Snail Mail Beats Email Any Day: On Effective Operator Security Notifications in the Internet

Max Maass
Secure Mobile Networking Lab
Technical University of Darmstadt
Darmstadt, Germany
mmaass@seemoo.tu-darmstadt.de

Marc-Pascal Clement
Secure Mobile Networking Lab
Technical University of Darmstadt
Darmstadt, Germany
mclement@seemoo.tu-darmstadt.de

Matthias Hollick
Secure Mobile Networking Lab
Technical University of Darmstadt
Darmstadt, Germany
mhollick@seemoo.tu-darmstadt.de

ABSTRACT

In the era of large-scale internet scanning, misconfigured websites are a frequent cause of data leaks and security incidents. Previous research has investigated sending automated email notifications to operators of insecure or compromised websites, but has often met with limited success due to challenges in address data quality, spam filtering, and operator distrust and disinterest. While several studies have investigated the design and phrasing of notification emails in a bid to increase their effectiveness, the use of other contact channels has remained almost completely unexplored due to the required effort and cost. In this paper, we investigate two methods to increase notification success: the use of letters as an alternative delivery medium, and the description of attack scenarios to incentivize remediation. We evaluate these factors as part of a notification campaign utilizing manually-collected address information from 1359 German website operators and focusing on unintentional information leaks from web servers. We find that manually collected addresses lead to large increases in delivery rates compared to previous work, and letters were markedly more effective than emails, increasing remediation rates by up to 25 percentage points. Counterintuitively, providing detailed descriptions of possible attacks can actually *decrease* remediation rates, highlighting the need for more research into how notifications are perceived by recipients.

CCS CONCEPTS

• **Security and privacy → Web application security**; Usability in security and privacy.

KEYWORDS

web security, notification study, information leakage

ACM Reference Format:
Max Maass, Marc-Pascal Clement, and Matthias Hollick. 2021. Snail Mail Beats Email Any Day: On Effective Operator Security Notifications in the Internet. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3465481.3465743

1 INTRODUCTION

Operating a website securely requires constant attention to keep both software and configurations up to date and to avoid security issues. It is inevitable that some operators will make mistakes, which can be dangerous and expensive. For example, the Equifax breach was caused by a missing software update [9], and even a well-publicized vulnerability like Heartbleed still saw 3 % of the Alexa Top Million websites remain vulnerable two months after disclosure [11].

Researchers have attempted to send notifications to operators of insecure [11, 19, 25, 26], compromised [3–5, 7, 29], or misconfigured [6, 16, 18, 20, 30] systems, finding increased remediation rates, but also problems with undeliverable messages [6, 7, 11, 18, 25, 26] and operator distrust [4, 5, 25, 30]. They also observed large numbers of systems that remained unfixed, even after multiple notifications.

We investigate the role of two factors in a notification campaign: the *medium* of the message, where we compare emails and postal letters, and the presence of *attack scenarios* in the message, where we describe attacks enabled by the reported issues. We also seek to avoid the reachability issues reported by previous studies by manually collecting contact information for all websites, thereby operating with the highest quality of contact information available. Our notifications also contain a link to a self-service tool where recipients receive additional information and can verify if their website is still vulnerable.

We conduct our study with $N = 1359$ operators of German websites suffering from unintentional information leakage that would allow attackers to gain access to detailed information about the software running on the server, cryptographic keys, or entire databases. These issues are easily remediated, and their remediation will not lead to incompatibilities. This is crucial, as operators may be hesitant to migrate away from old, insecure software versions because this would break compatibility with other software [28].

Our paper makes the following contributions:

- We compare the effectiveness of letters and emails in a randomized controlled notification experiment, using manually-collected address information to operate under best-case data quality assumptions.
- We investigate the effect of adding or withholding details about how the vulnerabilities could be used in a realistic attack, to act as an incentive to remediate.
- We provide notification recipients with a self-service tool to evaluate if their remediation attempts were successful, and monitor its use.

The rest of this paper is structured as follows: after discussing previous studies in Section 2, we describe our experimental setup in Section 3. We give an overview about the obtained results in Section 4 and offer an interpretation in Section 5. Finally, we conclude in Section 6.

ARES 2021, August 17–20, 2021, Vienna, Austria

2 RELATED WORK

Previous work has conducted notification campaigns about a variety of issues, including the security of websites [3, 7, 11, 19, 25, 26, 29, 30], or Domain Name System (DNS) servers [6], misconfigured systems leading to Distributed Denial of Service (DDoS) amplification [5, 16, 18], non-compliance [20], or malware infections [4].

Most studies attempted to reach the affected operators via emails to WHOIS or abuse contacts, or common aliases as defined in RFC 2142 [8]. Some also attempted to work with intermediaries like Computer Emergency Response Teams (CERTs) and vulnerability clearinghouses [6, 16, 18, 26], Internet Service Providers (ISPs) [5, 7], or Google [18, 19, 30]. Two studies evaluated more labor-intensive contact channels. Stock *et al.* used manually-collected contact information for a variety of channels like letters, phone calls, and social media, obtaining mixed results [25]. Maass *et al.* used manually-collected email and postal addresses and found letters to be significantly more effective than emails [20].

Many studies reported delivery problems [6, 7, 11, 18, 20, 25, 26], citing high bounce rates [6, 7, 26] and spam filters [25, 26] as hurdles for message delivery. Even if the messages were delivered, recipients often distrusted these unsolicited mails [4, 5, 20, 25, 30] or performed extra steps to validate them [4, 5, 20].

More detailed messages appear to increase remediation rates [7, 18, 29] and trust in the message [25]. Recipients also requested assistance in validating that the notified issue was fixed [6, 19, 20, 30]. However, one study reported that providing a verification tool did not have a large effect on remediation [6].

3 METHODOLOGY

In this section we present our vulnerability notification study. We introduce the vulnerabilities we used as a basis for our notification study and their respective detection technique, and the underlying dataset. We discuss the design of our study and partitioning of the test set. Finally we cover our monitoring system and online checking tool that was provided to the recipients. We close by describing our evaluation strategy and discussing the ethics of our research. Figure 1 provides an overview of the process.

3.1 Vulnerabilities

As a dataset for our vulnerability notification study, we collect a set of websites with different vulnerabilities that expose private information to the public due to misconfiguration of a webserver or unintentional placement of sensitive files in a public directory. We briefly describe the different vulnerabilities in the following.

Cryptographic Keys. Websites are frequently secured using public/private keypairs, either for use in encrypted Transport Layer Security (TLS) connections, or for authenticated remote access to the server using Secure Shell (SSH). The security of these schemes relies on keeping the private key secret. However, inexperienced system operators may place these sensitive files in publicly accessible locations on their webserver, e.g. `example.com/key.pem`.

Database Backups. Databases can contain sensitive information like customer data, passwords, or even payment information. It is thus imperative to keep them private. System operators frequently perform backups of these systems by serializing the data into files

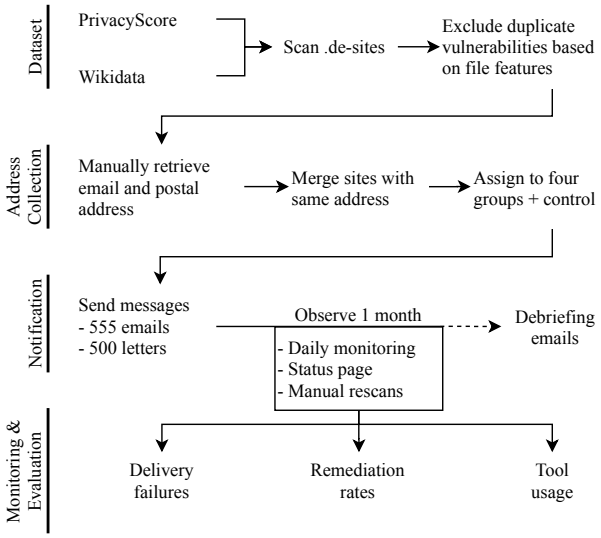


Figure 1: Methodology overview

using utilities like `mysqldump` [23]. The generated files may accidentally (or intentionally) be placed in publicly-accessible paths on webservers and thus have the potential to leak information to unauthorized parties.

VCS Repositories. Version Control Systems (VCSs) like Git or SVN are used to manage source code files while developing a system. They contain code and, in some cases, private configuration data. On a technical level, they use a hidden folder (`.git/` or `.svn/`) where they manage the history of the source code. If such a hidden folder is publicly accessible (as is the default for popular webservers), it can be used to retrieve the source code of the website [14] and thus potentially expose credentials or security vulnerabilities in the code. Prior research by Stock *et al.* also considered this issue and found it to be wide-spread [25].

Server Status Information. Web servers like Apache can be configured to display information about the server and open connections under special URLs like `example.com/server-info` [1] or `example.com/server-status` [2]. These status pages contain information that are not intended for the public and could be used to infer how much traffic a website is receiving, if it is running outdated software, or other sensitive information like session tokens.

PHPInfo Files. The scripting language PHP is widely used in web development. It contains a special command, `phpinfo()` [27], which will print out information about the PHP version, loaded extensions, and information about the environment and Operating System (OS) it is running on. While not directly harmful in itself, this information can be used to check if the server is running outdated software with known vulnerabilities, or leak secret information encoded in environment variables. It is thus advisable not to keep this information publicly available.

3.2 Dataset

We assemble our dataset by scanning a large set of domains with a custom vulnerability scanning system. We then manually collect contact information for all vulnerable sites. We describe the process in more detail in the following.

Data Sources. The dataset of domains that serves as starting point for the study is composed of two parts. The first part was provided by the *PrivacyScore* project [21], which, amongst several other checks, scans domains for exposed files. The operators provided us with information about approximately 700 exposed files spread accross 600 different domains, from which we include all domains under the `.de` Top-Level Domain (TLD) which are still vulnerable at the beginning of our study. This results in 248 vulnerabilities spread across 234 websites. In order to expand the dataset, we query the `official website` attribute of Wikidata.org and use the approximately 35 000 returned `.de`-domains as the second part of our dataset. The resulting combined dataset is used as the input to our vulnerability scanner. We do not use a dataset of popular websites like the Alexa toplist, as these websites are likely to have been notified about vulnerabilities before and thus might be prejudiced about such notifications. On the other hand, being listed in Wikidata.org guarantees a certain level of relevance. Additionally, toplists are known to have systemic problems [17, 24], and alternatives like the Tranco toplist [17] did not exist when the study was conducted.

We only consider `.de`-domains as we want to limit our notification study to a German-speaking population and also rely on the presence of imprints for contact retrieval, which are required by German legislation. We further exclude all German universities from our dataset, as they had already been notified by the PrivacyScore team in a different study [22]. Due to the origins of our dataset, most of the targeted websites belong to people and organisations of public interest.

Detecting Vulnerabilites. Our custom scanner consecutively requests a set of paths on each website with a `GET` request and downloads the first 20 kilobytes of the response if the response code is 200. Thereafter it verifies the contents with a regular expression. If it matches, we add the website and its exposed files to our dataset. The requested paths are chosen based on common filenames and in some cases customized to contain the name of the domain (e.g. `website.com/website.pem`). A full list can be found in Table 7 in the Appendix. To account for websites being temporarily unavailable, we scan the dataset two times on different days and discard all websites detected as non-vulnerable in both scans. For ethical reasons we include a link to a project website within the User-Agent header, which contains a description of our scans and guidance of how to opt out of our study. Our scans detected 1830 information leaks spread over 1736 different websites.

To account for vulnerabilities with the same source, e.g. hosted on the same server with a common configuration, we extract characteristic features from the exposed files and check whether they are shared with other sites. If this is the case, we exclude all but one from the study, as their remediations are related. In total, we identify 79 duplicate vulnerabilities that trace back to 23 common sources.

Gathering Contact Information. For each vulnerable website, we try to retrieve an email address as well as a postal contact address by manually checking the website for an imprint or a contact page. We prefer technical contacts over general-purpose addresses if both are given. Collecting both the email and the postal address for all domains also allows us to further deduplicate the domains in the dataset by merging related websites (run by the same operator, e.g. a publisher, or a music label that manages separate websites for their artists). In these cases, only a single message is sent that notifies the recipient about all vulnerabilities at once. This measure further reduces the likelihood that a single operator can have an undue impact on the overall results if it is managing several websites. In the following, we use the term *recipient* to describe a single contact (individual or organizational) that controls one or more websites.

3.3 Notification Groups

Our experiment uses two experimental factors: the delivery medium and the presence or absence of a detailed attack scenario description. All messages also contain a personalized link to a self-service tool (described in Section 3.5) that allows recipients to validate if the vulnerability persists and to trigger a manual scan to validate their remediation attempts. The full message texts are shown in Appendix A.

Notification Medium. Previous research showed that vulnerability notifications by email have only limited success in reaching the recipients [6, 7, 26] and encounter issues such as spam filtering [25, 26]. We thus evaluate the impact of using an alternative delivery medium, that is, compare emails to physical letters. We denote these notification classes as EMAIL and LETTER, respectively.

Email messages are sent using a purpose-specific email account linked to our research group (`web-survey@group.university.de`). The email account is hosted on the Google Apps for Education platform and thus uses the Google Mail infrastructure for message delivery. Emails are sent as plaintext, as Stock *et al.* [25] previously found HTML emails to be less effective.

Physical letters are sent using the official letterhead of our research group, and contain a scanned signature from one of the researchers. The letterhead also contains contact information for letters, fax, and emails, where it lists the same purpose-specific email account. It does not list a telephone number. See Appendix A for an example letter.

Attack Scenarios. As the risk posed by some of the vulnerabilities may not be obvious, we compare two different framings for our messages: The *baseline* message simply contains information about the detected information leaks, without discussing the potential impact. The second class of messages, denoted with an +ATK suffix (e.g., EMAIL+ATK), also contains a description of an *attack* enabled by the vulnerability, under the assumption that such an attack scenario illustrates the risks of the vulnerability and thus serves as an incentive to remediate.

Group Assignment. Adding an unnotified control group, we have a total of five experimental groups. Before assigning the groups, we scan all vulnerable websites again and remove those that have already been remediated. Recipients are then assigned randomly to the different groups, without considering address availability.

Table 1: Number of notified recipients per group and vulnerability (recipients can be affected by more than one vulnerability)

Group	Status	VCS	DB	Key	PHPInfo	Total
EMAIL	17	18	2	1	243	275
EMAIL+ATK	13	16	3	0	253	280
LETTER	17	19	3	1	250	287
LETTER+ATK	14	18	2	0	180	213
CONTROL	21	18	4	0	269	304
Total:	82	89	15	2	1196	1359

Recipients assigned to a medium for which we did not find an address are not contacted and not considered in later parts of the evaluation, thereby slightly reducing the sample size, but avoiding self-selection bias.

As remediation behavior may differ for different vulnerabilities, the different vulnerability classes are stratified between the groups. Table 1 shows the final distribution of vulnerabilities in the groups, considering only recipients for which the correct address type is available (i.e., that were actually notified). Due to the limited number of letters we can send, LETTER+ATK has fewer members than the other groups. We will consider the possible effects of this imbalance between the groups in Section 5.2.

Experiment Timeline. After monitoring the websites for five days to be sure the vulnerabilities persisted, we finalize the groups on June 10th, 2018, and send the letters on June 11th. To account for the higher delivery times, we hold off on sending the emails for two days, finally sending them on June 13th. Due to the high effort and cost of sending postal mail we do not send any reminder messages. We monitor remediation for one month before finishing the experiment. For ethical reasons, we then (re-)notify all recipients that have not remediated yet by email, including those in the CONTROL group, informing them that are (still) vulnerable to give them an opportunity to remediate.

3.4 Monitoring

Each night we initiate a check of all websites that we still consider to be vulnerable. If our scan shows a vulnerability as remediated, we repeat the scans for four days to confirm that the reading was not caused by a transient server or scan error. We say that a recipient has remediated when *all* vulnerabilities on *all* of their websites are fixed.

3.5 Online Checking Tool

In order to ease remediation for the recipients of our vulnerability notifications, we provide an online status page that can be accessed with a personalized link included within the notification (cf. Figure 2). It lists the vulnerabilities, each with the current remediation status indicated by a simple traffic light scheme. A *red* dot indicates the vulnerability has been detected by the last daily scan, while a *yellow* dot indicates the last check was negative, but we have seen the vulnerability within the last five days. After five consecutive

Schwachstellen-Status

Betroffene Webseite: http://[redacted].de/
Eine Schwachstelle wird als behoben betrachtet, wenn sie für mindestens fünf Tage nicht mehr festgestellt werden konnte.

http://[redacted].de/dump.sql
Status: *Behoben*

Über diese Adresse lässt sich eine Sicherung Ihrer Datenbank herunterladen. Obwohl wir deren Inhalte nicht im Einzelnen überprüft haben, lässt sich davon ausgehen, dass darin nicht für die Öffentlichkeit bestimmte Inhalte enthalten sind.

Dieses Problem können Sie ganz einfach beheben, indem sie entsprechende Datei auf dem Server in ein Verzeichnis verschieben, das nicht zu Ihrer Webseite gehört und nicht über das Internet erreichbar ist. Am besten wäre es, wenn Sie Datensicherungen gar nicht erst auf dem gleichen Server wie Ihre Webseite speichern, sondern diese auf Ihren lokalen PC herunterladen und dann auf dem Server löschen. Sie finden die Datei auf dem Server im Verzeichnis Ihrer Homepage unter dem gleichen Namen, der oben in fett angegeben ist.

http://[redacted].de/info.php
Status: *Behoben*

Über diese Adresse lassen sich Informationen über verwendete Software und deren Versionen, als auch interne Konfigurationsdetails Ihres Servers abrufen, die aus Sicherheitsgründen nicht öffentlich verfügbar sein sollten.

Dieses Problem können Sie ganz einfach beheben, indem sie entsprechende Datei von Ihrem Webserver entfernen. Sie dient nur zum Überprüfen der Installation und wird im laufenden Betrieb nicht benötigt. Sie finden die Datei auf dem Server im Verzeichnis Ihrer Homepage unter dem gleichen Namen, der oben in fett angegeben ist.

Jetzt erneut überprüfen

Das Projekt

Im Rahmen einer Bachelorarbeit haben wir tausende Webseiten auf typische Fehlkonfigurationen überprüft, die möglicherweise eine Schwachstelle darstellen. Wir benachrichtigen nun die im Impressum genannten Betreiber und untersuchen, wie diese auf die Hinweise reagieren.

Kontakt

Max Maass
SEEMOO / TU Darmstadt
Mornewegstraße 32
64293 Darmstadt

Fax +49 6151 16 - 25471
E-Mail web-survey@seemoo.tu-darmstadt.de

Datenschutz

Figure 2: The German-language status page

negative checks the dot turns *green*. Each vulnerability is accompanied with information about its impact and guidance on how to fix the respective issue.

To further help the recipients to verify the success of the remediation, the status page offers the possibility to manually trigger a check every 15 minutes, which will immediately switch the color of the respective vulnerability from *red* to *yellow* if it has been fixed. Recipients are only able to scan their own website(s), as identified by the token included in their personalized link. In order to evaluate the adoption of this status page, we record timestamps of the page visits and manually triggered scans.

Table 2: Reachability of the recipients per contact group

Group	Assigned	No Contact		Bounced		Reached		Unknown	
EMAIL	302	27	(8.94 %)	4	(1.32 %)	76	(25.17 %)	195	(64.60 %)
EMAIL+ATK	302	22	(7.28 %)	6	(1.99 %)	74	(24.50 %)	200	(66.23 %)
LETTER	304	17	(5.61 %)	3	(0.99 %)	97	(32.01 %)	187	(61.51 %)
LETTER+ATK	224	11	(4.89 %)	3	(1.34 %)	58	(25.78 %)	152	(67.86 %)
Sum	1132	77	(6.80 %)	16	(1.41 %)	305	(26.94 %)	734	(64.84 %)

3.6 Evaluation

To evaluate the effectiveness of our notifications, we measure the remediation rates over time. Only notified recipients are considered (i.e., those that were assigned to a medium for which no contact information was available are not counted into the total number). Not all email servers send a notice if they discard a message as spam, leading to an unknown number of silently discarded messages for the EMAIL and EMAIL+ATK groups. To avoid introducing any biases in comparison to the letters, we do not attempt to exclude recipients where message delivery failed from the evaluation, thereby slightly lowering reported remediation rates compared to studies that exclude these messages.

As previously described, a website counts as remediated if all of its vulnerabilities are remediated, and a recipient counts as having remediated if all of their websites are remediated. Considering recipients instead of websites ensures that all recipients make the same contribution to the overall remediation rates, regardless of the number of websites they control.

However, such an evaluation will only give us information about how *our sample* behaved. As some combinations of factors result in very small samples sizes, we want to estimate how much variation we could expect, if we were to repeat the experiment with another dataset of websites with the same characteristics. For this, we turn to *bootstrapping*. Bootstrapping allows us to approximate the variation in the results we would expect if we were to repeat the experiment many times with similar samples of websites. It can thus serve to quantify how much uncertainty remains in our results.

Mathematically speaking, our evaluation considers the empirical distribution of remediation vs. non-remediation, F^*, for our sample of n recipients, $x_1, \ldots, x_n$, drawn from the base distribution F (i.e., the distribution we would have obtained, had we performed a notification experiment with *all* affected recipients in Germany). We can use this sample to estimate the *variation* of a statistic u computed over F. For this, we take a sample with replacement of size n from the (known) *empirical* distribution F^*, denoted $x^* = x_1^*, \ldots, x_n^*$, and compute a statistic u^* over it. According to the *bootstrap principle* [12] (as described in [15]), the variation of u^* approximates the variation of u well. We can thus compute many resamples x^*, compute u^* for each of them, and then compute the measure of choice for the variation from these results.

While this technique has some limitations (in particular, the implied assumption that the original sample was representative for F, and that $x_1, \ldots, x_n$ are independent), it can serve to give an indication of how much we would expect the statistic to vary. We use this technique with 10 000 iterations to compute the 1st and 3rd quartile in addition to the median of the remediation rates on each day. While this cannot repair issues caused by very small samples sizes, it serves to indicate how imprecise we can expect our results to be due to them.

3.7 Ethical Considerations

Large-scale vulnerability scanning operates in a legal and ethical gray area. We avoid collecting sensitive information with our scans by limiting the amount of data we download, and discarding the data after the end of the study. Through this, we also avoid putting any undue strain on the infrastructure of the site operator. Our scanner identifies itself with a custom user agent and a reverse DNS entry for the IP of the scanning machine, and offers website operators a way to opt out of the study. The sent messages state that they are part of a study, and contain our contact information to allow recipients to opt out as well. All unnotified recipients for whom contact information is available are notified by email after the end of the study to give them an opportunity to remediate.

At the time of the study, our institution did not require ethics approval for this type of research. While we thus did not seek out an ethical review for this study, we successfully obtained approval for a different study with a substantially similar setup that employed similar safeguards. We also discussed the study with legal experts to ensure its legality in our jurisdiction.

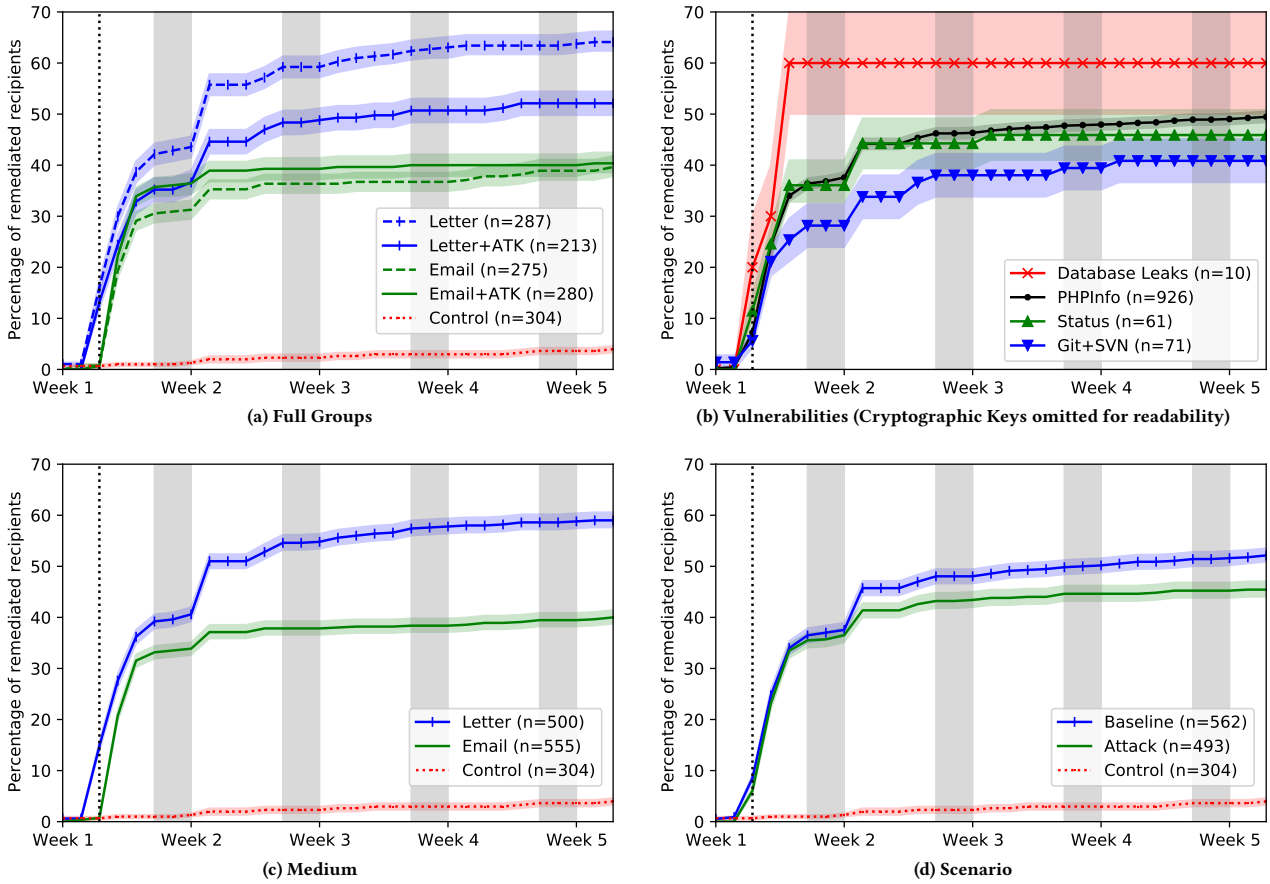
4 RESULTS

In this section, we investigate the effect of our notifications, the use of our self-service check tool, and briefly discuss the interactions with the recipients. These results will be interpreted in more detail in Section 5.

4.1 Notifications

Table 2 gives an overview of the delivery success of the notifications. In total, 77 (6.8 % of the original 1132 recipients, not counting the 304 in the control group) were assigned to a medium for which no address could be found, and thus were not notified. 10 emails (1.7 %) and 6 letters (1.1 %) could not be delivered and were returned to the sender. The true number of undelivered emails may be higher, as spam filters may have silently discarded messages. At least 305 messages (26.9 %) were read (i.e., we received a non-automated response or the self-service tool was accessed), which leaves 734 messages (64.8 %) in an unknown state.

Overall Remediation Rates. Table 3 shows the median remediation rates for the different groups, and their first and third quartiles. Overall, between 39.3 (EMAIL) and 64.3 % (LETTER) of recipients



(a) Full Groups

(b) Vulnerabilities (Cryptographic Keys omitted for readability)

(c) Medium

(d) Scenario

Figure 3: Median, 1st and 3rd quartiles for the remediation rates in different experimental groups, vulnerabilities, mediums, and presence of attack scenarios. Gray areas denote weekends, dotted line shows when the emails were sent.

remediated all issues with their website within the month of the study, depending on the notification channel and presence of the attack scenarios. All groups outperformed the control group by a large margin, which had a remediation rate of only 4.3 %. Figure 3a shows that most remediations take place within two weeks, after which only comparatively few additional vulnerabilities are fixed.

Communication Medium. Overall, letters markedly outperformed the email group (cf. Figure 3c), showing remediation rates of 59.0 % compared to the 40.0 % achieved by sending emails. After a week, almost half the letter recipients had already remediated, compared to only a third of those notified by email.

Attack Scenarios. The impact of including attack scenarios is less pronounced and, interestingly, different across mediums—while the attack scenario slightly increased remediation for emails, it actually greatly *decreased* remediation for letters (cf. Figure 3a), reducing the remediation rate by 12.4 percentage points. When considering vulnerabilities separately (cf. Table 4), we see that this effect is dominated by the PHPInfo group, which showed a large drop in remediation rates, loosing 8.3 percentage points when including descriptions of attacks. Interestingly, this effect is not spread evenly: the EMAIL+ATK and EMAIL groups have almost identical performance for the PHPInfo vulnerability (40.7 vs. 41.1 %), while LETTER+ATK has a much worse performance than LETTER, dropping from 65.2 to 51.1 % remediation at the end of the study timeframe. The other vulnerability classes actually saw increased remediation rates when adding the attack scenarios (between 6.3 and 10.7 percentage points), although the small sample sizes lead to a large spread in the quartiles, which overlap heavily. In aggregate, this leads to messages featuring the attack scenario achieving a remediation rate of 45.3 %, and being outperformed by their less explicit counterparts with 52.0 % remediation (cf. Figure 3d).

Vulnerability Type. Figure 3b shows the remediation rates of the different vulnerability types, considering only those not assigned to the control group. Database leaks show the highest remediation rates with 60.0 % ($n = 10$), however, their low overall number leads to limited expressiveness. PHPInfo leaks were fixed by 48.3 % of recipients ($n = 926$), with the server status ($n = 61$) and VCS ($n = 71$) misconfigurations following with 45.9 and 40.8 %, respectively. The figure omits the publicly available cryptographic keys ($n = 2$) for readability—one of them was removed by the end of the study,

Table 3: Median, 1st (Q1) and 3rd (Q3) quartiles of bootstrapped remediation rates in percent for different groups at the end of the study timeframe

Group	n	Median	Q1	Q3
EMAIL	275	39.3	37.5	41.5
EMAIL+ATK	280	40.4	38.2	42.1
LETTER	287	64.3	62.6	66.1
LETTER+ATK	213	51.9	49.5	54.2
All emails	555	40.0	38.4	41.3
All letters	500	59.0	57.4	60.4
All baseline	562	52.0	50.8	53.5
All +ATK	493	45.3	43.7	46.8
PHPInfo	926	48.3	48.3	50.5
VCS	71	40.8	36.6	45.1
Status	61	45.9	41.0	50.8
Database	10	60.0	50.0	70.0
Keyfile	2	50.0	50.0	50.0
CONTROL	304	4.3	3.6	4.9

while the other remained available, leading to a remediation rate of 50 %.

Effect of Reachability. The obtained results, especially for the message medium, raise the question if we are only measuring how well messages are delivered, or if the medium also has an effect outside of the rates of successful delivery. We thus repeat the evaluation with only those recipients that either used our self-service tool or from whom we received a non-automated response. Both of these indicate that the message was read. By its nature, such a sample is heavily self-selected and unrepresentative, but it may serve as an indication what factors influence the most motivated recipients (i.e., those that either contact us or use our tools).

In this sample of 305 recipients, we observe overall high remediation rates of 85.3 % for emails and 90.3 % for letters (cf. Figure 5 in the Appendix). Both versions of the messages (baseline and attack) achieve almost identical average performance (88.4 vs 87.1 %, respectively), but when considering all four combinations of medium and attack scenario, we still see that the attack scenario seems to help emails and hurt letters, although the gap has shrunk. The trends for the different vulnerability classes remain similar as well.

4.2 Self-service Tool

The self-service tool was accessed by 266 (25.2 %) recipients, with 192 (18.2 %) performing a manual scan. The distribution over the experimental groups is shown in Table 5. 65.8 % of visited status pages were accessed only on one day, although some were viewed on up to 15 separate days (median: 1, Q1: 1, Q3: 2). Similarly, while 37 % of tool users required only a single scan, some triggered up to nine scans (median: 2, Q1: 1, Q3: 3). 116 recipients (11 % of the total and 60.4 % of scan users) scanned their website after remediating to validate that their remediation attempt was successful.

Use of the tool and remediation seem to be linked—90.2 % of recipients that opened the tool and 95.8 % of those that triggered a scan before attempting a remediation successfully remediated the issue(s) afterwards. We note that this does not imply that the tool *caused* the remediation, as the users were self-selected (recipients that clicked the link clearly received the notification and trusted it enough to open a link, which makes them much more likely to remediate, even without the tool). We thus cannot make any statements about the effect of the tool on remediation.

Table 4: Median and quartiles of bootstrapped remediation rates for different vulnerability types at the end of the study timeframe, with and without attack scenarios. (Database and Keyfile omitted due to low sample size)

	Group	n	Median	Q1	Q3
Baseline	PHPInfo	493	53.3	51.9	55.0
	Status	34	41.2	35.3	47.1
	VCS	37	37.8	32.4	43.2
Attack	PHPInfo	433	45.0	43.4	46.7
	Status	27	51.9	44.4	59.3
	VCS	34	44.1	38.2	50.0

Table 5: Percentage of recipients who viewed and used the tool before (B) and after (A) remediation

Group	$View_B$	$View_A$	Use_B	Use_A
EMAIL	23.6 %	13.5 %	13.5 %	7.3 %
EMAIL+ATK	22.1 %	15.4 %	15.0 %	11.4 %
LETTER	25.2 %	22.0 %	18.5 %	15.0 %
LETTER+ATK	22.0 %	15.4 %	15.9 %	9.8 %

Table 6: Number of contacted recipients and non-automated responses by group and medium

Group	n	Email	Phone	Fax	Letter	Sum
EMAIL	275	29	3	0	0	31
EMAIL+ATK	280	28	2	0	0	30
LETTER	287	25	3	1	0	29
LETTER+ATK	213	13	1	0	1	15
All	1055	95	9	1	1	105

4.3 Communication with Recipients

We received responses from 105 out of the 1055 contacted recipients, not counting bounces and autoreplies (a detailed overview is given in Table 6). Most of the respondents were grateful, only two were hostile, interpreting our messages as either unsolicited advertising or fraud. To these, we sent clarifications and offered not to contact them again. 86 respondents stated that the problem had been remediated or that the responsible person had been instructed to fix it. Three of these still had unremediated PHPInfo issues at the end of the study timeframe. Some also explicitly mentioned using the tool we provided and finding it helpful. We did not receive any

opt out requests, only one unspecific abuse notification directed at our network provider.

The majority of respondents sent emails, but we also received one letter and one fax. Interestingly, nine recipients chose to contact us via phone calls. As we did not provide a phone number in the notification messages, they checked the website of the university and research group to find phone numbers, with one person calling the central switchboard of the university and being forwarded via multiple intermediaries until they reached the responsible researcher. We thus consider these numbers to be a lower bound, as some may not have been able to reach the right person in the end. They often stated that they mistrusted the message and wanted to verify its authenticity using a different channel.

Some recipients asked if we were aware of other issues with their website, or if we could scan additional websites under their control. These, we referred to the scanning service PrivacyScore.org [21], which performs similar checks for information leakage.

5 DISCUSSION

In this section, we review and interpret our results, discuss the limitations of our study, and identify areas for future research into effective notifications.

5.1 Effectiveness of Notifications

Overall, 48.9 % of notified recipients remediated, compared to 4.3 % of the control group. This demonstrates that our notifications were effective at increasing remediation. However, the different experimental groups show a large spread of remediation rates, ranging from 39.3 to 64.3 %. This indicates that the different factors of the notification can have a large impact on remediation. We discuss these factors in more detail here.

Manual Address Collection Improves Deliverability. Previous studies attempting to contact website owners directly [7, 20, 25, 26, 30] frequently struggled with delivery problems, with many studies showing bounce rates of over 10 % for WHOIS contacts [25, 26] (although Zeng *et al.* reported only 3 % [30]) and over 50 % for standard aliases like abuse@ [7, 26].

Two prior studies used manual address collection to overcome this problem. Stock *et al.* reported no email bounces, but a bounce rate of 26.8 % from their letters [25]. The latter may be related to the international nature of their study, while our study was limited to Germany, where system operators are legally required to disclose a functional postal address. However, Maass *et al.* reported bounce rates of 3.5 and 5.8 % for letters and emails in a similar study of German website operators [20], exceeding the 1.1 and 1.7 % observed in this study. This indicates that other factors also influence the delivery success.

Overall, the labor required for manual address collection may be difficult to scale to very large notification campaigns, although it could be justified for smaller or important notifications. Long-term, increased adoption of standardized ways of providing contact information for security notifications, like the proposed security.txt standard [13], is needed to facilitate more reliable automated notifications.

Letters are Effective. Letters provided a large boost in remediation compared to emails, with an increase of almost 20 percentage points. This may be related to a higher *a priori* trust into postal messages, as (at least in Germany) this communication channel is less often abused for spam and scam messages compared to emails. It may also be partially related to the more reliable delivery and lack of spam filters in the postal system. However, even when considering only recipients that reacted to our message, letters still show higher remediation rates than emails, indicating that at least some of their increased effectiveness cannot be attributed to the higher delivery success.

Once again, this increased remediation rate comes at a cost—sending 500 letters cost around 400 €, and around five hours for the printing and manual enveloping of the messages (although the latter could be avoided through the use of commercial mailing services or machines). However, as with the manual collection of contact information, the added expense could be justified for critical notifications.

Our results are in agreement with those by Maass *et al.*, who observed an increase in remediation rate of 11.2 percentage points when switching from emails to letters [20]. At first glance, both results seem to conflict with prior results by Stock *et al.*, who reported only a slight increase in remediation rates for letters compared to fully-automated email notifications [25]. However, these numbers should not be compared directly, as their dataset contained only operators that did not react to an initial automated message.

Verifiability Fosters Trust. Gaining the trust of the recipients and convincing them that the message is legitimate is an important challenge when sending unsolicited notifications. While we did not specifically ask about it, several recipients mentioned that recognizing the name of the sending university helped overcome their inherent distrust. This is in line with previous results by Maass *et al.*, who reported similar results [20]. However, some recipients still wanted to ensure that the message was authentic (and not just printed on an official-looking letterhead). These recipients invested considerable effort in searching for a phone number and calling the senders. This matches previous studies that reported recipients reaching out to verify the authenticity of the messages [4, 5, 20].

Similarly, providing a tool for recipients to verify the claims made in the message can help to convince the recipients that the message is correct, especially for non-technical recipients. While previous research found that providing a tool did not significantly increase remediation rates [6], a well-designed tool with documentation about the issue and how to verify and remediate it could increase trust and decrease support requests. Previous studies reported recipients requesting [6, 19, 30] and using [20] such tools, which we also observed in our study.

Tangible Explanations have Limited Impact. We saw that adding illustrative attack scenarios seems to have had almost no effect on remediation for the EMAIL groups, and have actively reduced remediation rates for the LETTER+ATK group compared to the LETTER group for the PHPInfo vulnerability. This result is surprising, as previous research has shown that more comprehensive messages usually increase remediation rates [7, 18, 29] and trust [25].

We can only speculate about the reasons. It may be that the recipients perceived the attack scenarios as an attempt to pressure

them into action, which may evoke associations with spam or scam messages. This would have lowered their trust in the message, and thus their willingness to act upon it. However, it is intuitively unclear why this should only be the case for letters, and not for emails.

It may also be that the expanded explanations actually *detracted* from the perceived urgency, as they provided a more nuanced view of the risk, instead of a blanket statement that the data should not be accessible for security reasons (cf. Appendix A). This idea is supported by the fact that only the PHPInfo group saw decreased remediation rates, while other groups saw increases (cf. Table 4). However, due to the overall low number of samples in these groups and commensurate large quartile ranges, the statistical basis for claims about their effectiveness is weak. And again, it is unclear why this would only affect the letters, but not the emails that used identical wordings.

Finally, it may also be related to recipients doing their own research: when performing a web search for "phpinfo dangerous", some results claim that exposing the information is discouraged, but in many cases not actually dangerous. However, it is unclear why the rate at which recipients seek out further information would *increase* when they are provided with more details in the notification message, and neither is it clear why this should occur for letters, but not for emails.

The only way any of these theories may be plausible is that emails are read by a different group of people than letters, and that these groups interpret the messages differently. As most organizations only give purpose-specific email addresses for technical matters, but no dedicated postal address, this may have led to a different composition of recipients in the email group compared to the letters. This could explain the observed differences in behavior between letter and email recipients, although it is impossible to conclusively prove a connection as we did not note if the collected address was technical or general-purpose, and thus cannot differentiate these classes in the evaluation. It is also unclear why this difference should only affect the PHPInfo group and not the others. Regardless of the exact mechanism at play here, the variation in the results shows that the wording of notifications needs to be carefully considered, and that the optimal wording may also depend on the message medium.

5.2 Limitations

A large fraction of our dataset consists of PHPInfo leaks, which pose a less obvious danger than the other vulnerabilities and are very easy to remediate. However, their remediation rate is only marginally higher than that of other vulnerabilities, thus the impact of this should be limited. LETTER+ATK contains a smaller percentage of PHPInfo vulnerabilities than the other groups (83.7 vs. 86.2 to 88.7 %), which may affect its overall remediation rates. However, due to the limited difference in remediation rates, the impact should be small and does not explain the large observed differences between LETTER+ATK and LETTER.

We did not attempt to hide that the messages were sent as part of a study. Thus, the study may suffer from observer effects, where recipients behave differently because they are aware that they are part of a study. Our dataset is geographically limited to German sites, which may introduce a bias if German site operators are in some way different from those in other countries, and increases the effect of sender name recognition (although the actual effect of name recognition is disputed in the literature [7, 20, 25, 30]). It also increases the availability of contact information through the imprint, as providing this information is mandatory in Germany. Finally, the study was conducted three weeks after the General Data Protection Regulation (GDPR) came into force. Thus, the timing of the study may have coincided with a generally increased interest in data protection, the effects of which we are unable to quantify.

5.3 Future Work

In our study, we have shown that letters can help increase remediation rates. This raises the question of which other channels may prove helpful. As previously discussed, Stock *et al.* performed a small ($N = 364$) evaluation of channels such as contact forms, phone numbers, letters, and social media [25], and found the increase in remediation rates to be too low in relation to the cost to make it worthwhile. A study by Maass *et al.* disagreed and reported letters to significantly increase remediation rates [20]. Further research is needed into if and when alternative contact channels can be an effective tool for notifications.

In our experiment, we made use of a legal requirement to provide contact information on websites. However, notifications could also make use of other forms of legal requirements, namely, requirements to remediate issues with relevance to data protection or cybersecurity legislation. This may serve as an incentive for remediation. Maass *et al.* found promising results with such an approach [20], and Diop *et al.* evaluated different legal environments for relevant legislation [10], which may serve as a starting point for further effort.

The differences in effectiveness between the different message contents also highlight the importance of understanding how recipients perceive notification messages, and which factors influence the trust and perceived urgency. Even though prior studies often reported low engagement with feedback mechanisms [11, 18, 25, 30], future studies should include questionnaires or other methods to collect such data, and consider collaborating with researchers from the relevant fields.

If the costs of finding addresses and sending letters is prohibitive, and as long as standardized ways of providing contact information such as security.txt [13] are not widely used, it may be possible to ask notification recipients for voluntary contributions to help to cover them. Several recipients expressed gratitude for our notifications, and one museum sent us an (unsolicited) gift of two tickets for their exhibition, indicating that at least some recipients may have been willing to financially support such notifications, if asked. Maass *et al.* reported similar reactions [20]. An investigation into the willingness to pay for unsolicited notifications (and the impact on remediation of making such a request) is a promising avenue for future work, although any such study would be well-advised to consider the legal implications, in particular the relevant competition and advertising laws in their jurisdiction. Finally, since our study was performed with German websites only, more research is needed to determine if international notifications benefit from alternative contact channels.

6 CONCLUSION

In this paper, we reported on a randomized controlled notification experiment with 1359 German website operators affected by a set of unintentional information leak vulnerabilities. We compared the effectiveness of emails to that of an alternative notification medium—letters—and the inclusion of more detailed scenarios illustrating the danger of the vulnerability. We utilized manually-collected address information, finding greatly reduced bounce rates compared to previous studies, with less than 2 % of messages being returned as undeliverable. Overall, 48.9 % of notified recipients remediated within one month, compared to 4.3 % for the control group. Letters achieved a substantial increase in remediation rate compared to emails, with differences of up to 25 percentage points. However, including a more detailed description of the risk posed by the vulnerability not only failed to improve remediation for the email groups, but actually *reduced* remediation rates for letters in some cases. This counterintuitive result highlights that more work is needed to understand how recipients perceive unsolicited notifications.

CODE AND DATA

To facilitate reproduction of our work, we release the code and (anonymized) data necessary to reproduce the figures and tables in this paper, as well as the code of the collection system. Find the data at https://zenodo.org/record/4817464.

ACKNOWLEDGMENTS

This work has been co-funded by the DFG as part of project C.1 within the RTG 2050 “Privacy and Trust for Mobile Users" and by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

REFERENCES

[1] Apache Software Foundation. 2020. *mod_info - Apache HTTP Server Version 2.4.* https://httpd.apache.org/docs/2.4/mod/mod_info.html
[2] Apache Software Foundation. 2020. *mod_status - Apache HTTP Server Version 2.4.* https://httpd.apache.org/docs/2.4/mod/mod_status.html
[3] Davide Canali, Davide Balzarotti, and Aurélien Francillon. 2013. The role of web hosting providers in detecting compromised websites. In *Proceedings of the 22nd International Conference on World Wide Web – WWW '13.*
[4] Orçun Çetin, Lisette Altena, Carlos Gañán, and Michel Van Eeten. 2018. Let Me Out! Evaluating the Effectiveness of Quarantining Compromised Users in Walled Gardens. In *Proceedings of the 14th Symposium on Usable Privacy and Security – SOUPS '18.*
[5] Orçun Çetin, Carlos Ganan, Lisette Altena, Samaneh Tajalizadehkhoob, and Michel van Eeten. 2019. Tell Me You Fixed It: Evaluating Vulnerability Notifications via Quarantine Networks. In *IEEE European Symposium on Security and Privacy 2019 – EuroS&P '19.*
[6] Orçun Çetin, Carlos Ganan, Maciej Korczynski, and Michel van Eeten. 2017. Make Notifications Great Again: Learning How to Notify in the Age of Large-Scale Vulnerability Scanning. In *Proceedings of the 16th Annual Workshop on the Economics of Information Security – WEIS '17.*
[7] Orçun Çetin, Mohammad Hanif Jhaveri, Carlos Gañán, Michel van Eeten, and Tyler Moore. 2016. Understanding the role of sender reputation in abuse reporting and cleanup. *Journal of Cybersecurity* 2, 1 (2016).
[8] Dave Crocker. 1997. *Mailbox Names for Common Services, Roles and Functions.* RFC 2142. RFC Editor. https://www.rfc-editor.org/rfc/rfc2142.txt
[9] Dan Goodin. 2017. *Failure to patch two-month-old bug led to massive Equifax breach.* https://arstechnica.com/?post_type=post&p=1166391
[10] Serigne Mouhamadane Diop, Jema David Ndibwile, Doudou Fall, Shigeru Kashihara, and Youki Kadobayashi. 2019. To Coerce or Not to Coerce? A Quantitative Investigation on Cybersecurity and Cybercrime Legislations Towards Large-Scale Vulnerability Notifications. In *International Conference on Software Reliability Engineering Workshops – ISSRE '19 Workshops.*
[11] Zakir Durumeric, Mathias Payer, Vern Paxson, James Kasten, David Adrian, J Alex Halderman, Michael Bailey, Frank Li, Nicolas Weaver, Johanna Amann, and Jethro Beekman. 2014. The Matter of Heartbleed. In *Proceedings of the 2014 Internet Measurement Conference – IMC '14.*
[12] Brad Efron. 1979. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics* 7, 1 (1979).
[13] E. Foudil and Y. Shafranovich. 2020. *A File Format to Aid in Security Vulnerability Disclosure.* Internet-Draft draft-foudil-securitytxt-09. IETF Secretariat.
[14] Internetwache.org. 2015. *Don't publicly expose .git or how we downloaded your website's sourcecode - An analysis of Alexa's 1M.* https://en.internetwache.org/dont-publicly-expose-git-or-how-we-downloaded-your-websites-sourcecode-an-analysis-of-alexas-1m-28-07-2015/
[15] Jeremy Orloff and Jonathan Bloom. 2014. *18.05 Introduction to Probability and Statistics—MIT OpenCourseWare.* https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/
[16] Marc Kührer, Thomas Hupperich, Christian Rossow, and Thorsten Holz. 2014. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In *23th USENIX Security Symposium – USENIX Security '14.*
[17] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings of the 2019 Network and Distributed System Security Symposium – NDSS '19.*
[18] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon Mccoy, Stefan Savage, and Vern Paxson. 2016. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *25th USENIX Security Symposium – USENIX Security '16.*
[19] Frank Li, Grant Ho, Eric Kuan, Yuan Niu, Lucas Ballard, Kurt Thomas, Elie Bursztein, and Vern Paxson. 2016. Remedying Web Hijacking: Notification Effectiveness and Webmaster Comprehension. In *Proceedings of the 25th International Conference on World Wide Web – WWW '16.*
[20] Max Maass, Alina Stöver, Henning Pridöhl, Sebastian Bretthauer, Dominik Herrmann, Matthias Hollick, and Indra Spiecker. 2021. Effective Notification Campaigns on the Web: A Matter of Trust, Framing, and Support. In *30th USENIX Security Symposium – USENIX Security '21.*
[21] Max Maass, Pascal Wichmann, Henning Pridöhl, and Dominik Herrmann. 2017. PrivacyScore: Improving privacy and security via crowd-sourced benchmarks of websites. In *Annual Privacy Forum.*
[22] Tobias Mueller, Matthias Marx, Henning Pridöhl, Pascal Wichmann, and Dominik Herrmann. 2018. Sicherheit und Privatheit auf deutschen Hochschulwebseiten: Eine Analyse mit PrivacyScore. *25. DFN-Konferenz “Sicherheit in vernetzten Systemen”* (2018).
[23] MySQL. 2020. *MySQL Reference Manual: mysqldump - A Database Backup Program.* https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html
[24] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D Strowes, and Narseo Vallina-Rodriguez. 2018. A Long Way to the Top: Significance, Structure and Stability of Internet Top Lists. In *Proceedings of the 2018 Internet Measurement Conference – IMC '18.*
[25] Ben Stock, Giancarlo Pellegrino, Frank Li, Michael Backes, and Christian Rossow. 2018. Didn't You Hear Me? Towards More Successful Web Vulnerability Notifications. In *Proceedings of the 2018 Network and Distributed System Security Symposium – NDSS '18.*
[26] Ben Stock, Giancarlo Pellegrino, Christian Rossow, Martin Johns, and Michael Backes. 2016. Hey, You Have a Problem: On the Feasibility of Large-Scale Web Vulnerability Notification. In *25th USENIX Security Symposium – USENIX Security '16.*
[27] The PHP Group. 2020. *PHP: phpinfo - Manual.* https://www.php.net/manual/en/function.phpinfo.php
[28] Kami Vaniea and Yasmeen Rashidi. 2016. Tales of Software Updates: The process of updating software. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.*
[29] Marie Vasek and Tyler Moore. 2012. Do Malware Reports Expedite Cleanup? An Experimental Study. In *Workshop on Cyber Security Experimentation and Test.*
[30] Eric Zeng, Frank Li, Emily Stark, and Adrienne Porter Felt. 2019. Fixing HTTPS Misconfigurations at Scale: An Experiment with Security Notifications. In *Proceedings of the 18th Annual Workshop on the Economics of Information Security – WEIS '19.*

A EXAMPLE NOTIFICATION

Figure 4 shows an anonymized example letter in the original German version. In the following, we will provide translated versions of the explanations of the different vulnerability types that were used in the letter. All messages started with the following text:

TECHNISCHE UNIVERSITÄT DARMSTADT

Max Maass | SEEMOO / TU Darmstadt | Mornewegstr. 32 | 64293 Darmstadt
[redacted]

Schwachstellen auf Ihrer Website http://[redacted].de/

Sehr geehrte Damen und Herren,

wir wenden uns an Sie, da Sie im Impressum der Webseite als Verantworliche genannt sind. Im Rahmen einer Forschungsarbeit haben wir mehrere Schwachstellen auf Ihrer Website http://[redacted].de/ gefunden, über die wir Sie gerne in Kenntnis setzen möchten. Bei den Schwachstellen handelt es sich um Dateien, die unbeabsichtigt oder fahrlässigerweise öffentlich zugänglich sind und sensible Informationen preisgeben.
Es handelt sich um folgende Adressen:

`http://[redacted].de/dump.sql`
Über diese Adresse lässt sich eine Sicherung Ihrer Datenbank herunterladen. Obwohl wir deren Inhalte nicht im Einzelnen überprüft haben, lässt sich davon ausgehen, dass darin nicht für die Öffentlichkeit bestimmte Inhalte enthalten sind.

`http://[redacted].de/info.php`
Über diese Adresse lassen sich Informationen über verwendete Software und deren Versionen, als auch interne Konfigurationsdetails Ihres Servers abrufen, die aus Sicherheitsgründen nicht öffentlich verfügbar sein sollten.

Im Sinne der Sicherheit Ihrer Webseite raten wir Ihnen, diese Schwachstellen schnellstmöglich zu beheben bzw. Selbiges zu veranlassen.
Weitergehende Informationen zum Projekt, den Schwachstellen, Hilfestellungen zur Behebung und eine Statusabfrage für Ihre Webseite finden Sie unter:

`https://web-survey.seemoo.tu-darmstadt.de/[redacted]`

Für Fragen stehe ich sehr gerne zur Verfügung.

Mit freundlichen Grüßen

[redacted]
Max Maass

Fachbereich Informatik
Sichere Mobile Netze
Secure Mobile Networking

SECURE MOBILE NETWORKING

Max Maass

SEEMOO / TU Darmstadt
Mornewegstraße 32
64293 Darmstadt

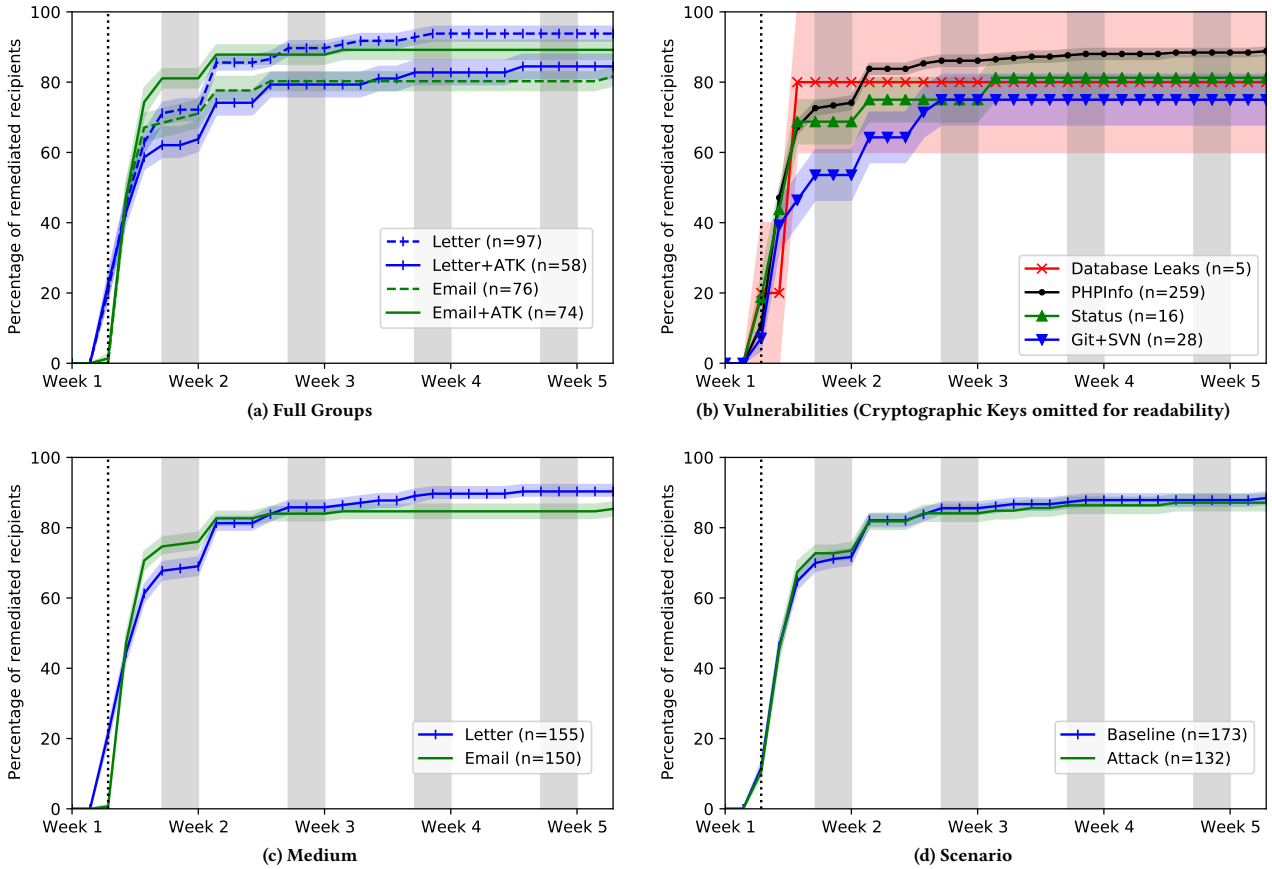
Fax +49 6151 16 - 25471
web-survey@seemoo.tu-darmstadt.de

Datum: 11.6.2018

CRISP Center for Research in Security and Privacy

Privacy and Trust for Mobile Users

Figure 4: German example notification letter (translation provided below)



(a) Full Groups

(b) Vulnerabilities (Cryptographic Keys omitted for readability)

(c) Medium

(d) Scenario

Figure 5: Median, 1st and 3rd quartiles for the remediation rates in different experimental groups, vulnerabilities, mediums, and presence of attack scenarios, for reached recipients only.

> To Whom it may concern,
>
> we are contacting you because you are listed as the responsible person for the website in its imprint. Within the context of a research project we found multiple vulnerabilities on your website http://www.example.com about which we would like to kindly advise you. The vulnerabilities stem from files that are publicly accessible, either unintentionally or by carelessness, and reveal sensible information. The following addresses are affected:

It was followed by a list of URLs and the relevant explanations, listed below. The message then closed by stating:

> In the interest of your websites security we advise you to remedy those vulnerabilities as soon as possible. Further information on our project, your vulnerabilities, assistance in remediation and a status check for your website is available at: [URL with token]
>
> I will be happy to assist you with any further questions.
>
> With kind regards,
> Researcher Name

The message signature contained the name, fax number, email and postal address of the sender. We will now provide the individual text blocks that describe the different vulnerabilities.

A.1 SSH Key

Baseline. At this address anyone can download an access key which can presumably be used to log in to your website and get full access. Please consult an expert for the next steps, because we can not determine the full impact of this problem.

Attack. This vulnerability was only observed once and thus does not contain a version with attack scenario.

A.2 TLS Key

Baseline. At this address anyone can access the private key to your websites transport encryption. Because of that we have to assume that the encryption can actually be abrogated. Please deactivate the access to the key, renew the key as well as your encryption certificate and revoke the old key.

Table 7: Requested Paths for the different vulnerabilities

Vulnerability	Paths
Keyfile	`id_rsa, .ssh/id_rsa, privatekey.key, private.key, myserver.key, key.pem, privkey.pem, [domain].key, [domain_full].key, [subdomain].key, [domain].pem, [full_domain].pem, [subdomain].pem, cert.pem, certificate.pem, domain.key`
Database	`dump.db, dump.sql, sqldump.sql, sqldump.db, db.sqlite, data.sqlite, sqlite.db, [domain].sql, [domain_full].sql, [subdomain].sql, [domain].db, [domain_full].db, [subdomain].db`
Core dump	`core`
VCS	`.git/HEAD, .svn/wc.db`
Status	`server-status/, server-info/`
PHPInfo	`phpinfo.php, test.php, info.php`

Attack. This vulnerability was only observed once and thus does not contain a version with attack scenario.

A.3 Database Backup

Baseline. At this address, anyone can download a backup of your database. Although we did not review its content in detail, it is very likely that its contents are not intended for public consumption.

Attack. An attacker can likely extract the list of users from that backup and can possibly extract the passwords. This may grant them full access to the website and allow them to manipulate the content, which can lead to defamation.

A.4 VCS

Baseline. The availability of this file indicates that the source code of your website is publicly accessible. While we did not verify those contents in detail, we assume that it contains content that is not meant to be publicly accessible, such as login and contact data or internal configuration files.

Attack. Depending on whether an attacker discovers information like login or contact data, she can in some circumstances acquire full access to your website or impersonate you with the help of the contact data to gain access to further information by fraud.

A.5 Server-Status

Baseline. At this address anyone can see which pages on the website are currently accessed from which IP address using which parameters. This means that in doing so you illicitly disclose the identity and activities of your visitors.

Attack. Using this information an attacker can trace who visits your website and learn about visit duration and links clicked. In the worst case she thereby learns the so called “Session ID”, which enables her to seize the role of a visitor and impersonate them.

A.6 Server-Info

Baseline. At this address anyone can retrieve information about software modules in use as well as their versions and internal configuration details of your server which should not be public for security reasons.

Attack. With the help of such version information an attacker can very easily determine whether outdated software with known vulnerabilities is in use. If this is the case, she can exploit those easily and in the worst case gain access to the server.

A.7 PHPInfo

Baseline. At this address anyone can access information about the software in use as well as internal configuration details, which should not be publicly accessible for security reasons.

Attack. With the help of such version information an attacker can easily determine whether outdated software with known vulnerabilities is in use. If this is the case, she can exploit them with ease and in the worst case gain access to the server.