Large-scale Analysis of DNS-based Tracking Evasion - broad data leaks included?

User tracking technologies are ubiquitous on the web. In recent times web browsers try to fight abuses. This led to an **arms race** where new tracking and anti-tracking measures are being developed. The use of one of such evasion techniques, the **CNAME cloaking** technique is recently quickly gaining popularity. Our evidence indicates that the use of the CNAME scheme **threatens web security and privacy** systematically and in general.

I pointed to some of such risks back in 2014, but I must say that the landscape changed significantly. Our current work is the **first systematic, deep security & privacy analysis of the technique**. Because the CNAME technique is today increasingly used on the web, the results are worrying. I would actually say that we're near the worst scenario, with systemic data leaks and security vulnerabilities found in the wild, making the ecosystem more fragile. In this post, I explain the implications, including regulatory-wise.

The research paper is a joint work with Yana Dimova, Gunes Acar, and Tom Van Goethem, to whom I am grateful for the excellent collaboration and work. It was an amazing collaboration that spans 2019, 2020, and 2021. Thanks for all the hard work put into this important problem! Our results are correct and complete, and the work is accepted to Privacy Enhancing Technologies Symposium, 2021 . The paper is here. Description follows.

We stress how the **CNAME technique bypasses anti-tracking measures and some of its users are targeting some specific web browsers (Apple Safari)**. We show that the use of the CNAME tracking method leads to the **proliferation of security and privacy bugs on the web**. This tracking measure potentially leads to a massive data breach, where user data is leaking or is being acquired by trackers. This happens before our eyes.

We show how:

- The use of **CNAME cloaking is introducing web security bugs** that let compromise/hack unsuspecting users. Websites could misuse the bugs to compromise the security of web users, systematically. It's a systemic issue in the case of a few trackers.
- The use of the CNAME cloaking technique leads to massive cookie leaks. **In 95% of cases of websites using this technique, we found cookies leaking to external tracker servers** in an unsanctioned manner, invisible to the user. In some cases, we confirm that the leaked cookies contain private/sensitive data. All these likely trigger the violation of data protection regimes such as the GDPR, or maybe even the CCPA.

We report that this tracking technique is prevalent on popular websites. We find it on 9.98% of the top 10,000 websites. The use of this method is rising (21% up, over the past 22 months). We detect 13 providers of such tracking “services” on 10,474 websites. This scheme leads to data leaks on 95% of the websites employing it. Such data leaks sometimes involve unambiguously private data. GDPR alert lights should be flashing red.

What is CNAME cloaking?

Under ordinary tracking setup, web trackers are typically third-party scripts embedded on the websites visited by users. Website example.com including a tracker.com served from another domain forms third-party content (technically not belonging to the website).

In the CNAME cloaking scheme, this is more complicated. The tracker is injected in the first-party context, the context of the visited website. A website *example.com* is embedding the content of the form *xxx.example.com*. But in reality, **this subdomain *xxx.example.com* is an alias for the tracker domain, the** *yyy.tracker.com*, a separate domain hosted at a third-party server. This scheme works thanks to a DNS delegation. Most often it is a DNS CNAME record. The tracker technically is hosted in a subdomain of the visited website. Employment of such a scheme has certain **consequences. It kind of fools the fundamental web security and privacy protections** - to think that the user is willfully browsing the tracker website. When a web browser sees such a scheme, **some security and privacy protections are relaxed**.

This has **substantial implications for web security and privacy**. Web browsers treat such a tracker xxx.example.com as legitimate first-party content of the visited website example.com. Such a measure unlocks many benefits, for example, **access to first-party cookies**. So there is **no longer a need to use third-party cookies**, which are increasingly restricted or phased out anyway. Technically speaking cookies may be set with a scope of the xxx.example.com (or even example.com), but they are of course sent to remote, third-party servers controlled by the tracker. This **circumvents many anti-tracking measures**.

Targeting Apple Safari browsers

In the case of a particular tracker provider, *Criteo*, we observe, identify and offer evidence of **targeted treatment of Apple’s web browser Safari**. We detected that upon the detection of the Safari web browser in use, Criteo’s tracking scripts reverted to the use of this custom CNAME cloak scheme. We saw this targeted **approach used deliberately. In this respect, our observation is unambiguous**. While we may only wonder about the motivations, we suspect that it is an intentional measure to track Apple Safari users, bypassing Safari's Intelligent Tracking Technology. We note:

“one tracker, namely Criteo, would only resort to first-party tracking for Safari users. Previously, this tracker was found to abuse top-level redirections [41] and leverage the HTTP Strict Transport Security (HSTS) mechanism to circumvent Safari’s ITP [22, 48].”

So it’s fair to say that Criteo is well-known in the privacy community.

Privacy leaks

This is the **worrying part**. Due to how web architecture works, the CNAME **scheme unlocks a way for broad cookie leaks**. When the user’s web browser connects to a website example.com, cookies previously set that was scoped “*to*” this particular domain are sent to the server by the web browser.

But since the CNAME scheme involves a scheme of the form tracker.example.com (the visited website example.com delegates a subdomain pointing to the tracker server), involving the subdomain of the websites, many unrelated, legitimate, cookies set for example.com are thus **sent to tracker.example.com as well**! In my 2014 study (1,2) I observe and validate the issue of such leaks. But a massive change in this space happened. The problem exploded. As the tip of the iceberg, we found **broad data leaks on 7,377 websites. Some data leaks happen on almost every website using the CNAME scheme** (analytics cookies commonly leak). This suggests that **this scheme is actively dangerous. It is harmful to web security and privacy**. It can be worse than that:

“The cookies sent in the POST bodies indicate that certain CNAME tracker scripts actively read and ex- filtrate cookies they may access on first-party sites. … We found 1,899 cookie leaks in request URLs to CNAME subdomains on 1,295 distinct sites. ”

In other words, we have evidence of both **passive and active data leaks**. Such a CNAME scheme is present on many websites. We found **cookies leaking on 95% of the studied websites**. Trackers receive cookies of other domains, not meant for them and totally unrelated to the tracker. This **happens without the user's awareness and consent**. It is actually ridiculous even, because why would the user consent to a third-party tracker receiving totally unrelated data, including of sensitive and private nature? We even spotted leaks of the cookies set by other third-party scripts. Some of the leaked cookies would then allow the CNAME tracker to track users across websites.

“certain CNAME-based trackers use third-party cookies for cross-site tracking and at times receive cookies set by other third-party domains, allowing them to track users across websites.”

What is worse, we also found that in some cases the **leaked cookies contained private or sensitive information**. The tracker servers obtain all such data. Specifically, we found instances of the following sensitive information leaked to the CNAME tracker.

- the user’s full name
- location
- email address
- the authentication cookie (farewell to web security)

This is far from nice. But it may be even worse.

Web security issues

CNAME scheme increases web security threat surface. That's for sure. It also leads to security vulnerabilities.

We found that many CNAME trackers are included over HTTP, not HTTPS. This may facilitate man-in-the-middle attacks, and generally, negatively impact the integrity of the main website visit because the JavaScript delivered over HTTP works in the context of the main site.

Vulnerabilities

Our paper reports two systemic risks reporting from the result of a free-of-charge security audit that is also part of the work. Security vulnerabilities, for example: session fixation, cross-site scripting.

“We found that the tracker endpoint did not adequately validate the origin of the requests, nor the cookie names and values. Consequently, through the functionality provided by the tracker, which is enabled by default on all the websites that include the tracker in a first-party context, it becomes possible to launch a session-fixation attack. For example, on a shopping site, the attacker could create their profile and capture the cookies associated with their session. Subsequently, the attacker could abuse the session-fixation vulnerability to force the victim to set the same session cookie as the one from the attacker, resulting in the victim being logged in as the attacker. If at some point the victim would try to make a purchase and enter their credit card information, this would be done in the attacker’s profile. Finally, the attacker can make purchases using the victim’s credit card, or possibly even extract the credit card information.”

I like this one even more:

“In this case, the tracker offers a method to associate a user’s email address with their fingerprint (based on IP address and browser properties such as the User-Agent string). This email address is later reflected in a dynamically generated script that is executed on every page load, allowing the website to retrieve it again, even if the user would clear their cookies. However, because the value of the email address is not properly sanitized, it is possible to include an arbitrary JavaScript payload that will be executed on every page that includes the tracking script. Interestingly, because the email address is associated with the user’s browser

and IP fingerprint, we found that the payload will also be executed in a private browsing mode or on different browser profiles. We tested this vulnerability on several publisher websites and found that all could be exploited in the same way. As such, the issue introduced by the tracking provider caused a **persistent** XSS vulnerability in several hundreds of websites."

It's a complex scheme that melts sites' web security.

CNAME use is rising

I first encountered the use of the CNAME cloaking technique, with negative consequences for user's privacy, back in 2014 (1, 2; largely overlooked at that time); while investigating real-time bidding. Today the number of providers of such 'services' increased. The technique is used on tens of thousands of websites. In our new research paper, we detect the use on over 10k websites. We note that as of today there are no consistent techniques that block such tracking automatically. Many typical approaches do not account for such a tracking technique. They are ineffective.

Defenses

Because today most anti-tracking works on the principle of filter lists (pattern matching of HTTP requests), the CNAME scheme effectively renders such defenses ineffective. There needs to be a different way to filter/validate/block such a scheme. As of today, from the major web browser vendors only Firefox offers technical capabilities enabling defenses by extensions, Safari has partial mitigations (cookie leaks still exist), and Brave browser offers defences based on blocking of CNAME leaks. Since uBlock version 1.25 under Firefox, the extension dynamically resolves hosts and sanitises such requests if a match is found. Such a measure does not work under Chrome because this web browser does not offer a way for extensions to dynamically resolve hostnames.

"Because Chrome does not support a DNS resolution API for extensions, the defense could not be applied to this browser. Consequently, we find that four of the CNAME-based trackers (Oracle Eloqua, Eulerian, Criteo, and Keyade) are blocked by uBlock Origin on Firefox but not on the Chrome version "

Most web users are vulnerable to such a scheme. It is an effective bypass of anti-tracking measures.

" To block CNAME-based tracking, blocklists would need to contain an entry for every website that uses the CNAME-based tracking service, instead of a single entry per tracker or match all DNS-level domains, leading to greater performance costs."

This would be a big efficiency problem (such a measure would not scale) and don't forget about the additionally emitted CO2 to the atmosphere of our planet.

Summary

As a former member of the W3C Technical Architecture Group, I must also say that I'm particularly worried about how this technique is misusing the way that the web works, specifically in the part where the cookies are leaking. In a way, this is the new low.

The analysis described in this post shows that the use of the emerging CNAME tracking technique is rising. The technique **evades anti-tracking measures. It introduces serious security and privacy issues**. Users can be hacked. User data is leaking , persistently and consistently, without user awareness or consent. This likely triggers GDPR and ePrivacy related clauses. Since it appears that Europe should have a functioning data protection framework, the public should well expect an enforcement action. We leave further action for the community.

Lastly, if there was an adequate privacy process, or a data protection officers in place, in the concerning places - the script providers, or the websites using the technique, at least some of the privacy risks would be found. This did not happen.

Did you like the assessment and analysis? Any questions, comments, complaints, or offers for me? Feel free to reach out: me@lukaszolejnik.com