Title: Vulnerability Disclosure Considered Stressful

Authors: Giovane C. M. Moura and John Heidemann

Journal: In: ACM SIGCOMM Computer Communication Review

DOI: https://doi.org/10.1145/3610381.3610383

Citation:

- Giovane C. M. Moura and John Heidemann. 2023. Vulnerability Disclosure Considered Stressful. SIGCOMM Comput. Commun. Rev. 53, 2 (April 2023), 2–10. https://doi.org/10.1145/3610381.3610383
- Bibtex:

```
@article{Moura23d,
author = {Moura, Giovane C. M. and Heidemann, John},
title = {Vulnerability Disclosure Considered Stressful},
year = {2023},
issue_date = {April 2023},
publisher = {Association for Computing Machinery},
address = {New York, NY, USA},
volume = {53},
number = {2},
issn = {0146-4833},
url = {https://doi.org/10.1145/3610381.3610383},
doi = {10.1145/3610381.3610383},
journal = {SIGCOMM Comput. Commun. Rev.},
month = {jul},
pages = {2{10},
numpages = {9},
keywords = {software vulnerability,
coordinated vulnerability disclosure, denial-of-service attacks}
}
```

Vulnerability Disclosure Considered Stressful

Giovane C. M. Moura
SIDN Labs and TU Delft
Arnhem and Delft, The Netherlands
giovane.moura@sidn.nl

John Heidemann
USC/ISI and CS Dept.
Los Angeles, California, USA
johnh@isi.edu

ABSTRACT

Vulnerability disclosure is a widely recognized practice in the software industry, but there is a lack of literature detailing the firsthand experiences of researchers who have gone through the process. This work aims to bridge that gap by sharing our personal experience of accidentally discovering a DNS vulnerability and navigating the vulnerability disclosure process for the first time. We document our mistakes and highlight the important lessons we learned, such as the fact that public disclosure can be effective but can also be more time-consuming and emotionally taxing than anticipated. Additionally, we discuss the ethical considerations and potential consequences that may arise during each step of the disclosure process. Lastly, drawing from our own experiences, we identify and discuss issues with the current disclosure process and propose recommendations for its improvement. Our ultimate aim is to provide valuable insights to fellow researchers who may encounter similar challenges in the future and contribute to the enhancement of the overall disclosure process for the benefit of the wider community.

CCS CONCEPTS

• **Security and privacy** → **Denial-of-service attacks**; **Vulnerability management**; **Social aspects of security and privacy**; • **Networks** → **Application layer protocols**;

KEYWORDS

Software vulnerability, Coordinated Vulnerability Disclosure, Denial-of-service attacks

1 INTRODUCTION

A vulnerability is defined as "a flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy" [49]. Many vulnerabilities are caused by software bugs, which are virtually unlimited in supply [6].

Software vulnerabilities are found relatively often. The Common Vulnerabilities and Exposures (CVE) system [12] has provided an index and reference system for publicly known vulnerabilities since 1999. The CVE index has grown a few thousand vulnerabilities per year to more than 30k since 1999 (Figure 1, covering 1999 through 2022). Even though the CVE lists a subset of all vulnerabilities, it illustrates the growth in public vulnerabilities.

Software testers, individual (white hat) hackers, and dedicated companies (for example, HackerOne [24]) often find vulnerabilities [55, 57]. Upon discovering a vulnerability, an individual has three options: keeping it private, selling it, or disclosing it. Keeping the vulnerability private or selling it in the vulnerability market will not fix the issue [3, 5, 31, 43, 48, 53, 56, 57]– the later raises serious ethical concerns [16], given it can empower attackers elsewhere. *Disclosure* is the most effective option for fixing the vulnerability and contributing to the public good, and is in line with the principles outlined in the ACM and IEEE Codes of Ethics for professional conduct [1, 28], as well as in the ethics guidelines for security incident response teams devised by FIRST [50].

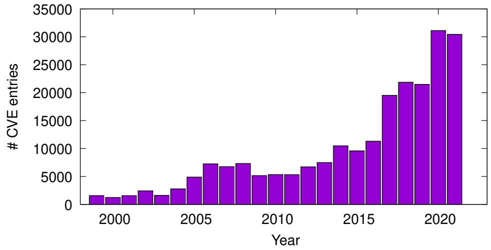

Figure 1: Yearly vulnerabilities listed by CVE. Source: [13].

Researchers occasionally encounter software vulnerability in their research. Below we describe one such vulnerability. To contribute to the public good and have the vulnerability fixed, we chose to disclose it, but we found the process confusing, poorly documented, and burdensome.

The vulnerability we found in 2020 was that many DNS resolvers, including Google Public DNS (GDNS) [21] and Cisco OpenDNS [41], could be exploited in DNS amplification attacks[37]. These attacks could potentially cause serious harm if exploited against large operators and country code top-level domains (ccTLDs) , such as the Netherlands' `.nl` and Japan's `.jp` . We summarize this vulnerability, known as TsuNAME, in §2.

We were surprised by the lack of first-hand scientific papers reporting the disclosure process. Prior work focused on notifying users of already-solved vulnerabilities who have not yet patched their systems instance (*e.g.,* [11, 51]). Our focus is on disclosing a new (zero day) vulnerability to both vendors and operators. Other studies have examined the side effects of vulnerability disclosure, such as the impact of software quality [46] and on attack frequency [4], but not on the hands-on experiences. We also found that there is not even consensus on disclosure *terminology*: private, public, responsible, full, and coordinated disclosures are unsettled terms in both academia and industry. We cover them in §3 and discuss their subtle differences.

This paper aims to fill the gap in knowledge about the hands-on experience of disclosing zero-day vulnerabilities. It is based on our own experience (§4) and our presentation on this topic at the RIPE83 meeting [35]. We made several mistakes and learned valuable lessons during this process, which we discuss in detail in §5. In our experience, the process of disclosing and fixing vulnerabilities turned out to be more demanding than we expected. In §6, we put forward two suggestions on how we, as a community, can improve this process. We hope that these lessons may assist other researchers

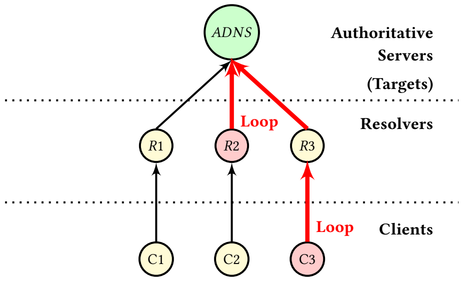

Figure 2: TsuNAME attack.

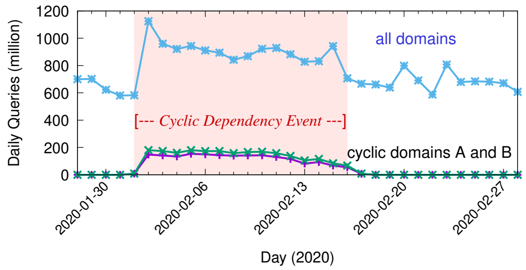

Figure 3: Traffic growth observed at `.nz` ccTLD caused by a TsuNAME vulnerability. Source: [37].

in their own future vulnerability disclosures and help to improve the disclosure and fixing processes.

2 TSUNAME VULNERABILITY

TsuNAME is a vulnerability in DNS [32] implementations where parts of DNS infrastructure to loop [37], stressing DNS servers with excessive load. TsuNAME affects DNS client and DNS resolvers, as shown in Figure 2. Vulnerable clients and resolvers end up sending non-stopping queries to authoritative DNS servers (ADNS), which are a type of DNS server that knows content from a DNS zone from memory and can answer queries from resolvers [26].

In Figure 2, we observe that clients and resolvers in light red are the source of loops. They continuously send queries in the presence of cyclic dependencies. C1-R1 is a non-vulnerable client-resolver pair and sends few queries to ADNS. Conversely, C2-R2 is vulnerable and a single query from C2 can cause R2 to loop, resulting in numerous queries. Our findings show that the Windows 2008R2 DNS server is vulnerable in this manner. Lastly, in the C3-R3 pair, it is the client that loops, leading to a large number of queries being sent to the resolver R3. If R3 does not cache the results, each client query will generate a new set of queries. The GDNS resolver was found to be vulnerable in this manner.

Ultimately, ADNS servers can then become overwhelmed and fail to serve real clients – configuring a denial-of-service (DoS) attack. While the TsuNAME vulnerability can be found in clients and or DNS resolvers, it does not weaken them or allow for data exfiltration: it can be exploited only to *overload* ADNS servers.

A typical vulnerability, such as Heartbleed [15], which affected OpenSSL cryptographic software library, requires developers (vendors) to fix the source code and operators to deploy the patches. Fixing the source code can be done by a single organization or individual, but patching thousands of servers and clients that use the vulnerable software can take much more effort and time [11, 51].

TsuNAME, however, is slightly different from these types of vulnerabilities. While it affected vendors such as GDNS and OpenDNS, the attacks could be used against any operators who had no relationship with these vendors. An attacker could harness the large capacity of GDNS and OpenDNS to overwhelm their ADNS victims.

When ADNSes fail, all zones under them may become unreachable, with potential severe consequences. In 2016, a large DNS provider (Dyn) suffered a large DDoS attack and became (partially) overwhelmed. Users reported reachability issues with “Twitter, Netflix, Spotify, Airbnb, Reddit, Etsy, SoundCloud and The New York Times” [42]. One could only wonder the impact of a ccTLD unavailability (*e.g.,* Germany’s `.de`), preventing citizens from doing online shopping, banking, and accessing government websites.

2.1 Root Causes

TsuNAME vulnerability caused resolvers to send large amounts of queries in the presence of cyclic dependencies. A cyclic dependency consists of misconfigured DNS NS records, which point to each other. (In DNS, NS records are used to indicate which servers are authoritative for a given DNS zone [32]).

Consider Listing 1 as an example. It shows a snippet of the `.org` DNS zone – which is hosted at the `.org` ADNS servers. In this example, if a client (Figure 2) wants to visit cat.org, if first needs to resolve its NS records, which are `ns[1,2].dog.com`. So the DNS resolver must then query the `.com` ADNS for this information.

```
cat.org NS   ns1.dog.com
cat.org NS   ns2.dog.com
```

Listing 1: `.org` zone sample

```
dog.com NS   ns1.cat.org
dog.com NS   ns2.cat.org
```

Listing 2: `.com` zone sample

Suppose the resolver retrieves the Listing 2 records from the `.com` ADNS servers: it says that the ADNS servers for `dog.com` are then under `cat.org`. This setup is *cyclic* dependent, given it has a DNS records loop.

When finding such a loop, we discovered that some clients and resolvers will send non-stop queries to ADNS servers, which may become overwhelmed: it caused a 50% traffic increase to the `.nz` ccTLD authoritative servers (Figure 3). We wondered what an attacker could achieve if a carefully-designed attack was executed.

How we found it: we first came across it by coincidence in a previous study [36], when comparing the DNS traffic characteristics to authoritative servers of the Netherlands’ `.nl` country-code top-level domain (ccTLD) and New Zealand’s `.nz` ccTLD. We found that GDNS – one of the largest public DNS services on the Internet had distinct traffic characteristics to `.nz` servers (§4.2.1 in [36]) compared to the `.nl` servers. The root cause was TsuNAME. We refer the interested reader to [37] for a detailed discussion.

3 VULNERABILITY DISCLOSURE FLAVORS

There are various types of vulnerability disclosure and there is currently no consensus in both academia and industry on their precise definitions.

Private disclosure is the practice of disclosing a vulnerability only to the vendor, with the expectation that it will be fixed. This method was commonly used in the past, but has been shown to be ineffective as researchers were often ignored by vendors and in some cases legally threatened [47]. It is important to note that private disclosure is not limited to a single vendor and can also be performed within a group, such as a group of trusted workshop attendees – which we did with TsuNAME (§4).

On the opposite end of the disclosure spectrum, we have *full* (or public) disclosure, where all information related to the vulnerability is made publicly, sometimes even before a patch is available – which can create potential problems. Full disclosure has been praised as the only way to bring public scrutiny to vulnerabilities [47] by removing the veil of secrecy from private disclosures.

Between these two extremes lies what is referred to as "responsible" or "coordinated vulnerability disclosure" (CVD). In this type of disclosure, a researcher first notifies a vendor about the vulnerability, just as in private disclosure. After a grace period, all information related to the vulnerability is publicly disclosed. As such, CVD can be seen as public disclosure with a short embargo – and the public aspect adds pressure for vendors to fix their software [47]. The length of the grace period varies; for example, Google Project Zero stipulates a 90-day embargo before full disclosure [23]. The US CERT (now CISA), in turn, advocates for 45 days [10], but this is not a hard limit, and it can be negotiated with the vendor [27] depending on the vulnerability. (In §6, we delve into the problems caused by the the absence of a well-defined timeframe).

Evolving terminology: over the last year, the industry has been shifting from the term "responsible" disclosure to CVD [27, 29, 39, 52]. *Responsible* implies a moral duty on whoever found the vulnerability, whereas the real responsibility lies with who created the vulnerability in the first place: the vendor itself [52]. *Coordinated* vulnerability disclosure is the preferred term, given that it removes the onus on the researcher and its moralistic label.

Notifying operators: one thing is to notify the vendor, and the other is to notify operators who use the vulnerable software that they must patch their system. While it may be far easier to fix a software, update all clients that use it may take years (*e.g.,* [11, 51]). As we show in §4, we notified both vendors and operators in *private* first – but also in private group sessions, where we disclosed privately to groups of people in trusted venues.

4 TSUNAME DISCLOSURE

In a typical vulnerability disclosure, a researcher would notify the vendor, who, in turn, would release a patch fixing it. Clients would then have to update their local software (if applicable), and their systems would be fixed after that. This was the case of Heartbleed [15] and Log4j Shell [25] (which affected Log4j).

TsuNAME, however, is a vulnerability on clients and resolver software that can be exploited against *third parties*: ADNS server operators (§2). As such, we needed to notify the resolver developers, operators, and ADNS operators who could become victims of attacks. As we shall see in §5, this took much more effort than we initially anticipated.

Figure 4 shows the timeline of our disclosure. As can be seen, we have three phases in this disclosure: Google disclosure, Group disclosure, and full, public disclosure. We address them next.

4.1 Google Public DNS private disclosure

In our analysis of the New Zealand TsuNAME event, we found that 99% of the queries were originated from GDNS [37]. We took that into account and decided first to notify the GDNS operators.

Instead of using Google's bug reporting system [20], we reached out to GDNS operators we knew personally via e-mail, on 2020-09-01. Our hope was that direct contact would reduce the time to fix the bug, compared to the official reporting channel (which turned out to be incorrect assumption). Our contacts acknowledged the issue and said they would look into it.

After several months without a resolution from GDNS, we decided to notify Google through their official bug report channel in the hopes of having the vulnerability fixed. We did this on 2020-11-26, almost three months after our initial email contact (Figure 4). This report was then logged as issue #174297111 within Google's bug tracking system. We also informed our private contacts at GDNS of this step.

Choosing only to notify GDNS first: While we may have inadvertently been unfair to other vendors at this stage, we prioritized based on the data we analyzed. We now recognize that this was a mistake, and that our lack of experience played a role in this decision. We should have notified all vendors we could reach, regardless of the data we collected. This would have been a more fair and equitable approach. We expand on the ethical consequences behind this choice in §5.2.

4.2 Operators and other vendors disclosure

Since the GDNS had not been fixed after our two private notifications, we felt that it was necessary to disclose the TsuNAME vulnerability to other vendors and potential third-party victims: the community of authoritative DNS server operators. We had developed a tool (`CycleHunter`) that could detect cyclic dependencies, and operators could use it to detect errors on their zones and prevent TsuNAME-based attacks. Up until that point, we had not set a public disclosure date, and it had been only private disclosure.

Contacting operators can be a challenging task, as it can be difficult to identify the appropriate individuals or organizations to notify. With so many operators on the Internet, it is not feasible to contact every ADNS server operator.

To assist operators and other vendors in preventing such attacks, we sought help from the DNS-OARC [14], a DNS operators community, and requested to conduct a group disclosure, by disclosing the vulnerability during an online meeting for DNS-OARC members only. We chose this venue because it is popular among DNS operators and can be trusted, and the operators are contractually obliged not to disclose the information further. Additionally, the authors have been long-term members of this community.

We scheduled a presentation for the OARC34 meeting for 2021-02-05 [18]. Using their bug tracking system, we informed GDNS of our plans to perform the group disclosure on this date, providing all the details and tools for the members attending the session.

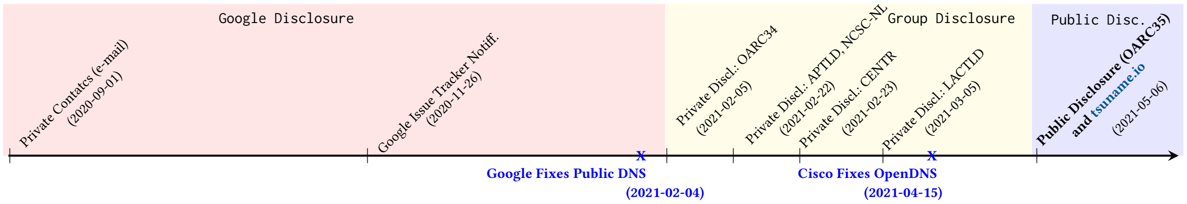
Google Disclosure
Group Disclosure
Public Disc.
Private Contatcs (e-mail) (2020-09-01)
Google Issue Tracker Notiff. (2020-11-26)
Private Discl.: OARC34 (2021-02-05)
Private Discl.: APTLD, NCSC-NL (2021-02-22)
Private Discl.: CENTR (2021-02-23)
Private Discl.: LACTLD (2021-03-05)
Public Disclosure (OARC35) and tsuname.io (2021-05-06)
X
Google Fixes Public DNS (2021-02-04)
X
Cisco Fixes OpenDNS (2021-04-15)

Figure 4: TsuNAME disclosure timeline

4.2.1 Google fixes GDNS. A few weeks before our scheduled group disclosure, our contacts at GDNS reached back to us and scheduled online meetings to discuss the issue. They had identified the root cause of the vulnerability: GDNS did not loop in the presence of cyclic dependencies, but downstream clients that used GDNS did (as C3–R3 in Figure 5).

Although GDNS did not loop, it did not cache the results of these queries, as DNS standards do not require it. The fix was to limit these new queries from GDNS to ADNS servers by caching the responses on GDNS resolver caches. Google operators deployed the fix on 2021-02-04 [37].

4.2.2 Group disclosure. The day after GDNS had been fixed, we disclosed the TsuNAME vulnerability to the OARC34 members. Following that, we decided to disclose it to more operator venues. We then disclosed it to events such as the APTLD, the Netherlands National Cybersecurity Center (NCSC-NL), the CENTR meeting, and LACTLD. Our goal was to increase operator awareness and assist them in preventing such attacks.

In the process, OpenDNS fixed their software on 2021-04-21 (they learned only on 2021-02-05 during the OARC 34 presentation). After that, most resolver developers and many authoritative server operators were aware of TsuNAME.

4.3 Public Disclosure

The final step in our disclosure process was public disclosure. We chose to make this disclosure on 2021-05-06, at the OARC35 meeting [19], which was roughly three months after our initial group disclosure at OARC34 and eight months after our initial report to Google. We created a website (https://tsuname.io) to provide technical reports and tools for operators to better understand and mitigate the vulnerability.

By the time of public disclosure, 248 days after our initial contact with Google, both GDNS and OpenDNS had been fixed. Additionally, other resolver vendors had released reports with regards TsuNAME and their software , including BIND [7], Unbound [40], and PowerDNS [44].

5 LESSONS LEARNED

None of the authors had previous experience with zero-day vulnerability disclosure. This lack of experience took a toll on the process, which we learned from and share next the most important lessons.

5.1 Public Disclosure Improves Security for Everyone

Upon discovering TsuNAME, we initially questioned the potential severity of its impact. It had already caused a 50% increase in total traffic on the .nz ccTLD. Its ease of weaponization raised concerns about the potential for large-scale disruption of top-level domains. The potential damage caused by disrupting a ccTLD was a source of concern (as discussed in §2 of [37]).

Given the potential impact of this vulnerability, we wondered why there had been no prior public reports of it. Was it because attackers had not yet discovered it, or were there other, more accessible and effective methods available? We faced an ethical dilemma: whether or not to disclose the vulnerability. Despite the risk of being perceived as alarmist, we ultimately decided to proceed with group and public disclosure, as outlined in §4.

Looking back now, it was the right choice. As we have seen in the vendor reports and fixes provided by GDNS and OpenDNS, it managed to fix the most vulnerable systems, and other vendors evaluated their own software for it. Thus, we recommend that the researchers perform vulnerability disclosure.

We learned in this case that private disclosure did not work as we anticipated (§4.1). We had later to use the official bug notification channel and set a public disclosure date, which may have helped to motivate GDNS to fix it.

5.2 Disclosure choices have ethical implications

When disclosing a vulnerability, a researcher may have the best intentions, but must be aware that choices must be made and each decision may have consequences for others.

No technology is value-neutral; the field of value-sensitive design (VSD) [17, 38] in Ethics help engineers to understand that their design choices have ethical, social, and cultural values implications for others. Disclosing a vulnerability has ethical implications that go beyond the technical aspects. A researcher may encounter multiple ethical dilemmas and choices during the disclosure process, and each decision should be carefully weighed considering the public interest and available resources (*e.g.,* time). We discuss some of these choices we made.

We believe that we made the *right* choice in disclosing TsuNAME. The vendors were able to fix the vulnerability, preventing it from being used in amplification attacks. Moreover, the choice made by ccTLD operators who discovered TsuNAME before us had consequences, as seen in the .nz event (as shown in Figure 3). While we do not intend to judge them (vulnerability disclosure is not their primary responsibility and the vulnerability was caused by other vendors, not the operators), it is important to recognize that their choice had unforeseen consequences for others.

In retrospect, we realize that we made an error in our choice of only notifying GDNS initially (§4.1). We now see that we should have treated all DNS resolver vendors equally and notified them simultaneously. Our wrong choice resulted in a delay in the mitigation of the vulnerability.

Another *right* choice we made was to not fully disclose the vulnerability without first notifying vulnerable vendors. This decision came naturally to us, as we understood the potential consequences of releasing all the details publicly. Doing so would have made it easy for attackers to weaponize the vulnerability, causing significant problems for potentially multiple companies, including GDNS. This almost occurred in another vulnerability disclosure, where an operator was given only a two-day notice before full disclosure [9] and did not receive the vulnerability details.

In summary, we urge researchers to consider the ethical dilemmas and implications of their decisions carefully. One way to navigate these dilemmas is to consult with their institution's Ethics boards, as the consequences of these decisions are not always immediately clear – a recent work provides guidelines for ethics boards on how to address CVDs [45]. By taking this approach, researchers can help ensure that their work serves the public interest and minimizes any negative impact on others.

5.3 Ask for help to reduce the burden

Disclosing TsuNAME required more time and energy than we initially expected. In addition to preparing presentations, we also created guides for operators and developers outlining the steps needed to reproduce TsuNAME.

In retrospect, we may have gone beyond what was necessary. It may have been sufficient to notify GDNS and other vendors and publicly disclose TsuNAME after a grace period. As we discussed in §5.2, our goal was to minimize the potential impact of TsuNAME attacks by notifying as many operators and vendors as possible. Being a part of these communities did aid in this process.

We conducted presentations at multiple venues (§4), in four different languages (Dutch, English, Portuguese, and Spanish), and presented virtually due to the COVID-19 pandemic from three continents. Despite not being able to reach every operator, we made efforts to reach regional forums where many operators were present.

Many operators and developers may be discouraged from disclosing vulnerabilities when it is not part of their daily duties as they may not have the time and energy to do so. In fact, after our presentations, two ccTLD operators reached out to us privately, reporting that they had experienced TsuNAME events before (§5.4). Additionally, many researchers may not wish to endure the attention and exposure that comes with public disclosure (§5.5), which can be overwhelming.

To manage the communication process, a researcher may seek help from a vulnerability disclosure coordinator. This coordinator can handle the responsibility of contacting vendors, relieving the researcher of this burden and the associated exposure. Organizations such as CISA, for example, offer assistance with this task [2].

5.4 You do not have the complete picture

During the Q&A session at the group disclosure at OARC34, two ccTLDs operators confirmed that they had previously experienced TsuNAME events from GDNS. The first operator, a European ccTLD, kindly shared their traffic statistics of their TsuNAME event. In contrast to the .nz incident, which saw a 50% increase in traffic, this operator experienced a ten-fold increase. In Figure 5, we can observe the operator's aggregate traffic, with each color representing the traffic to each authoritative server they operate. We can see a sharp increase in traffic beginning at 19:00 UTC and reaching a peak of 10 times their normal traffic, before drastically reducing after 11:00 UTC the following day when they manually removed the cyclic dependency from their zone.

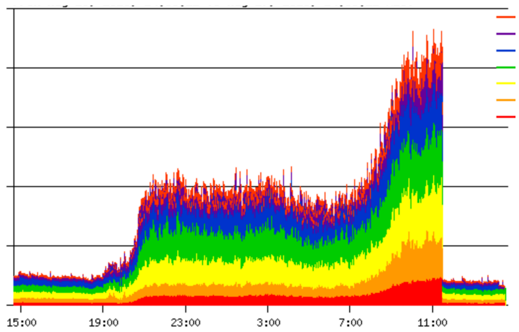

Figure 5: TsuNAME event at an EU-based ccTLD operator.

A second ccTLD operator in the Americas contacted us via email after the presentation and shared that they had been affected by similar events multiple times. They had also carried out private disclosure to their contacts at Google, but the issue persisted for years, causing frustration. Although we cannot verify their claims, this further illustrates that private disclosure may not be effective [47].

From their experiences, it is clear that TsuNAME was previously known, yet there was no public disclosure. Disclosing this vulnerability provided vendors with the necessary incentives to fix it. Therefore, we encourage researchers to disclose vulnerabilities, placing the public interest at the forefront and ultimately leading to a safer Internet for all.

5.5 Prepare for (stressful) reactions

Vulnerability disclosures are likely to garner significant attention. In our case, we received both positive and negative reactions.

Most *positive* reactions came from the vendors themselves (Google, Cisco OpenDNS) and the resolver developers (BIND, Unbound, PowerDNS). Although they were a small subset of folks we notified, they are the ones in the position to *fix* vulnerable software. For example, one GDNS operator thanked us for the "heads-up and not posting it online first".

We have also received *negative* responses: one operator accused us of fear-mongering and exaggeration. As we have discussed in §5.1, it was difficult to estimate the full impact of TsuNAME in the real world. However, after conducting our research and seeing the results, such as a 1000% traffic growth in the wild as shown in Figure 5, we deemed it necessary to publicly disclose the vulnerability in order to protect the public interest.

Another operator stated the problem has already been known: ("it's a shame that cycle prevention was not in the early DNS RFCs. Oh wait, it was" [8]). While it is true that previous RFCs had addressed the issue of cyclic dependencies [30, 32, 33], they did not fully cover it, which is why the vulnerability was still present. We took this feedback into consideration and used it to write an IETF draft on on how to make sure resolvers cache negative cyclic dependencies in DNS [34].

When disclosing a vulnerability, it is important to be prepared for the potential exposure and vulnerability that comes with it. Feedback or criticism can escalate quickly on social media platforms, such as Twitter, and it can easily be amplified. It is important to be prepared for this and to understand that not all feedback will be presented in a constructive manner. Additionally, it is important to recognize that the process of vulnerability disclosure can be emotionally taxing, and researchers may not always have the capacity or desire to handle it.

We understand that not all researchers may be comfortable with the attention and potential stress that can come with publicly disclosing a vulnerability. To avoid this, researchers may choose to disclose the vulnerability anonymously by using new e-mail accounts, aliases, and anonymizing tools. Alternatively, researchers can seek assistance from a vulnerability disclosure coordinator, as discussed in §5.3.

We also observed that not all parties were pleased with the disclosure process, however, this should not be the primary objective of disclosing vulnerabilities. The ultimate goal should be to rectify and secure vulnerable systems for the benefit of all users. Through the collaboration of all parties involved, we were ultimately able to achieve this goal.

6 IMPROVING THE DISCLOSURE PROCESS

We make two recommendations to improvement the process for disclosure when addressing vulnerabilities.

Clarifying vendor roles and timeframes: while existing documents primarily focus on the disclosure itself, there is limited discussion of expectations from vendors. After a vulnerability is disclosed to the vendor, it becomes their responsibility to determine when and how it will be addressed. However, what happens if the vendor refuses to fix the issue or indefinitely delays its resolution? In our case with Google, it took over 60 days for them to address the problem after we utilized their official notification system. Although Google's bug tracking system provided us with updates at every stage, the timeline for bug resolution remained unclear. Although they assigned priority level for TsuNAME was P2, with P0 being the most severe [22], this classification does not specify timeframes for issue resolution. Their statement is only an imprecise "[P2 issues] need to be addressed on a reasonable timescale" [22].

We were deeply concerned about the potential for other operators to become targets of DDoS victims while we were waiting for a fix, potentially making us complicit if if such attacks occurred. We encourage a clearer timeframe for resolution in vendor issue handling, in part to bound the risks (and stress) of the bug leaking or being discovered in parallel during an otherwise indefinite window.

Updating and endorsing CVD guidelines: The community would benefit from well-defined, succinct guidelines for vulnerability disclosure. These guidelines should protect the individuals who report the vulnerabilities and address the ethical considerations involved at each stage. Furthermore, they should define expected behaviors and timeframes for vendors. Presently, the absence of regularly updated, succinct, and widely endorsed documents leads to confusion, as we have personally encountered.

7 RELATED WORK

While we did not find research works detailing first-hand zero-day vulnerability disclosures, we found other works related to ours.

Unpatched systems notifications: after a vulnerability is disclosed and software patches are released, systems must be updated. In the case of Heartbleed, that meant updating thousands of servers worldwide. There have been previous studies of such notifications. For example, a previous study carried out notifications to 24k domain names affected by two types of web vulnerabilities [51]. They evaluated the efficacy of notification methods (e-mail, phone calls, and social media). Another work evaluated the efficacy of notifying ISP clients of Mirai-botnet malware [11]. In both cases, the focus was on the notification of users/clients and not zero-day disclosure to vendors and operators.

CVD impact: side-effects of CVD have also been evaluated. One study addressed whether software quality improved after CVD [46], while another investigated if attack frequency changed [4]. Ours deals with a notification of a single vulnerability, and with the hands-on experience of disclosure and not its aftermath beyond the expected patches.

CVD industry guidelines: the industry has guidelines on how to carry out CVD. The most comprehensive we found is provided by the CISA [27], which we recommend for any organization/individuals carrying out CVD. A more succinct version is provided by NCSC-NL [39]. They are both designed to help in the process of disclosure. We were unaware of such guides when we disclosed TsuNAME. A recent study (thus not available at our disclosure time) has put forth guidelines for vulnerability disclosure for researchers, their institutions, and ethical boards [45], and it has been adopted as their university-wide CVD policy [54].

Terminology: the industry and academia have been moving away from "responsible" disclosure to CVD – as shown by both Microsoft [52] and ISO standards [29]. CVD is a more neutral term. When we disclosed TsuNAME,we disclosed it under "responsible" disclosure, given that we were not aware of CVD at the time.

8 CONCLUSIONS

We have shown that vulnerability disclosure pays off: fixed software protects people elsewhere. However, the road to disclosure is not easy. We have shown our own mistakes and our ethical choices, and the disclosure can be more intense and burdensome than a researcher may expect.

We hope that our experience helps researchers navigate the disclosure process in the future. Moreover, we believe it is essential for the community to establish well-defined and widely accepted guidelines that ensure the protection of individuals who disclose vulnerabilities. These guidelines should also outline the specific responsibilities and timeframes for vendors to address and resolve vulnerabilities. By establishing such guidelines, we can collectively work towards enhancing software security for the benefit of all users.

Acknowledgements

We thank the anonymous CCR reviewers and Jeroen van der Ham for their valuable reviews of our paper drafts.

We are grateful for the support of individuals and organizations who have assisted us in the disclosure and repair processes. They have either provided us with a platform to present, contributed to enhancing our open-source tool, or have carried out the repairs themselves – and even sharing their experiences and data. Fixing TsuNAME turned out to be a community effort.

John Heidemann's work on this paper is partially supported by NSF/CISE award CNS-1925737 (DIINER).

REFERENCES

[1] ACM. 2023. ACM Code of Ethics and Professional Conduct. https://www.acm.org/code-of-ethics

[2] ACM. 2023. CISA Coordinated Vulnerability Disclosure (CVD) Process. https://www.cisa.gov/coordinated-vulnerability-disclosure-process

[3] Abdullah M Algarni and Yashwant K Malaiya. 2014. Software vulnerability markets: Discoverers and buyers. *International Journal of Computer and Information Engineering* 8, 3 (2014), 480–490.

[4] Ashish Arora, Anand Nandkumar, and Rahul Telang. 2006. Does information security attack frequency increase with vulnerability disclosure? An empirical analysis. *Information Systems Frontiers* 8, 5 (2006), 350–362.

[5] A. Arora and R. Telang. 2005. Economics of software vulnerability disclosure. *IEEE Security & Privacy* 3, 1 (2005), 20–25. https://doi.org/10.1109/MSP.2005.12

[6] Al Bessey, Ken Block, Ben Chelf, Andy Chou, Bryan Fulton, Seth Hallem, Charles Henri-Gros, Asya Kamsky, Scott McPeak, and Dawson Engler. 2010. A Few Billion Lines of Code Later: Using Static Analysis to Find Bugs in the Real World. *Commun. ACM* 53, 2 (feb 2010), 66–75. https://doi.org/10.1145/1646353.1646374

[7] ISC BIND. 2021. TsuNAME DNS Vulnerability and BIND 9. https://www.isc.org/blogs/2021_tsuname_vulnerability/.

[8] Randy Bush. 2021. it's a shame that cycle prevention was not in the early DNS RFCs. oh wait! it was. https://twitter.com/enoclue/status/1390388281020321793.

[9] Randy Bush. 2021. possible rsync validation dos vulns. https://mailman.nanog.org/pipermail/nanog/2021-October/216309.html.

[10] cert.gov. 2021. Vulnerability Disclosure Policy. https://vuls.cert.org/confluence/display/Wiki/Vulnerability+Disclosure+Policy.

[11] Orçun Çetin, Carlos Ganán, Lisette Altena, Takahiro Kasama, Daisuke Inoue, Kazuki Tamiya, Ying Tie, Katsunari Yoshioka, and Michel Van Eeten. 2019. Cleaning Up the Internet of Evil Things: Real-World Evidence on ISP and Consumer Efforts to Remove Mirai.. In *Proceedings of the 26th Annual Symposium on Network and Distributed System Security (NDSS '19)*.

[12] MITRE Corporation. 2022. CVE. https://cve.org/.

[13] MITRE Corporation. 2022. CVE List Downloads. https://cve.org/Downloads.

[14] DNS OARC. 2021. Introduction to DNS-OARC. https://www.dns-oarc.net.

[15] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, and J. Alex Halderman. 2014. The Matter of Heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (Vancouver, BC, Canada) *(IMC '14)*. Association for Computing Machinery, New York, NY, USA, 475–488.

[16] Serge Egelman, Cormac Herley, and Paul C. van Oorschot. 2013. Markets for Zero-Day Exploits: Ethics and Implications. In *Proceedings of the 2013 New Security Paradigms Workshop* (Banff, Alberta, Canada) *(NSPW '13)*. Association for Computing Machinery, New York, NY, USA, 41–46. https://doi.org/10.1145/2535813.2535818

[17] Batya Friedman, David G. Hendry, and Alan Borning. 2017. A Survey of Value Sensitive Design Methods. *Foundations and Trends® in Human–Computer Interaction* 11, 2 (2017), 63–125. https://doi.org/10.1561/1100000015

[18] Giovane Moura. 2021. OARC Members Only Session: Vulnerability Disclosure (DDoS). https://indico.dns-oarc.net/event/37/contributions/821/.

[19] Giovane Moura. 2021. Public Disclosure DNS vulnerability. https://indico.dns-oarc.net/event/38/contributions/849/.

[20] Google. 2022. Google and Alphabet Vulnerability Reward Program (VRP) Rules. https://bughunters.google.com/about/rules/6625378258649088/google-and-alphabet-vulnerability-reward-program-vrp-rules.

[21] Google. 2022. Google Public DNS. https://developers.google.com/speed/public-dns/

[22] Google Developers. 2023. Issue Tracker Concepts. Website. https://developers.google.com/issue-tracker/concepts/issues

[23] Google Project Zero. 2021. Vulnerability Disclosure FAQ. https://googleprojectzero.blogspot.com/p/vulnerability-disclosure-faq.html.

[24] HackerOne. 2022. #1 Trusted Security Platform and Hacker Program. https://www.hackerone.com/

[25] Raphael Hiesgen, Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. 2022. The Race to the Vulnerable: Measuring the Log4j Shell Incident. In *In Proc. of Network Traffic Measurement and Analysis Conference (TMA '22)* (Enschede, The Netherlands). IFIP.

[26] P. Hoffman, A. Sullivan, and K. Fujiwara. 2018. *DNS Terminology*. RFC 8499. IETF. http://tools.ietf.org/rfc/rfc8499.txt

[27] Allen D Householder, Garret Wassermann, Art Manion, and Chris King. 2017. *The CERT Guide to Coordinated Vulnerability Disclosure*. Technical Report. Carnegie-Mellon Univ Pittsburgh Pa Pittsburgh United States. https://resources.sei.cmu.edu/asset_files/specialreport/2017_003_001_503340.pdf

[28] IEEE. 2023. 7.8 IEEE Code of Ethics. https://www.ieee.org/about/corporate/governance/p7-8.html

[29] ISO/IEC. 2018. *ISO/IEC 29147:2018 – Information technology — Security techniques — Vulnerability disclosure*. Technical Report. ISO. https://www.iso.org/standard/72311.html

[30] A. Kumar, J. Postel, C. Neuman, P. Danzig, and S. Miller. 1993. *Common DNS Implementation Errors and Suggested Fixes*. RFC 1536. IETF. http://tools.ietf.org/rfc/rfc1536.txt

[31] David McKinney. 2007. Vulnerability Bazaar. *IEEE Security & Privacy* 5, 6 (2007), 69–73. https://doi.org/10.1109/MSP.2007.180

[32] P.V. Mockapetris. 1987. *Domain names - concepts and facilities*. RFC 1034. IETF. http://tools.ietf.org/rfc/rfc1034.txt

[33] P.V. Mockapetris. 1987. *Domain names - implementation and specification*. RFC 1035. IETF. http://tools.ietf.org/rfc/rfc1035.txt

[34] Giovane Moura, Wes Hardaker, John Heidemann, and Sebastian Castro. 2021. *Negative Caching of Looping NS records*. Internet-Draft draft-moura-dnsop-negative-cache-loop-00. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-moura-dnsop-negative-cache-loop/00/ Work in Progress.

[35] Giovane C. M. Moura. Nov 22, 2021. Responsible Disclosure. https://ripe83.ripe.net/archives/video/625/

[36] Giovane C. M. Moura, Sebastian Castro, Wes Hardaker, Maarten Wullink, and Cristian Hesselman. 2020. Clouding up the Internet: How Centralized is DNS Traffic Becoming?. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. Association for Computing Machinery, New York, NY, USA, 42–49.

[37] Giovane C. M. Moura, Sebastian Castro, John Heidemann, and Wes Hardaker. 2021. TsuNAME: exploiting misconfiguration and vulnerability to DDoS DNS. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Virtual, 398–418. https://doi.org/10.1145/3487552.3487824

[38] Lisa P. Nathan, Batya Friedman, Predrag Klasnja, Shaun K. Kane, and Jessica K. Miller. 2008. Envisioning systemic effects on persons and society throughout interactive system design. In *Proceedings of the 7th ACM conference on Designing interactive systems* (Cape Town, South Africa) *(DIS '08)*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/1394445.1394446

[39] NCSC-NL. 2018. *Coordinated Vulnerability Disclosure: the Guideline*. Technical Report. National Cybersecurity Center (NCSC-NL). https://english.ncsc.nl/publications/publications/2019/juni/01/coordinated-vulnerability-disclosure-the-guideline

[40] NLnetLabs. 2021. tsuNAME vulnerability and Unbound. https://nlnetlabs.nl/news/2021/May/10/tsuname-vulnerability-and-unbound/.

[41] OpenDNS. 2021. Setup Guide: OpenDNS. https://www.opendns.com/. https://www.opendns.com/

[42] Nicole Perlroth. 2016. Hackers Used New Weapons to Disrupt Major Websites Across U.S. *New York Times* (Oct. 22 2016), A1. http://www.nytimes.com/2016/10/22/business/internet-problems-attack.html

[43] Nicole Perlroth and David E. Sanger. 2013. Nations Buying as Hackers Sell Flaws in Computer Code. *New York Times* (Jul. 13 2013). https://www.nytimes.com/2013/07/14/world/europe/nations-buying-as-hackers-sell-computer-flaws.html

[44] PowerDNS. 2021. TsuNAME vulnerability and PowerDNS Recursor. https://blog.powerdns.com/2021/05/10/tsuname-vulnerability-and-powerdns-recursor/.

[45] Dennis Reidsma, Jeroen van der Ham, and Andrea Continella. 2023. Operationalizing Cybersecurity Research Ethics Review: From Principles and Guidelines to Practice. In *Proceedings EthiCS 2023*. Internet Society. https://doi.org/10.14722/ethics.2023.237352 2nd International Workshop on Ethics in Computer Security, EthiCS 2023, EthiCS ; Conference date: 27-02-2023 Through 27-02-2023.

[46] E. Rescorla. 2005. Is finding security holes a good idea? *IEEE Security & Privacy* 3, 1 (2005), 14–19. https://doi.org/10.1109/MSP.2005.17

[47] Bruce Schneier. 2007. Schneier: Full Disclosure of Security Vulnerabilities a 'Damned Good Idea'. https://www.schneier.com/essays/archives/2007/01/schneier_full_disclo.html

[48] Matthew Rosenberg Scott Shane and Andrew W. Lehren. 2017. WikiLeaks Releases Trove of Alleged C.I.A. Hacking Documents. *New York Times* (Mar. 7 2017). https://www.nytimes.com/2017/03/07/world/europe/wikileaks-cia-hacking.html

[49] R. Shirey. 2007. *Internet Security Glossary, Version 2*. RFC 4949. IETF. http://tools.ietf.org/rfc/rfc4949.txt

[50] FIRST Ethics special interest. 2023. Ethics for Incident Response and Security Teams. https://ethicsfirst.org/.

[51] Ben Stock, Giancarlo Pellegrino, Frank Li, Michael Backes, and Christian Rossow. 2018. Didn't you hear me?—Towards more successful web vulnerability notifications. (2018).

[52] MSRC Ecosystem Strategy Team. 2010. *Coordinated Vulnerability Disclosure: Bringing Balance to the Force.* Technical Report. Microsoft. https://learn.microsoft.com/en-us/archive/blogs/ecostrat/coordinated-vulnerability-disclosure-bringing-balance-to-the-force

[53] Vlad Tsyrklevich. 2015. Hacking Team: a zero-day market case study. https://tsyrklevich.net/2015/07/22/hacking-team-0day-market/.

[54] University of Twente. 2023. Coordinated Vunerability Disclosure. Website. https://www.utwente.nl/en/eemcs/research/ethics/coordinated-vulnerability-disclosure/

[55] Daniel Votipka, Rock Stevens, Elissa Redmiles, Jeremy Hu, and Michelle Mazurek. 2018. Hackers vs. Testers: A Comparison of Software Vulnerability Discovery Processes. In *2018 IEEE Symposium on Security and Privacy (SP)*. 374–391. https://doi.org/10.1109/SP.2018.00003

[56] ZERODIUM. 2022. The Premium Exploit Acquisition Platform. https:/zerodium.com.

[57] Mingyi Zhao, Jens Grossklags, and Peng Liu. 2015. An Empirical Study of Web Vulnerability Discovery Ecosystems. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. Association for Computing Machinery, New York, NY, USA, 1105–1117. https://doi.org/10.1145/2810103.2813704