Security Now! #925 - 05-30-23

Brave's Brilliant Off the Record Request

This week on Security Now!

This week, before we address what I think is a brilliant new idea from the Brave Browser's Privacy Team, we're going to see why people are suggesting that the initials HP stands for "Huge Pile"?, What was Google thinking when they created the .ZIP TLD that no one was asking for? How has the Python Foundation responded to attacks and subpoenas? Do we believe a VPN service when it promises that no logs are saved anywhere? Will Twitter be leaving the EU? Does Bitwarden now support Passkeys? Who just got fined 1.2 billion euros? — and why so little? What feature did WhatsApp just add, and what's the story about Google's new bug bounty for their Android apps? Then, after answering those questions and a brief bit of good news about SpinRite, we're going to look at Brave's Brilliant "Off the record" request concept and new feature.

The "Old School" way to add remote control to a light switch:

Security News

HP = "Huge Pile"

I'd never heard it suggested that HP stood for "Huge Pile" until I went to catch up on the current state of those HP OfficeJet Pro 9020e series printers, all of which were effectively destroyed at the firmware level more than three weeks ago, the morning of Monday May 8th, by an aberrant autonomous firmware download and install. And to add insult to injury, it may be that the entire purpose for the badly failed update was simply to tighten HP's grip over the use of non-HP printer ink. But whatever the motivation, the entire world, which has now been waiting for 22 days, is now filled with HP branded dead printers which no longer print – using anyone's ink.

And what's becoming more and more glaring with each passing day, is that, incredibly, there has been very unsatisfying communication from HP about this issue. Any call to HP service replies that HP is aware of the issue and is working diligently on a solution. No matter what solution is eventually forthcoming, this has not been very impressive responsibility-taking on HP's part. Twenty two days and counting, without any official statement to HP's users, has stepped over the line of corporate responsibility. It's one thing to control the message. But it's another thing to offer no message at all.

The ".ZIP" TLD — What could possibly go wrong?

Not quite four weeks ago, back on May 3rd, Google, which is now a fully fledged domain name registrar, announced their eight new top level domains with a posting headline: *"8 new top-level domains for dads, grads and techies".*

The TL;DR was: "Google Registry launches 8 new top-level domains: .dad, .phd, .prof, .esq, .foo, .zip, .mov and .nexus." For "Dads" they said:

> Knock, knock. Who's there? With Father's Day right around the corner, .dad is here for the jokes, the games and the advice. Whether you're a fit.dad, a gay.dad or a dude.dad, .dad is the place to celebrate fatherhood. Check out these interesting .dad websites:
>
> - **Classic.dad**: Play this fun 8-bit game where your goal is to successfully mow the lawn while dodging pets and obstacles, and preventing weeds from spreading.
> - **Dear.dad**: Check out this media platform dedicated to telling stories of Black fathers.
> - **Daily.dad**: Daily.dad is a new book from Ryan Holiday that provides 366 accessible meditations on parenthood, a manageable slice for each day.

Okay. For Grads they write:

> *May means graduation season for many in higher education. We're celebrating graduates and the professors who taught them well by launching .prof, .phd, and .esq. Whether sharing legal advice for everyday life or teaching courses on behavioral science, these new domains are perfect for showing off your credentials. Hats off to these early adopters:*

- ***Erika.esq**: Erika Kullberg is an attorney and money expert who is passionate about better positioning people for success.*
- ***Casey.prof**: Professor Casey Fiesler is a technology ethics educator and science communicator.*
- ***Rafael.phd**: Rafael Misoczki is an expert in post-quantum cryptography, fully homomorphic encryption, privacy enhancing technologies, and the application of these constructions.*

I suppose this makes possible an entirely new set of vanity domains. But the real worry, and the reason for my bringing this up today, is the new TLD's Google chose to create for Techies. Under the heading of "Techies" they write:

*May is also the month of Google I/O, our annual developer conference. Whether you're learning to code, deploying a helpful tool, building your portfolio, or starting a new community, **.foo, .zip, .mov and .nexus** have you covered. Here are some examples from our developer community:*

- ***gamers.nexus**: Use gamers.nexus to review computer hardware and plan your next gaming PC.*
- ***helloworld.foo**: Learn how to code "hello world" in each programming language.*
- ***url.zip**: Create short, powerful and trackable links with url.zip*
- ***david.mov**: Watch videos by David Imel in this liminal space.*

And just to finish off their posting, before we look at not only what **could** possibly go wrong, but what **did** go immediately and catastrophically wrong, Google finished with:

Starting today, you can register all of these new extensions as part of our Early Access Program for an additional one-time fee. This fee decreases according to a daily schedule through the end of May 10. On May 10 at 16:00 UTC, all of these domains will be publicly available at a base annual price through your registrar of choice. To make it super easy for anyone to get their website live, we've worked with Google Sites to launch new templates for graduates, professors and parents.

To learn more about pricing and our participating partners, visit registry.google.

If those **grads** are looking for jobs in Internet security, protecting the assets belonging to those **Dads**, then Google's difficult-to-understand decision to offer domains under the **.ZIP** top level domain may create the full employment guarantee those new grads have been looking for.

The malicious exploitation of these .ZIP TLD's took exactly no time since their abuse was so obvious to everyone except for, apparently, Google. The new phishing technique is already going by the name "file archiver in the browser." It can be leveraged to "emulate" file archiver software in a web browser when a victim visits a .ZIP domain.

"mr.d0x", a security researcher we haven't quoted in some time, recently said: *"With this phishing attack, you simulate file archiver software (e.g., WinRAR) in the browser and use a .zip domain to make it appear more legitimate."*

In other words, threat actors are creating realistic-looking phishing landing pages using modern web tools HTML, CSS and JavaScript which mimics legitimate file archive software, hosted on a .zip domain to elevate social engineering campaigns. In a potential attack scenario, a miscreant could redirect users to a credential harvesting page when a file which is apparently "contained" within the fake ZIP archive is clicked. mr.d0x noted: *"Another interesting use case is listing a non-executable file and when the user clicks to initiate a download, it downloads an executable file. Let's say you have an "invoice.pdf" file. When a user clicks on this file, it will initiate the download of a .exe or any other file."*

Additionally, the search bar in the Windows File Explorer can emerge as a sneaky conduit where searching for a non-existent .ZIP file opens it directly in the web browser should the file name correspond to a legitimate .zip domain. Mr.d0x said: "This is perfect for this scenario since the user would be expecting to see a ZIP file. Once the user performs this, it will auto-launch the .zip domain which has the file archive template, and able to appear legitimate."

The problem, of course, is that the new .ZIP and .MOV TLD's are also both legitimate file extension names. This invites confusion when unsuspecting users mistakenly visit a malicious website when they believe that they've opening a file. They could then be misled into downloading malware.

Trend Micro agrees that this is not a good idea, writing: "ZIP files are often used as part of the initial stage of an attack chain, typically being downloaded after a user accesses a malicious URL or opens an email attachment. Beyond ZIP archives being used as a payload, it's also likely that malicious actors will use ZIP-related URLs for downloading malware. The crux of the concern is that with the introduction of TLD's that are identical to well known file extensions, bad guys are going to cook up clever new ways to take advantage of this resulting confusion and ambiguity. If nothing else, it clearly equips bad actors with another vector for phishing.

I got a kick out of MalwareBytes Labs, which titled their recent discussion of this ill-advised move by Google: *"ZIP domains, a bad idea nobody asked for."* They wrote:

> *If you heard a strange and unfamiliar creaking noise on May 3, it may have been the simultaneous rolling of a million eyeballs. The synchronized ocular rotation was the less than warm welcome that parts of the IT and security industries—this author included—gave to Google's decision to put .zip domains on sale.*

MalwareBytes Labs' author had some additional useful things to add which I want to share:

> *Domain names and filenames are not the same thing, not even close, but both of them play an important role in modern cyberattacks, and correctly identifying them has formed part of lots of basic security advice for a long, long time.*

The TLD is supposed to act as a sort of indicator for the type of site you're visiting. Dot com was supposed to indicate that a site was commercial, and dot org was originally meant for non-profit organizations. Despite the fact that both dot com and dot org have been around since 1985, it's my experience that most people are oblivious to this idea. Against that indifference, it seems laughable that dot zip will ever come to indicate that a site is "zippy" or fast, as Google intends. Quoting Google: "When you're offering services where speed is of the essence, a ***.ZIP*** *URL lets your audience know that you're fast, efficient, and ready to move."*

Meanwhile, plenty of users already have a clear idea that .zip means something completely different. Since the very beginning, files on Windows computers have used an icon, and a filename ending in a dot followed by three letters to indicate what kind of file you're dealing with. If the three letters after the dot spell z-i-p, then that indicates an archive full of compressed—"zipped up"—files. The icon even includes a picture of a zipper on it (because reinforcement is good, and confusion is bad.)

As it happens, cybercriminals love .zip files and the last couple of years has seen an explosion in their use as malicious email attachments. Typically, the zip file is first in a sequence of files known as an "attack chain". In a short chain, the zip file might simply contain something bad. In a longer chain it might contain something that links to something bad, or contain something that contains something that links to something bad, or contain something that links to something that contains something that links to something bad. You get the idea.

The key to it all is misdirection. The attack chain is there to confuse and mislead users and security software.

Criminals use other forms of misdirection in file extensions too. An old favorite is giving malicious files two file extensions, like evil.zip.exe. The first one, .zip in this case, is there to fool you. The second is the real one: A dangerous executable type, .exe in this example. Given a choice of two, users have to decide which one to believe. Most aren't even faced with that choice though because Windows helps with the subterfuge by hiding the second file extension, the one you really should be paying attention to, by default. So you just see the ".ZIP".

Domain names get the same treatment. Criminals make extensive use of open redirects for example—web pages that will redirect you anywhere you want to go—to make it look as if their malicious URLs are actually links to Google, Twitter or other respectable sites. Less sophisticated criminals just throw words like "paypal", or anything else you might recognise, into the link and hope you'll notice that bit and ignore the rest.

Against that backdrop, Google inexplicably decided to introduce something that will generate no useful revenue but will give cybercrooks an entirely new form of file and domain name misdirection, to add to all the others we're still wrestling with.

So, what could criminals do with this new toy? There is no better example than that provided by security researcher Bobby Rauch, in his excellent article ***"The Dangers of Google's .zip TLD"****.*
In it, Rauch challenges readers to identify which of the following two URLs "is a malicious phish

that drops evil.exe?"

https://github.com/kubernetes/kubernetes/archive/refs/tags/v1.27.1.zip
https://github.com/kubernetes/kubernetes/archive/refs/tags/@v1.27.1.zip

The answer is that the evil link is the second one. The top one opens a zip file called v1.27.1.zip from the github.com domain. The second would go to the domain v1.27.1.zip, which in this hypothetical example triggers the download of whatever that site wants to provide.

In other words (and this is me speaking), the Internet just became a whole lot more dangerous, even for those of us who take some pride in understanding all this stuff, in knowing how to examine URL's, and in knowing our way around.

I've placed a link to Bobby Rauch's full write-up which explains, in detail, how simple it is to craft .ZIP URLs that look convincingly like URL's for well-known sites that are offering .ZIP archives for download... whereas they are actually taking the user to a .ZIP domain where an internal redirect would result in the user's browser silently downloading something entirely different from an entirely untrusted domain:

https://medium.com/@bobbyrsec/the-dangers-of-googles-zip-tld-5e1e675e59a5

I hope that everyone will think about this and take it seriously, since the danger is very real. Bobby's article shows how UNICODE characters which look like forward slashes are not treated as separators by Chromium browsers, which enables this to appear all the more convincing.

Our MalwareBytes labs guy continues...

If you figured it out, well done, but remember you knew that one of them was bad. Would you have spotted it if you hadn't been forewarned? And if you didn't spot it, don't feel bad, that's the whole point. It's hard to read URLs even if you know you're looking for something out of place. Of course, the invention of dot zip domains didn't suddenly make URLs hard to read, they already were, but that's no excuse.

Google does an awful lot of really good stuff for computer security, for which it deserves enormous credit, and this is an uncharacteristic misstep. The search giant was under absolutely no pressure to create a dot zip TLD and it hardly seems destined to become a major income stream.

Dot zip domains are not yet a serious problem. At the time of writing, a little fewer than 4,000 have been registered, some of which were almost certainly bought by security researchers wanting to demonstrate what a bad idea they are, or to deprive criminals of some of the more dangerous names.

Criminals may yet decide they don't need the built-in confusion of the dot zip domain (or at least, not today). They already have a whole bag of tricks that work very well and if a new one doesn't make their life easier or richer, they won't use it.

It is also possible that dot zip will simply die on the vine if enough companies choose to block it. Last week, Citizen Lab's John Scott-Railton urged his nearly 200,000 Twitter followers to simply "block it all", saying "The chance that new .zip and .mov domains mostly get used for malware attacks is 100%."

It's for you and your organization to decide if you should block it, but I will point out that if you are going to, the best time to do it is now: Almost nobody is currently using it, and nobody is going to use it in future if it's routinely blocked.

It is true that the entire security community is reacting with a huge amount of WTF and disbelief that Google actually did this.

Robert Lemos writing for DarkReading titled his coverage: *"Google's .zip, .mov Domains Give Social Engineers a Shiny New Tool"* Wired wrote: *"The Real Risks in Google's New .Zip and .Mov Domains"* TechTarget: *"The potential danger of the new Google .zip top-level domain"* and Y Combinator: *"The Dangers of Google's .zip TLD"*

So, we are all now forewarned. Be on the lookout for .ZIP domains masquerading as .ZIP files.

PyPI gets more serious about security AND privacy

Last week I noted that the Python Software Foundation had been forced to take the unprecedented step of temporarily shutting down all new account and Python package creation due to an overwhelming automated attack on the world's leading Python software registry. In response to the attack, plans are now underway to require two-factor authentication (2FA) for PyPI accounts, and even more interestingly, to reduce the instances where the PyPI portal needs to store a user's IP address.

In a major but welcome and probably inevitable change of policy, by the end of the year, all accounts that maintain a Python library on the PyPI portal must choose a 2FA method or have their access to some PyPI features limited. The Foundation says the move comes to improve the supply chain security of the Python ecosystem. The requirement for 2FA is intended to help block account takeovers and also to raise the bar for automated attacks like the ones PyPI has been subjected to over the past few months.

2FA has been supported and available, but its use has been optional. That's what's changing. Last year, the Foundation offered free security keys to the maintainers of its Top 1% packages in a move to show PyPI devs that 2FA would not be the serious disruption that some believed it would be.

But it's the second piece of news that I thought was most interesting. As I was reading the reports that the Foundation was making an effort to purge IP addresses from its server logs,

replacing them with hashes of the IP addresses, or perhaps geographic data provided by Feedly, their CDN provider, I didn't understand what that had to do with anything. The Foundation said that the only systems that will see plaintext versions of a user's IP address will be their rate-limiting systems, and the Foundation says it's working on replacing those as well.

Providing some additional detail, the Foundation explained that they would be salting their hashes because the IPv4 address space, being relatively small at only 32-bits, would allow for the creation of hash-reversing lookup tables. As we know, the normal practice with salted hashes is to pick a unique per-hash salt at random and store it alongside the hashed value. This would allow an IP comparison by using the same salt against a candidate IP to see whether their hashes matched. But in this case, that normal approach doesn't provide the level of protection the Foundation requires due to the fact that the 32-bit IPv4 search space is so small. Instead, the Foundation said that they're going to use a **single secret salt** which will not be stored in the database, and that they would be taking steps to protect against the leakage of this information.

So, that was all well and good, but it still left unanswered the question of **why** the Foundation was bothering to hash their database's IP addresses? What was the point of that? The answer was somewhat chilling... and it’s the same reason ExpressVPN, a sponsor of this network, goes to great lengths to protect their users’ IP addresses. Get this:

The news of this major engineering effort came two days after the Foundation revealed that it had been subpoenaed by the US Department of Justice to reveal information on five PyPI accounts. The requested information included the users' IP addresses, but also the IP addresses of all users who had downloaded packages from those five users. The Foundation wants to eliminate their own ability to respond to such subpoenas in the future, thus protecting the IP addresses and thus to some degree the identities of those who are uploading and downloading Python libraries.

The problem here is essentially identical to the issue with encryption. While no one wants to provide care and comfort to criminals, there is also a reasonable presumption of privacy and unfortunately we keep seeing instances of government overreach... such as the other story we covered last week where the US FBI abused the powers of FISA data collection to improperly search the personal communications of Americans more than 300,000 times in a little more than one year, according to the US Senate Intel Committee. If the government wants the trust of its citizens, the government needs to first be worthy of that trust. These revelations don’t engender much trust. And we now have very strong privacy enforcing technologies.

Other package registries have also announced plans to require their users to enable 2FA, and while The Python Foundation’s move to purge IP addresses from logs and databases is a novel solution, it may develop into more standard practice in the future.

“No logs saved anywhere” ???

An object lesson story appeared in the news last week. The so-called “Super VPN” service by the publisher “SuperSoftTech” whose Android App has been downloaded more than 100 million times, was exposing more than 360 million records due to its Chinese developer leaving the app’s database open and exposed to the internet.

And that database contained all of the information the VPN provider explicitly promised would never be exposed, including information on all of its paid customers, including details such as emails, the customers' real IP addresses, geolocation information, and the VPN servers they connected to.

The database was secured after it was discovered and reported to the developer by security researcher Jeremiah Fowler. And it's worth noting that Jeremiah's findings contradict the app's Gopogle Play Store listing at the time (it's since been removed), where the service claimed "no logs saved anywhere."

Our obvious takeaway lesson here is that it's very easy to make the claim to not be retaining any logs. No one using a VPN wants their usage to be logged. There's zero benefit to the user from that. But it's obvious that the anti-logging claim needs to be true for it to have any value whatsoever. So it matters who's making the claim.

Twitter in the EU?

Twitter may be facing stiffer headwinds with the EU's 27-nation bloc after Twitter, last Friday, chose to drop out of a voluntary European Union agreement to combat online disinformation. The Associated Press reported that European Commissioner Thierry Breton tweeted (right) that Twitter had pulled out of the EU's disinformation "code of practice" that all other major social media platforms including Google, TikTok, Microsoft, Facebook and Instagram, among others, have all pledged to support. He added that Twitter's "obligation" remained, referring to the EU's tough new digital rules taking effect in August. Breton said: *"You can run but you can't hide."* Tough talk.

There were early signs that Twitter wasn't prepared or planning to live up to its commitments. The European Commission blasted Twitter earlier this year for failing to provide a full first report under the code, saying it provided little specific information and no targeted data. Breton said that under the new digital rules that incorporate the code of practice, fighting disinformation will become a *"legal obligation."* Okay. But Elon is notorious for ignoring any legal obligations that he doesn't need to heed. Breton concluded by adding, somewhat ominously, *"Our teams will be ready for enforcement."* It will be interesting to see what, if anything, that amounts to. Perhaps some monetary fines that Elon will simply ignore? I doubt that the EU has the huevos to block all Twitter access outright. So Elon is probably successfully calling their bluff.

Bitwarden's support for Passkeys

There was a flurry of confusion in the tech press last week over Bitwarden's announcement of their increased support for Passkeys. It's not what we've been hoping for. At least not yet. Bitwarden is saying that end users will be getting Passkey support for their Bitwarden clients sometime later this summer.

The news was that their passwordless.dev site -- the group they acquired not too long ago -- is offering APIs and code libraries to enable enterprises to add back-end Passkey WebAuthn support to their authentication flows. And there was also news that Bitwarden client users would

be able to use Passkeys to authenticate to the Bitwarden client.

So, a bit more patience will be required. And remember that even once we have it, it's still only supported by a few sites. But that will change over the next several decades.

A €1.2 billion fine will grab your attention:
As I was scanning recent news for interesting updates and discussion, the phrase €1.2 billion Euro GDPR fine was difficult to ignore. It appears that Ireland's Data Protection Commission wanted to get everyone's attention. So they levied a €1.2 billion Euro fine against Facebook's parent, Meta, for their failure to comply with the EU's GDPR laws. The Irish officials claim that Meta illegally transferred the personal information of EU users to the US without their approval. What's apparently meant by this is the issue of where Facebook data resides, as in, which data center. So the IDPC — the Ireland Data Protection Commission — has ordered the company to cease any data transfers and delete existing user data within six months. And in case anyone was wondering, yes, €1.2 billion is, indeed, the largest fine so far imposed under the EU's GDPR. To my mind, this represents a failure of imagination on the part of those plucky Irishmen. If they wanted to really get everyone's attention over a fine that was never going to be paid anyway, they should have gone directly to €1.2 gazillion Euros (with a 'G'). Why settle for a mere billion when you might as well ask for a gazillion?

Meta, for their part, said: “Let us think about it... Okay, no.” To exactly no one's surprise, Meta said that they would be appealing this nonsense. In somewhat more detail, Meta explained that they were being unfairly singled out for attack because thousands of businesses and organizations rely on the ability to transfer data between the EU and the US in order to operate and provide everyday services. They explained that the real issue was not about one misbehaving company's GDPR-violating privacy practices, but rather that there are fundamental conflicts of law between the rules in the US and the EU, and that it's those laws that need to be brought into alignment. And, notably, respective policymakers are expected to be resolving all of this in the summer. So, yes, Meta formally stated that they would be appealing the ruling, including the unjustified and unnecessary fine, and seek a stay of the orders through the courts. And meanwhile, Facebook continues unimpeded throughout Europe.

I've hugely abbreviated and summarized the situation, since I think this is all going to dissolve pretty quickly once the US and the EU figure out how to deal with this; and Meta is certainly correct in noting that business could not be conducted with Europe — to Europe's significant benefit — if personal information could not readily flow across the EU's borders. So, whatever grandstanding Ireland is doing through the GDPR won't amount to more than wasting everyone's time.

Meta posted an interesting and fact-filled response, which I've linked to in the show notes for anyone who might be interested in additional, and somewhat more serious, details. I obviously don’t think that it warrants much more time or attention.

https://about.fb.com/news/2023/05/our-response-to-the-decision-on-facebooks-eu-us-data-transfers/

Editing WhatsApp messages
Briefly... while we're on the subject of Meta and Facebook, I wanted to note that WhatsApp messaging announced Monday before last that they would be adding a feature that iOS users received with the update to the "16" versions of iOS and iPad OS... which is the ability to edit, for up to 15 minutes, the content of any recently sent message. In the cast of iOS, Apple allows up to five edits of a message during that 15-minute window,which I suppose is useful if you're really having a difficult time getting it right.

I was glad to be reminded of this new feature of iOS 16 since I have a buddy who frequently follow-up his messages by re-reading them and then sending separate typo correction notes. So the next time this happens I plan to let him know that he's now able to edit the original message in vivo rather than in vitro.

A new Google Bug Bounty
And, finally, Google has announced a new bug bounty program aimed at their own franchises' Android apps. They've named it "Mobile VRP" for "Mobile Vulnerability Reward Program." Describing the program they wrote:

> *Google's Mobile Vulnerability Rewards Program (Mobile VRP) focuses on first-party Android applications developed or maintained by Google. The Mobile VRP recognizes the contributions and hard work of researchers who help Google improve the security posture of our first-party Android applications. The goal of the program is to mitigate vulnerabilities in first-party Android applications, and thus keep users and their data safe. Only apps published by the developers in the list below, or apps in the Tier 1 list (see Application tiers) are in scope for the Mobile VRP: • Google LLC • Developed with Google • Research at Google • Red Hot Labs • Google Samples • Fitbit LLC • Nest Labs Inc. • Waymo LLC • Waze*

There are three general classes of vulnerabilities that qualify under their rewards program: The first is **Arbitrary code execution (ACE)** when they explain: Vulnerabilities of this type allow an attacker to execute arbitrary code in the context of the vulnerable application. In order to qualify, the ACE should allow an attacker to run native code of their choosing on a user's device without the user's knowledge or permission, in the same process as the affected app (there is no requirement that the OS sandbox needs to be bypassed). They provide three examples:

- An attacker gaining full control of the application, meaning code can be downloaded from the network and executed.
- Overwriting a .so file with a malicious .so file that is executed by the victim app.
- Executing Java code in order to call "exec" and thus run arbitrary native code.

They add that merely tricking a user into installing an app and executing code within that app itself does not qualify.

The second of three qualifying classes is **Theft of sensitive data** which includes vulnerabilities that lead to unauthorized access to sensitive data from an app on an Android device, where the scope of qualifying data is:

- Data that enables unauthorized access to a user's account (e.g. login credentials, or authentication tokens that are able to perform sensitive state-changing actions that result in non-trivial damage to the victim).
- Sensitive user-generated data: contact list information, photos (unless made public by default), content of a user's messages (email, instant messages, text messages), call/SMS logs, web history (being able to profile or enumerate a specific user based on their web history), or browser bookmarks.
- Information that is linked or linkable to an individual, such as medical, educational, financial, or payment data, and employment information.

Location information alone does not qualify (unless combined with the ability to uniquely identify an individual) and access to non-sensitive internal files of another app also does not qualify. But examles of vulnerabilities that impact sensitive data include, but are not limited to:

- Insecurely stored data files containing sensitive data that are accessible to other apps
- Sensitive data sent over insecure network connections that can be intercepted
- Insecurely designed app internals like content providers or activities that can be manipulated to expose sensitive data

And, finally, under **Additional vulnerability types in scope** Google says that other vulnerabilities which are not strictly within either of the two explicit scopes will be taken into consideration if they are shown to have a security impact. In other words, we're willing to be surprised. This would typically be security weaknesses that need to be used in conjunction with other vulnerabilities to create an exploit chain. Examples of vulnerabilities which in themselves might not result in direct **arbitrary code execution** or **theft of sensitive data**, but could still qualify for a reward are:

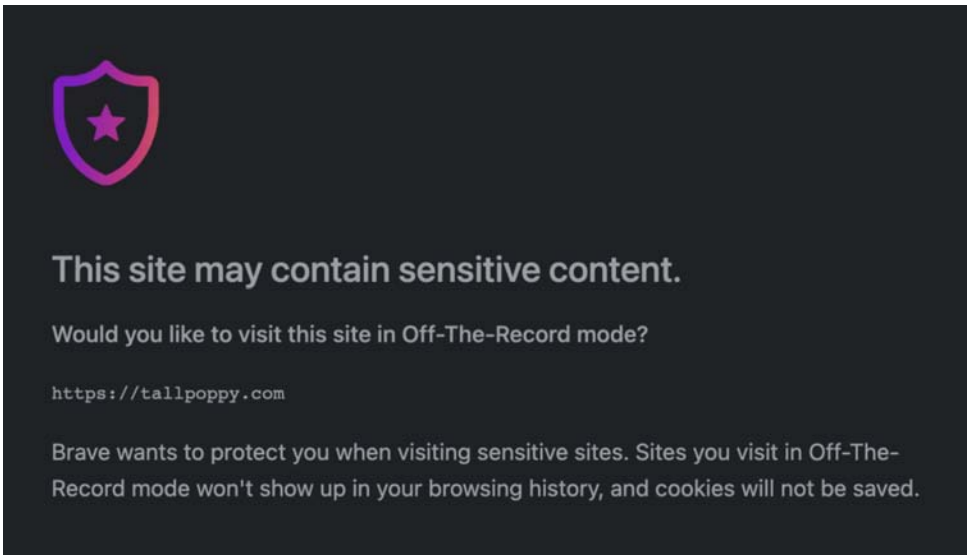
- Path traversal / zip path traversal vulnerabilities leading to arbitrary file write
- Intent redirections leading to launching non-exported application components
- Vulnerabilities caused by unsafe usage of pending intents
- Orphaned permissions

Qualifying rewards range as high as $30,000 for a zero-click arbitrary code execution. We've witnessed the emergence and clear success of the concept of bug bounties where good guys are paid for finding and responsibly disclosing previously unknown and important bugs.

SpinRite

I finished the full rewrite of SpinRite's mass storage data operations back-end, and I finished testing it all by the end of last weekend. So, Sunday evening I released Alpha-28 to the gang over in GRC's spinrite.dev newsgroup and they have begun to put it through its paces. It's too soon to appraise its performance, but I'm sure I'll have something to report next week. In my own testing, it felt significantly more solid and responsive than ever, and it incorporated many new features thanks to the *"Holy crap! **THAT** can happen?"* education I've received over the past six months since the release of SpinRite's first Alpha. So, I'm very glad that I took that month to scrap and re-engineer that part of SpinRite which is, after all, where all of the actual work happens.

Brave's Brilliant Off the Record Request



Last Wednesday, The Brave browser's privacy team announced a slick new forthcoming feature that I would **love** to see obtain some traction within the wider browser community, meaning also being adopted by Google for Chromium and Mozilla for Firefox. It's called "Request OTR" where OTR is short for "Off The Record." The shortest summary of this interesting idea is that this feature allows websites to suggest to the browser that this user's visit to the site should probably be forgotten and thus remain "off the record."

Here's how the Brave Privacy Team explained their idea:

Starting in version 1.53, Brave will begin rolling out a new feature called "Request Off the Record (OTR)." This feature aims to help people who need to hide their browsing behavior from others who have access to their computer or phone.

For example, a person who is the victim of intimate partner violence who needs to find support services without their partner knowing, or someone needing to find personal healthcare without others in their home finding out.

Request OTR** allows **websites** to optionally describe their own content as **"sensitive."
The browser can then ask if the user would like to visit the site in OTR mode, where the site is visited in a clean, temporary storage area. Sites visited in OTR mode are not saved to your browsing history, and any cookies, permissions, or other site data do not persist to disk. Meanwhile, all other normal sites visited are stored and treated normally, hiding the fact from anyone who may access the device later, that any "unusual" behavior happened.

Brave intends to work with other browser vendors to standardize OTR, so that at-risk browser users can be private and safe, across the Web, regardless of which browser they're using.

This feature has been designed with the input of, and in collaboration with, several civil society and victim advocacy groups. We agree with Mallory Knodel, the CTO at the Center for Democracy and Technology, who said:

> *Brave Browser's attention to detail with OTR Mode, where users can more easily choose which websites are recorded in their browsing history, is an important privacy innovation that can protect users in "attacker you know" situations or anyone who wants more control over what their browser remembers and what it doesn't. This feature empowers people who browse the web—all of us—and gives us more agency over content consumption.*

Some users need to hide their browsing from people who have access to their device Most often, when people talk about Web privacy they're talking about protecting personal data from other websites – (e.g. blocking Google from recording the sites visited).

However, Web users have other privacy needs too, needs that are currently poorly served by most browsers. Consider "Sarah," a hypothetical Web user who lives with "Stan," a physically abusive partner. Sarah needs to use the Web to learn about legal, medical, and other support services in her area, so she can safely exit her relationship. Stan, though, suspects Sarah may be planning to leave, and begins monitoring Sarah's phone, computer, and other devices to see if she's contacting support services.

Unfortunately, not only do browsers fail to protect users like Sarah, they actually make it easier for abusers like Stan to digitally surveil others. Browsers record a wealth of information about our browsing behavior and interests, both explicitly (e.g. browsing history, DOM storage, and cookies) and implicitly (e.g. cache state, saved credentials, URL autocomplete). Worse still, the tools browsers do include to protect people like Sarah are incomplete and / or difficult to use correctly.

Browsers currently provide some tools to help users hide their activity on sensitive sites. However, these tools are insufficient to protect people whose safety depends on hiding visits to specific sites from people who have access to their device. Existing tools either hide too much (thus inviting suspicion from abusers), too little (thus allowing abusers to recover browsing history), or are otherwise difficult to use successfully.

Private (also known as Incognito) windows allow users to browse the Web without their browsing activity being permanently recorded. Unfortunately, private windows do a poor job protecting users from on-device surveillance. It's easy to forget to open a private window before visiting a site, especially under stress, thus causing the site visit to be permanently recorded. And it's equally as easy to forget to close the private window, and thus continue browsing in the private window beyond just the target sensitive site. This can reveal to the abuser that private browsing modes have been used, which on its own may elicit suspicion or put the victim at further risk.

Similarly, some browsers include advanced browser controls that can be used to delete browser storage for specific sites. This approach has the drawback of needing to be performed after the site was visited, instead of protecting the user during the visit, which may put the

user at risk if the browser needs to be closed quickly. Additionally, these controls are often difficult to find, and more difficult still to use correctly for non-technical users. And finally, these browser controls typically only allow the user to delete stored values for the site (e.g. cookies or permissions), but do not allow the user to delete other traces of the site (e.g. browsing history or caches).

Finally, some sensitive sites include quick-exit buttons in the site themselves, which allow a visitor to quickly navigate away from the site in a way that may be semi-difficult for an abuser to detect. While useful, this approach is also incomplete. Quick-exit buttons cannot delete many types of site data (e.g. permissions or caches), and are constrained in their ability to modify browsing history. Further, they depend completely on the correct implementation by the site; the browser is unable to protect the user.

In contrast, Brave's "Request OTR" approach provides a comprehensive way for sensitive sites to request to be omitted from a user's browsing history and local storage.

Any site can request to go OTR, *and the user is prompted to determine whether they would like to do so. If so, the Brave browser creates a temporary storage area for the site,* ***and does not record the site visit in the user's browsing history.*** *The OTR session is tied to the site, and any other sites the user visits, even in the same tab (along with any sites visited in any other tabs) are recorded in browsing history as usual.*

Brave's implementation of Request OTR protects the user in the following ways:

- *The user is prompted and proactively asked, up front, whether they wish to have their visit forgotten after they leave.*
- *The user is protected the entire time they're visiting a sensitive site; they don't need to attempt to scrub their browsing history later.*
- *Other, non-sensitive sites are recorded as usual, which prevents the appearance of large gaps in browsing history that might look suspicious to an abuser.*
- *All target site behaviors are prevented from persisting to disk, including cookies, caches, browsing history, permissions, etc.*
- *And OTR prevents sites from abusing the feature; a site* ***cannot*** *go off-the-record unless a user explicitly gives the site permission to do so.*

Brave has developed ***Request OTR*** *specifically to help people suffering from intimate partner violence, or people otherwise needing to hide visits to sensitive sites from their browsing history. However, OTR is intentionally a general browser feature, and is intended to be usable by any site on the Web.*

There are currently two ways for a site to request to go off the record in Brave. The primary, intended way is for the site to include the header ***Request-OTR: 1*** *in the website's response to the initial navigation request to a site. If the browser receives this header, the browser will halt the navigation and ask the user if they would like to visit the site* ***off the record.***

If the user says yes, then the browser does two things:

- *It does not record the site visit in the browser history, and,*
- *It creates a temporary storage area for caches, cookies, permissions, etc.*

The browser continues using this temporary storage area for all subsequent pages visited within the same tab, within the same site. When the user closes the tab, or navigates away from the site, the temporary storage area is discarded, and browsing behaviors return to being recorded as normal.

The second way for a site to request to go off the record is to be included in Brave's preloaded list of "request off the record" partner sites. These are sites that serve victims of intimate partner violence, and have told Brave they're interested in being considered a sensitive site by default by the browser. This list is intended as a bridge measure, until all such sites have the opportunity to implement the previously mentioned header approach.

There are a few caveats:

Users should be aware that Brave's Request OTR feature cannot protect users from other software on their computer that might record information about what sites they visit. Examples of software the Brave browser cannot hide browsing history from include:

- *Browser extensions*
- *Network spying*
- *Malware or spyware installed on the device*
- *Information saved by sites before or after you visit the "off the record" (such as if you have "Google Web History" enabled on Google Search or Gmail)*
- *Operating-system level logging*
- *Crash logs*

Brave is exploring what additional protections can be provided against such threats, but users should be aware that (as with systems like private browsing mode) Brave's Request OTR mode only prevents recording of core browsing behaviors and data.

We're excited to release Request Off the Record in the upcoming version 1.53 of our desktop browser, with an Android version coming in the 1.54 release. We'll be rolling it out to users shortly, though people interested in testing the feature now can enable it by visiting ***brave://flags*** *and enabling* ***#brave-request-otr-tab****. Please note that this should only be done if you understand the risks of testing experimental browser features. We welcome all feedback on the feature.*

We're also excited about the next steps we're taking to further improve the Request OTR feature.

First, we're working with experts and researchers at George Washington University and Paderborn University to evaluate how Request-OTR is understood by users, and how we can further convey to users exactly what protections the feature does (and does not) provide. We will both share the research that results from this collaboration on this blog, and incorporate it into future versions of Request-OTR in Brave.

Second, we're interested in working with other browsers, organizations, and Web companies to potentially standardize Request-OTR, so that users of other websites and browsers can benefit from the protection. Our current implementation is the result of working with a wide range of abuse advocates, technologists, browser specialists, and NGOs, and we're eager to continue working with similar organizations to best support Web users.

To all of that foregoing, I say a huge Bravo!

In 2023, with our web browsers at their current level of maturity, it's extremely rare to encounter something so clear, clean and simple that's able to offer new and useful benefits to a user's browsing experience. This new feature is one such.

The user's experience of this is perfect. It could be baked into every browser, and most of us would never have any idea that it was present. But the moment someone who was in an environment that might make them vulnerable, visited a website that understood that by virtue of the services it offers, its visitors might need the site's help protecting themselves, the website could immediately prompt the browser to proactively ask the visitor whether they would like all use of this site to remain "off the record" and be immediately forgotten after they've left.

It's a clean and simple and wonderful addition to the traditional incognito mode of browser usage. I hope that the clarity and objective goodness of this idea captures the imaginations of those at Google and Mozilla and that it will grow into the world wide web standard that it deserves to be.

https://brave.com/privacy-updates/26-request-off-the-record/