonable policies like "do not use the same password on multiple accounts" are laughably impractical in reality. These guidelines are so out of step with actual human behavior that they have, in practice, offered little to no increased security.

Bill Burr—since retired—admirably admitted to the world that his 2003 recommendation was incredibly misguided. "Much of what I did I now regret," Burr told *The Wall Street Journal*, in light of the countless millions he personally subjected to "guidelines" that are seemingly ripped straight from some deranged Orwellian oppression playbook. Burr concedes that the level of frustration stemming from the NIST guidelines he authored is not "commensurate with the overall value" they provide. Hundreds of man hours researching passwords simply concludes what most non-NIST bureaucrats know implicitly: creating a password in the modern era is a harrowing experience, which either results in some derivative of `mydogsname1!` or that familiar process of recovering your account from a password as strong as it is impossible to

remember—a process which of course cycles us back to making a brand-new password.

As tempting as it is to pin entirely on Bill Burr fifteen years of typing—and retyping *slowly*—these awkward strings, mathematicians and probability theory are squarely at the root of this problem. Key components for thinking about password strength in the language of probability are the length of the password, and how many characters there are to choose from. Users typically have a pool of 94 characters (52 letters, 32 symbols, and 10 numbers) available when constructing a password, and, fortunately, NIST has socially engineered people to interpret their recommended minimum of eight characters as "my password should be exactly eight characters." A particularly useful probability formula—raise the number of options to the power of the length—can quickly marshal these two components to give an idea of how many such passwords can be made, and consequently how hard it might be to randomly guess one. With 94 characters to choose from and our NIST-sanctioned 8-character length, we get 94^8 possible passwords—that's 6,095,689,385,410,816, or roughly six quadrillion possibilities. NIST and the mathematicians would now like to think of password security as winning the lottery, with users having some random string of characters and nefarious adversarial hackers blindly guessing in hopes of winning the lottery that is your Instagram account. When the NIST guidelines were written, this amount of complexity along with such a simplified view of hackers was thought to afford users centuries of protection from an attempt to crack their password. But Bill Burr miscalculated both the human capacity for subverting rules and the ability of hackers to exploit their understanding of human behavior.

These guidelines were meant to help you outfox an adversarial hacker; however, human beings, in practice, seem to rate actually *remembering* their passwords higher than achieving the cryptographically secure string of characters NIST intended. People unapologetically repudiate the spirit of these guidelines in order to achieve something usable; we trade in the potential for a true one-in-six quadrillion random snowflake of a password for a banality that can be recalled. A 2011 study performed by students of Carnegie Mellon University along with professor Lorrie Faith Cranor subjected 5,000 brave souls to the grueling process of both creating and remembering a password—for science. One participant eloquently captures our collective struggle, highlighting the predictable

methods we all employ when faced with making a password under the watchful eye of our NIST overseers.

Our humble participant's first attempt at a password comes through as `cheese`. Something so viscerally memorable as creamy smoked Gouda meets exactly zero of the oppressive password requirements. The second attempt `1cheese1`, gets closer, but lacks—the height of password pedanticism—a symbol. A dramatic change in direction, `12#$asdf`, makes attempt number three seem promising; but, alas, no capital letter. The devolution nearly complete, attempt number four—`12#$qwER`—is a cry for help, and, moreover, it fails the dictionary lookup. A tragic misnomer covertly referencing some esoteric computer science concept while masquerading as something so commonplace as Merriam-Webster—this is not that kind of dictionary. A better name might be a password blacklist, a repository of veritable clichés so woefully unclever they are no longer allowed to be passwords. Although `12#$qwER` may look like gibberish, `qwer` are just the first four letters of the top row of your keyboard, and `1234` is not magically subtle if you hold shift on `34` to achieve `#$`. The final attempt, with the old Missy Elliot "flip it and reverse it," gives us a winner with `43@!reWQ`. Pseudo-account now created with pseudo-random password. I imagine our beloved participant—in a misguided effort to recall this monstrosity later on—scrawls it on a post-it before defiantly affixing it at the bottom left corner of their monitor; a position of honor next to their Bank of America login information.

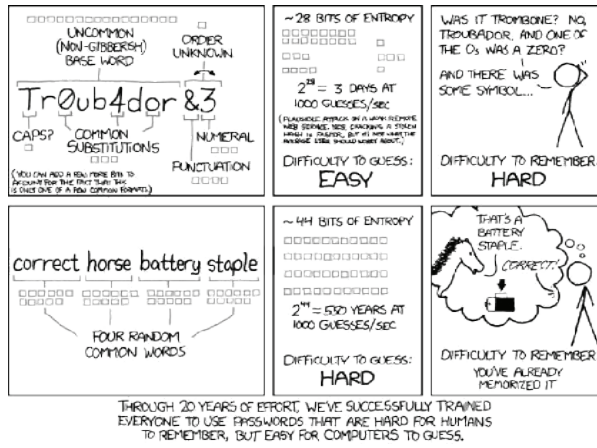

Figure 1. Password entropy comic courtesy of xkcd.com.

The physicists, in their never-ending pursuit to outperform mathematicians, possess a much better tool than probability for modeling password strength: entropy. Entropy is all about chaos—the measure of random disorder in a system. People subjected to NIST's guidelines find themselves with shockingly little password entropy. Cartoonist Randall Munroe's 2011 comic (see Figure 1) provides an apt assessment of this reality: "Through 20 years of effort, we have correctly trained everyone to use passwords that are hard for humans to remember, but easy for computers to guess." In her March 2013 TED talk, Carnegie Mellon professor Lorrie Cranor points out that "there is not actually a standard measure of [password] entropy," yet brutally effective modern password cracking algorithms abuse those formulaic—low entropy—password creation techniques we are all guilty of using. Cranor created a quilt she calls "security blanket" (see Figure 2), which depicts the 1,000 most common passwords as a word cloud. This not-so-comforting blanket demonstrates how common these practices are.

Figure 2. Security Blanket courtesy of Lorrie Cranor.

When large services have a security breach—like Yahoo's 2013 debacle resulting in three billion compromised accounts—your actual password should not be immediately known to hackers due to what is known as "hashing," or a hash function. Hash functions allow a service

to store gibberish rather than your actual password on their servers to authenticate a login attempt. Sending your password through a series of hash functions transforms your `p@$$w0rd` into actual random nonsense, but the magic here is that every time you enter your `p@$$w0rd`, it hashes to the same exact nonsense. Hash functions can also not be reversed, thus when a service—say Yahoo—has a security breach, only gibberish is recovered initially, and hackers theoretically have to spend more than enough time hashing random strings of characters—playing the password lottery—for you to change your `p@$$w0rd` to something more secure (e.g., `p@$$w0rd1`). Thus, Yahoo has no idea what your actual password is, but authenticates your login using the knowledge that whatever you type into the password field on a login page exits their hash function as the precise gibberish they have stored on the server.

Dr. Mike Pound, a computer science research professor at the University of Nottingham, has a series of videos on the cat and mouse game of passwords. His July 2016 video, "Password Cracking Pound," demonstrates the terrifying efficiency with which a specialized computer and modern algorithms can crack passwords. Using only a naive brute force attack, the specialized computer achieved an amazing 40 billion hashes-per-second; at this rate Pound can guess any targeted 8-character password in 42 hours or less by simply attempting every possible password until a match is found. Pound then demonstrates a tool (HashCat) performing a modern dictionary—that esoteric kind of dictionary—attack. Pound explains that a dictionary attack hashes a "list of commonly used passwords" first, and then "we manipulate them slightly, with rules, and we try them again." This technique models the iterative formulaic process people actually use when constructing passwords, allowing Pound to go from one password cracked every 42 hours to thousands of passwords in a few minutes. The dictionary Pound uses is made up of actual users' passwords, accumulated through a long history of security breaches like what happened to Yahoo in 2013.

The art of designing cryptographically secure passwords may seem like a zero-sum game, but there are modern approaches to regain your edge over hackers. The model of using four random words (see Figure 1) is a simple but effective direction to take. If we assume a hacker knows you are using a password like this, we can apply our probability formula—raise the number of options to the power of the length—to see how secure this might be. Using a pool of just 10,000 words and

choosing four at random, we get a massive $10{,}000^4$ (ten quadrillion) possible passwords. Dr. Pound recommends (if you're still worried) to have one word be "a bit weird," or even "just made up," and then "add a random symbol in the middle of one of those words." The likelihood that any known password-cracking approach will combine four random words in the correct order with the correct random symbol in the right position is as close to zero as we can reasonably hope. Now just make sure not to use the same password on multiple accounts, and you will have achieved NIST-approved nirvana. Or: just use a password manager and simplify your life.