# Let Me Out! Evaluating the Effectiveness of Quarantining Compromised Users in Walled Gardens

Cetin, Orcun; Altena, E.M.; Hernandez Ganan, Carlos; van Eeten, Michel

**Publication date**
2018

**Document Version**
Final published version

**Published in**
Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)


# Let Me Out! Evaluating the Effectiveness of Quarantining Compromised Users in Walled Gardens

**Orçun Çetin, Lisette Altena, Carlos Gañán, and Michel van Eeten,**
***Delft University of Technology***

https://www.usenix.org/conference/soups2018/presentation/cetin

**This paper is included in the Proceedings of the Fourteenth Symposium on Usable Privacy and Security.**

**August 12–14, 2018 • Baltimore, MD, USA**

ISBN 978-1-939133-10-6

**Open access to the Proceedings of the Fourteenth Symposium on Usable Privacy and Security is sponsored by USENIX.**

# Let Me Out! Evaluating the Effectiveness of Quarantining Compromised Users in Walled Gardens

Orçun Çetin, Lisette Altena, Carlos Gañán, and Michel van Eeten

Department of Multi-Actor Systems, Delft University of Technology
{F.O.Cetin,E.M.Altena,C.Hernandezganan,M.J.G.Vaneeten} @tudelft.nl

## ABSTRACT

In the fight to clean up malware-infected machines, notifications from Internet Service Providers (ISPs) to their customers play a crucial role. Since stand-alone notifications are routinely ignored, some ISPs have invested in a potentially more effective mechanism: quarantining customers in so-called walled gardens. We present the first empirical study on user behavior and remediation effectiveness of quarantining infected machines in broadband networks. We analyzed 1,736 quarantining actions involving 1,208 retail customers of a medium-sized ISP in the period of April-October 2017. The first two times they are quarantined, users can easily release themselves from the walled garden and around two-thirds of them use this option. Notwithstanding this easy way out, we find that 71% of these users have actually cleaned up the infection during their first quarantine period and, of the recidivists, 48% are cleaned after their second quarantining. Users who do not self-release either contact customer support (30%) or are released automatically after 30 days (3%). They have even higher cleanup rates. Reinfection rates are quite low and most users get quarantined only once. Users that remain infected spend less time in the walled garden during subsequent quarantining events, without a major drop in cleanup rates. This suggests there are positive learning effects, rather than mere habituation to being notified and self-releasing from the walled garden. In the communications with abuse and support staff, a fraction of quarantined users ask for additional help, request a paid technician, voice frustration about being cut off, or threaten to cancel their subscriptions. All in all, walled gardens seem to be a relatively effective and usable mechanism to improve the security of end users. We reflect on our main findings in terms of how to advance this industry best practice for botnet mitigation by ISPs.

## 1. INTRODUCTION

Fighting the scourge of malware-infected end user machines is an ongoing challenge that involves many different actors, from software vendors, incident response organizations, antivirus vendors, network operators and, last but not least, the end users themselves. Some efforts are more focused on preventing infections, others on remediation – i.e., cleaning up the compromised hosts. In the context of cleanup, the role of Internet Service Providers has become more salient over time, as it became clear that many end users struggle to detect and remediate infections. The ISPs are a critical control point providing the infected machines with access to the rest of the Internet. In the past 5-10 years, a range of best practices and code of conducts have been published by leading industry associations [22, 24], public-private initiatives [11, 17] and governmental entities [12, 16]. These documents share a common set of recommendations for ISPs around educating customers, detecting infections, notifying customers, and remediating infections.

The effectiveness of these best practices is disputed. When the U.S. National Institute of Standards and Technology (NIST) was developing its own guidance on 'Models To Advance Voluntary Corporate Notification to Consumers Regarding the Illicit Use of Computer Equipment by Botnets and Related Malware', it considered using the Australian iCode as an example [26]. The SANS Institute and other stakeholders criticized this idea, arguing the Australian code had not managed to significantly improve cleanup rates of infected users [26]. Academic research has also questioned the effectiveness of these efforts [3, 4].

There are a variety of reasons for the limited impact of botnet remediation efforts by ISPs. At the core, however, is a usability problem: notifying customers that one of their machines is infected does not translate into actual cleanup. As we know from other areas in security, notifications are routinely ignored, especially if the step towards action is complicated and disrupts ongoing activities.

The lack of effectiveness of mere notifications has led some of the more security-minded ISPs to adopt what is arguably the most costly measure: putting infected customer machines into a quarantine network, also known as a 'walled garden', which only gives access to a small set of white-listed sites. Users are required to perform cleanup to get their connection restored – i.e., to be released from the walled garden. While the use of walled gardens is identified as a security best practice [25], it is also controversial. The ITU's Anti-Botnet Toolkit cites 'technical, financial, legal and customer satisfaction-related disincentives' that may be raised by an ISP [15].

Quarantining infected users is contested, but also one of the few measures that could improve cleanup rates and help end users to remediate and secure their machines. Remarkably, there has been no publicly available study on the effectiveness of walled gardens. Do they actually help end users to clean up? How often do users get reinfected? How much time do users spend in quarantine? How much support do they need? How much pushback do ISPs face from their users?

We present the first empirical study on the usability and effective-


*USENIX Symposium on Usable Privacy and Security (SOUPS)* 2018.
August 12–14, 2018, Baltimore, MD, USA.

ness of walled gardens as a notification and remediation mechanism. We analyzed 6 months of data (April-October 2017) from a real-world implementation of a walled garden at a medium-sized ISP that we collaborated with. The ISP is a market leader in its home market that serves retail broadband to several million customers. The ISP took 1,736 quarantining actions involving 1,208 retail customers. In collaboration with the ISP, we correlated these quarantining actions with independent observations from botnet sinkhole data to track remediation success. We also analyzed anonymized communications with quarantined users. In combination, these datasets allow us to estimate cleanup rates, recidivism rates, and user engagement with the walled garden environment.

In short, we make the following contributions:

- We present the first empirical study of a real-world 'walled garden' system to notify and quarantine end users with malware-infected machines – a widely-recognized security best practice for ISPs.
- We measure the effectiveness of the walled garden notifications in terms of end user cleanup efforts and find that the majority of users spend a relatively short time in quarantine, while still successfully removing the infection.
- We provide insight into the experiences of users by analyzing their communication with ISP employees and find that a fraction of them are frustrated about their access being cut off. This is especially true for users who turn out to operate business services over their consumer broadband connection.

The rest of this paper is structured as follows. Section 2 reviews prior work. Section 3 outlines the properties of walled garden systems and Section 4 presents the data collection methodology. Next, Section 5, we shed light on the effectiveness of the real-world walled garden and relationship between cleanup success and other factors. Section 6 presents key insights gathered from communications. Section 7 presents the ethical considerations and Section 8 discusses the limitations of the study. We conclude by covering the main lessons learned for the use of walled garden systems in securing end-user machines.

# 2. RELATED WORK

As far as we are aware, there is no prior work on the effectiveness of notifying end users in an access network and asking them to clean up malware infections on their machines. Here, we briefly survey four related areas of work. The work on abuse and vulnerability notifications has studied similar mechanisms , but typically with a different type of end user, namely webmasters, server admins and network operators, not home users. This makes the effectiveness of those mechanisms difficult to compare with malware notifications and cleanup by consumers. Another area of related work concerns the design of the notifications and warnings for regular end users. These notifications and warnings are mostly meant to prevent compromise, trying to steer the user back to safety. In contrast, we study a notification mechanism where the action is not avoiding danger, but dealing with the damage that has already occurred. Also, the action required of the user in case of compromise is not a single decision for or against a potentially dangerous action, but the execution of a rather complicated set of steps to resolve the incident that has already manifested itself. Finally, there is related work that studies whether and how end users understand the security situations they face and how they behave in those contexts. In our study, we do not observe the users directly, nor elicit their thoughts about the situation, but we do have data on some of their actions, as well as some visibility into their experiences through their communications with the ISP.

## 2.1 Abuse notifications

A range of studies has focused on if and how abuse notifications can expedite cleanup of compromised websites. Notifications can be sent to the affected owners of the site or to their hosting provider. An early study by Vasek *et al.* [1] indicated that more verbose abuse notifications to hosting providers resulted in higher cleanup rates than notifications with minimal information. Çetin *et al.* [10] found that around half of all compromised sites got cleaned up after a notification to the hosting provider. The reputation of the sender of the notifications had no observable impact on the cleanup rate. Li *et al.* [21] showed that direct notifications to webmasters via Google's Webmaster Console increased the likelihood of cleanup by over 50%. They report that 6.6% of sites cleaned up within a day of detection, 27.9% within two weeks, and 41.2% within one month. In a qualitative study, Canali *et al.* [8] set up vulnerable web servers on 22 hosting services, ran different attacks on them that simulated infections and then notified the providers about these attacks. Only one hosting provider notified their customers about a potential compromise of their website after the first notification and only half of the providers after the second notification. Additionally, around 13% of the notified providers warned the user of being compromised upon receiving abuse notifications.

## 2.2 Vulnerability notifications

Various studies have looked into the feasibility and efficacy of vulnerability notification mechanisms. For example, Kührer *et al.* [20] issued notifications to administrators of vulnerable Network Time Protocol (NTP) servers, in collaboration with CERTs, clearinghouses and afflicted vendors. Though their study lacks a control group to assess the impact of the campaign itself, they found that 92% of NTP server were remediated in 13 weeks. Stock *et al.* [29] studied large-scale vulnerability notification campaigns and found that only around 6% of the affected parties could be reached. Of that small fraction, around 40% were remediated upon notification. Similarly, in a study by Çetin *et al.* [9], the authors concluded that the deliverability of email-based notifications was very poor. They proposed searching for other mechanisms. Stock *et al.* [28] later tested the effectiveness of other channels such as postal mail, social media, and phone and concluded that the slightly higher remediation rates of these channels do not justify the additional work and costs.

## 2.3 Design of notifications and warnings

A large body of literature explored user responses to different types of security notifications and warnings, focusing on why users ignore warnings and how this could be avoided. A study conducted by Krol *et al.* [19] showed that users' misunderstanding of warnings and notifications is a reason for ignoring them. Almuhimedi *et al.* [2] studied user reactions to Google Chrome malware warnings. Up to half of the warnings were ignored under certain circumstances. Some users confused the malware warnings with SSL warnings. Sunshine *et al.* [30] examined users' reactions to existing and newly designed SSL warnings and suggested that, although existing SSL warnings can be improved, minimizing the use of SSL warnings by blocking users from making insecure connections proves to be more effective. Finally, Mathur *et al.* concluded that one of the reasons why users ignore software updates is that updates regularly interrupt users who often lack sufficient basic information to decide whether or not to update [23]. A closely related topic is

the problem of habituation of users to ignore warnings after they have learned that this does not seem to cause any harm [18, 27]. Bravo-Lillo *et al.* tested the effectiveness of user-interface modifications to draw users' attention to the most important information required for decisions [6, 7].

## 2.4 End user security behavior

Multiple studies have demonstrated that end users have difficulty securing their computers, either because of lack of knowledge or ignoring security advice that is hard to understand. In a study conducted by Wash *et al.* [31] on how users perceive automated software updates, the authors observed that the majority of users do not correctly understand the automatic update settings on their computer and cannot manage software updates the way they intend to. This mismatch between intention and behavior frequently led to computers being more or less secure than intended. Fagan *et al.* [13] studied user motivations regarding their decisions on following common security advice (i.e., update software, use password manager, change passwords) and concluded that the majority of users follow the usability/security trade-off. Finally, Forget *et al.* [14] developed a Security Behavior Observatory to collect data on users' behavior and their machine configurations. Their findings highlighted the importance of content, presentation, and functionality of security notifications provided to users who have different expertise, expectations, and computer security engagement.

# 3. WALLED GARDEN

The concept of a "walled garden" stems from the early days of the web, when ISPs implemented closed networks to control the applications, content and media that their subscribers could access. Some ISPs extended the capabilities of these networks to exclude rival content from the heavily curated garden. This model has all but disappeared.

These days, walled gardens are a method to notify subscribers about malware infections and restrict their access to the Internet while infected, so as to protect the infected user from further harm as well as preventing the user's machine from harming other users or networks. More precisely, a walled garden is a quarantined environment that restricts the information flow and services of an end user inside a network. Besides keeping the infected users safely in quarantine, the walled garden also plays an important role in informing the user. While the user tries to browse the Web, she or he will be redirected to a landing website with information about the type of infection and how to clean it up. Whereas emails or letters with the same content can be ignored relatively easily, this mechanism cannot.

There are different ways of implementing and deploying walled gardens to fight malware infections. RFC6561 [22] describes 2 different types: *strict*, a walled garden environment that restricts almost all services, except those to a whitelist of malware mitigation services; and *leaky*, an implementation that permits access to all Internet resources, except those that are deemed malicious, and ensures access to those that can be used to notify users of infections. In this paper, we focus on a strict implementation, which is what was installed at our partner ISP. A strict implementation is potentially more effective, but also more contested.

The quarantine period of an infected user mainly depends on three different processes: (i) the malware detection process; (ii) the infection notification and quarantining process; and (iii) the release process. The flow chart in Figure 1 shows the overall quarantine process in place at our partner ISP. It starts with the ISP realizing that a subscriber is infected and ends with the subscriber leaving

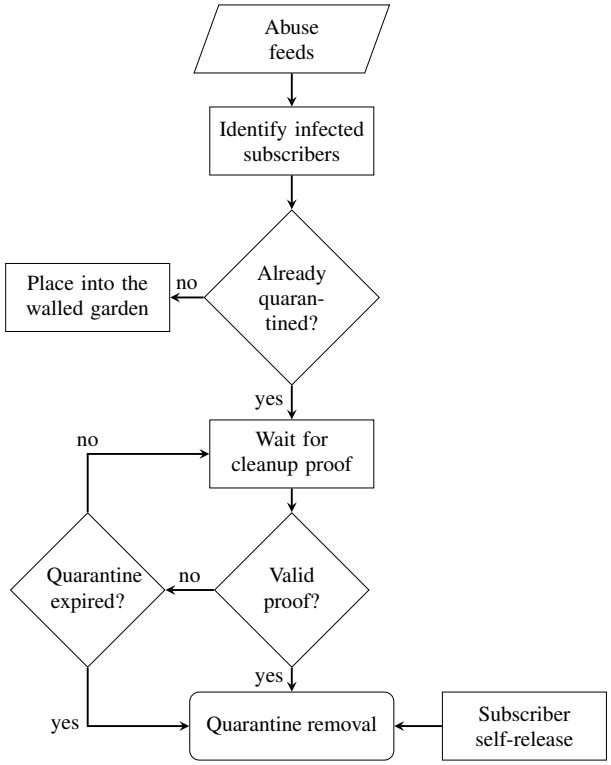



**Figure 1: Quarantine flow chart**

the walled garden. The starting point, i.e., the infection detection, is independent of the walled garden environment. Typically, this detection is not based on their own network monitoring, but on third-party notifications, e.g., from botnet sinkhole operators and security intelligence providers. The processing of abuse feeds varies per ISP, ranging from manually checking incoming notifications to highly automated systems that consume the feed and push the relevant incidents into abuse ticketing systems. When certain abuse data fits a predefined policy, on data trustworthiness, timeliness, the affected customer type and other criteria, the ISP places the connection of that particular customer into the walled garden.

In order to leave the walled garden, the customer is requested to provide proof of the cleanup actions that were taken to mitigate the infection. This proof might consist of the log of an anti-virus scan or some description of the steps taken by the user. To facilitate the cleanup, the walled garden can provide access to a range of white-listed services. Typically these services include free antivirus tools and trusted software suppliers. Other white-list entries may be added to protect critical services for the user, such as webmail services and online banking. Thus customers can perform basic remediation steps and communicate with the abuse desk, even though they are quarantined.

After leaving the walled garden, there is no guarantee that the malware infection was actually remediated. There are several reason by which a user could get out of the quarantine network while being still infected. First of all, certain walled garden implementations allow users to self-release at any time. Normally, this option is only available for the first and perhaps second infection event during a specific period of time. When a user is placed in quarantine for a third time, because of a reinfection or because the earlier infection was not actually removed, the option of self-release

is no longer available. The quarantine removal can now only be executed by the ISP's abuse or support staff. Second, a user can provide erroneous cleanup proofs. For instance, with an increasing number of connected devices in subscriber networks, it is possible for a non-savvy user to perform cleanup actions on a non-infected device and provide the wrong cleanup proofs to the ISP. It is also possible that advanced malware could remain undetected by common antivirus or removal tools. This will allow infected users to leave temporarily the walled garden until the same infection is detected again. Third, some walled garden implementations have an expiration period after which any user in quarantine is released. Fourth, and last, ISP staff might decide to release the user without cleanup. Infected users might request to leave the walled garden for other reasons, like an urgent need for certain online services or because the malware infection cannot be remediated while being in the walled garden. The ISP might allow the user to access the Internet to gather a non-whitelisted cleanup tool.

Our study has been conducted on a walled garden environment deployed for the home users of a medium-sized ISP. Their enterprise and mobile customers are not quarantined. The walled garden follows a strict implementation that redirects users to a landing page (see Appendix A) and limits the access to a set of 41 white-listed websites, including cleanup tools, antivirus solutions, Microsoft updates, webmail providers and online banking. Their implementation of the walled garden provides users with two chances to self-release within a period of 30 days. With the third quarantine action, the option to self-release is revoked and the intervention of the ISP's abuse staff is required. After a period of 30 consecutive days in quarantine, the walled garden automatically releases those quarantined customers who did not self-release or contact abuse staff.

# 4. DATA COLLECTION

In this section we describe the data that was provided by an ISP to analyze the effectiveness of a particular implementation of a strict walled garden. Our study consists of 1,736 quarantine events associated with 1,208 unique subscribers of a medium-sized European ISP's network during a 6 months period. The data was gathered from four different sources that support the ISP's abuse management process: (*i*) abuse feeds providing security incident data to ISPs; (*ii*) walled garden logs recording details of quarantine events in the ISP's network; (*iii*) help desk logs containing the ISP's help desk communication with customers; and (*iv*) abuse desk communication logs providing email exchange between abuse desk employees and customers.

## 4.1 Abuse feeds

In order to detect botnet-related infections, the ISP under study leverages abuse feeds provided by the Shadowserver Foundation. For our analysis, we gathered the Shadowserver botnet reports, collected over a time frame of 9 months between April 10th, 2017 and December 30th, 2017. Three different types of reports are analyzed:

- *Drone Reports*: Drone reports contain detailed information on infected machines discovered through monitoring sinkhole traffic, malicious scans and spam relays. We observed a total of 1,620 number of malware infected customers in the network managed by the ISP under review.
- *Sinkhole Reports:* Sinkhole reports contain information about sinkhole servers that did not use the conventional bot signatures such as HTTP referrers. Due to lack of conventional bot signatures, many IP addresses mentioned in this reports do not have a specific infection name. During our study period, we observed 1,598 unique infected users who had a subscription with the ISP under review.
- *Shadowserver's Microsoft Sinkhole:* Microsoft shares via Shadowserver the intelligence gathered from some of their sinkhole servers. Throughout our data collection period, a small number of malicious IP address related to our ISP were captured by Microsoft sinkholes. We only found 8 IP addresses during our study period.

| | Sinkhole | MS sinkhole | Drone |
|---|---|---|---|
| # infected users | 1,598 | 8 | 1,620 |
| % quarantined | 22% | 63% | 59% |

**Table 1: Infections per feed and quarantined users**

As shown in Table 1, we observe a total of 1,620 unique infected users in the Drone feed, 1,598 unique infected users in Sinkhole and 8 unique infected users in MS sinkhole feeds. Not all of these infections trigger a quarantine action, as Table 1 illustrates. There are several reasons why infected users are not quarantined: (i) the user is a mobile or enterprise customer; (ii) the abuse staff decides that quarantining would make matters worse (as in the case of ransomware, where users are by definition already aware of the infection and the lack of Internet access means they might have no viable way to recover their files); (iii) the walled garden environment was undergoing maintenance; and (iv) there are no quarantining actions during the weekend. Figure 2 shows the daily number of unique IP addresses seen in the feeds.

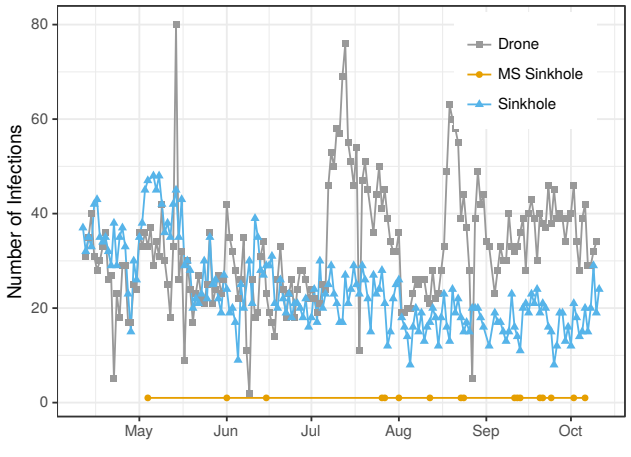



**Figure 2: Daily unique infected customers per abuse feed**

## 4.2 Walled garden logs

During our study period, 1,208 retail customers were placed into the walled garden based on the abuse feeds provided by Shadowserver. As some customers were quarantined more than once, this corresponds to 1,736 quarantining events. For each one of these events, several factors were recorded: (i) quarantine time-stamp; (ii) quarantine release mechanism; (iii) quarantine removal time-stamp; (iv) infection type; (v) quarantine event number; and (v) self-release option.

Beside the logs created by the walled garden itself, the quarantined users also have the possibility to submit a form through the walled garden landing page (see Appendix B). This form allows users to explain what cleanup actions they have taken, as well as any other feedback they might have. During the study period, 1,575 forms were received from 831 different infected customers (see Table 2).

| | Walled garden form | Abuse desk emails | Help desk phone calls |
|---|---|---|---|
| # Users | 831 | 600 | 468 |
| # Messages | 1,575 | 2,027 | 966 |

**Table 2: Messages and users per communication channel**

## 4.3 Help and abuse desks logs

In addition to the walled garden forms (i), customers can also contact the ISP in other ways. We also collected data on (ii) emails between infected customers and the abuse desk; and (iii) phone calls, store visits and social media chat calls between the help desk and the infected customers. Quarantined customers contacted the abuse desk twice as often as the help desk. Table 2 shows that the abuse desk received 2,027 emails, from 600 unique users while help desk employees reported 966 conversations associated with 468 quarantined users.

# 5. WALLED GARDEN EFFECTIVENESS

We evaluate the impact of the walled garden notification on remediation by looking at the percentage of users that managed to clean the infected machine and at the time an end user remains in the walled garden. We also analyze the relationship between cleanup success and other factors, most notably the type of malware infection, the release mechanism used to get out of the quarantine, and the time spent in the walled garden.

To evaluate cleanup, we distinguish three outcomes when users are released from the walled garden: (i) the user successfully performed cleanup and then stays clean for the rest of the study period; (ii) the user successfully performed cleanup, but the machine is reinfected at a later time in the study period, at least 30 days after the quarantine event; and (iii) the user did not successfully clean up the machine, as evidenced by seeing the offending IP address reported again for the same infection within 30 days of leaving the walled garden.

There is no clear basis for drawing the boundary between a persistent infections and a clean and reinfected machine. Even persistently-infected machines are not seen in the Shadowserver feed every day or even every few days. This depends on a variety of factors, like the malware type and whether the user even turns on the machine. He or she might be on vacation, for example. We decided to count conservatively in terms of cleanup success and use a long period (30 days) before considering the machine clean. Figure 3 shows how these metrics are calculated based on the abuse feeds and the walled garden logs.

There is no clear evidence on where to establish the cut-off point to distinguish persistently infected from clean and reinfected. Figure 4 shows the time between consecutive quarantine events. The median time between quarantine events is 4 days. Roughly 70% of the customers who are seen again after being released from quarantine, are seen within 10 days. As gaps in observations are normal for infected machines, this short interval suggests that these machine were probably not cleaned up. After 20 days, the distribution becomes more or less flat with a slow decay. Choosing a cut-off

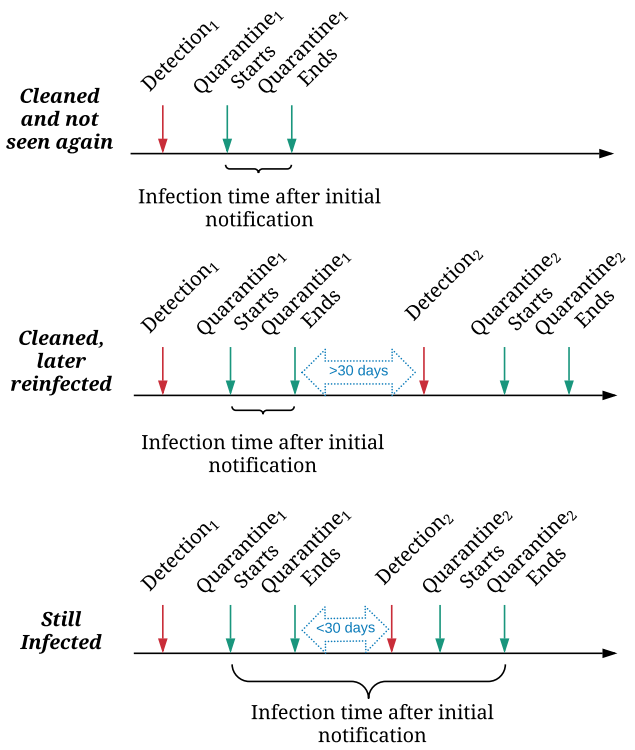



**Figure 3: Definition of quarantine outcomes**

beyond this point only a modest impact on the results. Reinfection rates would change from 16% (day 20 cut-off) to 13% (day 30) to 7% (day 40). As can be seen in the cumulative distribution, around 13% of the users had a gap between quarantine events of 30 days or more – in other words, these are the users we count as cleaned, but later reinfected.

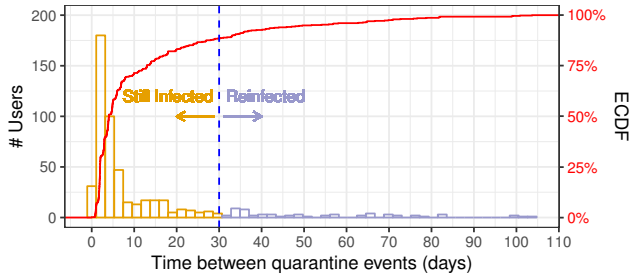



**Figure 4: Time between consecutive quarantine events**

## 5.1 Overall remediation rates

In order to understand the effectiveness of the walled garden notifications, we first observe the cleanup and infection rates of the quarantined users after the notifications. We find that 69% of the end users cleaned the infection during their first quarantine event, as shown in Table 3. Another 4% of the clean end users got reinfected with the same malware strain at a later point, more than 30 days after the quarantine event. This suggests they did not correctly address the root cause of the infection. The remaining 27% of users were not able to clean the infection.

Most, but not all, users who remained infected or suffered a rein-

| Status | Number of times in quarantine | | | | | | |
|---|---|---|---|---|---|---|---|
| | #1 | #2 | #3 | #4 | #5 | #6 | #7 |
| Clean and not seen again | 830 (69 %) | 148 (49 %) | 73 (52 %) | 18 (35 %) | 17 (65 %) | 3 (50 %) | 2 (67 %) |
| Clean and later reinfected | 51 (4 %) | 13 (4 %) | 5 (4 %) | 2 (4 %) | 1 (4 %) | 0 | 0 |
| Still infected | 327 (27 %) | 142 (47 %) | 61 (44 %) | 31 (61 %) | 8 (31 %) | 3 (50 %) | 1 (33 %) |

**Table 3: Cleanup success over number of times in quarantine**

fection, end up in a second quarantine event. Around 20% of them were not quarantined again for a variety of reasons, such as being allowed to leave the quarantine environment to download antivirus solutions. While this makes the infection show up again in the Shadowserver reports, the abuse desk employees withhold the second quarantining action to see if the user is able to resolve it or not.

Of those users who ended up in quarantine for the second time, 49% of them now successfully cleaned up the infection. Again, another 4% also cleaned up, but got reinfected later. Around 47% remained infected. We observed that 139 infected end users ended up in quarantine a third time. This time 56% of them managed to remove the infection, including those who got reinfected later on.

In the tail is a group of users, around 4% of all users who ended up in the walled garden during our study period, who suffered four or more quarantine events. At the extreme end, we found three end users who were put into the walled garden seven times over the course of six months.

Next, we explored the infection time after the initial notification for all quarantined end users. Figure 5 shows the Kaplan-Meier survival curve of the users' infection and the number of remaining infected users every other day. We find that more than 40% of the infected end users cleaned the infection within a day after initial walled garden notification, 70% within 5 days and only 22% remained after a week. After a month time, only 7% of the users remained infected.

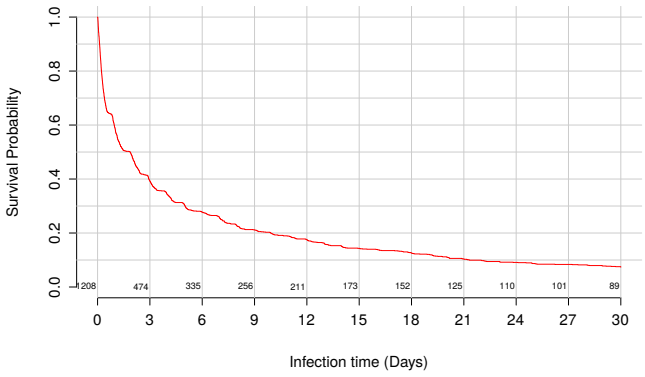



**Figure 5: Survival curve of the users' infections**

## 5.2 Malware type

We saw that most of the users in quarantine manage to clean up the infection. Does the complexity of an infection influences their success rate and time it takes them to perform the cleanup? Some malware infections might be harder to resolve than others and the white-listed cleanup tools might not always succeed. To understand the influence of the infection type on the cleanup rates, we use the infection names mentioned in the quarantine event logs. The events were triggered by 38 unique infection types. Table 4 shows the number of users and quarantined events for the top 10 most frequent infection types, which cover 89% of all the users in our dataset.

| Infection | # Users | # Quarantine events |
|---|---|---|
| Ramnit | 444 | 675 |
| Mirai | 275 | 410 |
| Nymaim | 145 | 159 |
| Downadup | 44 | 65 |
| ZeroAccess | 38 | 51 |
| Rovnix | 34 | 53 |
| Sality-p2p | 34 | 63 |
| Gozi | 21 | 30 |
| Fobber | 20 | 31 |
| Zeus | 20 | 22 |

**Table 4: Number of users and quarantine events per malware**

Figure 6 plots the survival curves for these infection types during a 30 days period. We can see significant differences in terms of infection duration for the different infection types (Gehan-Wilcoxon test, $\chi^2 = 58.6$ with $p - value = 2.5e-09$). For instance, end users infected with "Gozi" managed to cleanup all their infections during a 30 days period. On the contrary, cleanup of the more recent "Fobber" and "Rovnix" malware families was slower than the others. One possible explanation is that the more recent malware is more resistant to the standard cleanup tools linked to in the ISP notification [5].

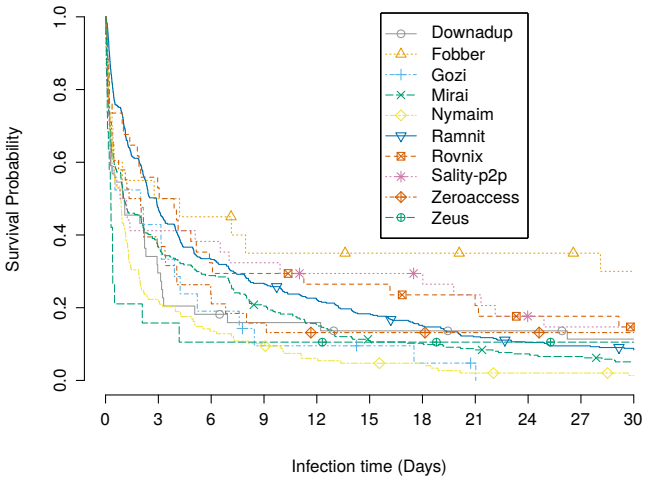



**Figure 6: Survival probabilities top 10 infection types during 30 days period**

## 5.3 Release mechanisms

As we mentioned in Section 3, the walled garden contains three mechanisms to release users from the quarantine environment: self-release, assisted release performed by the abuse staff, and quarantine expiry release. Self-release can be used only twice in one

| Status | 1st Quarantine Event | | | | 2nd Quarantine Event | | | | 3rd Quarantine Event | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Total # users | Cleaned, not seen again | Cleaned, later reinfected | Still Infected | Total # users | Cleaned, not seen again | Cleaned, later reinfected | Still Infected | Total # users | Cleaned, not seen again | Cleaned, later reinfected | Still Infected |
| Self release | 805 (67 %) | 539 (67 %) | 36 (4 %) | 230 (29 %) | 195 (64 %) | 84 (43 %) | 9 (5 %) | 102 (52 %) | 17 (12 %) | 5 (29 %) | 2 (12 %) | 10 (59 %) |
| Assisted | 361 (30 %) | 259 (72 %) | 11 (3 %) | 91 (25 %) | 102 (34 %) | 61 (60 %) | 3 (3 %) | 38 (37 %) | 114 (82 %) | 62 (54 %) | 2 (2 %) | 50 (44 %) |
| Expired | 42 (3 %) | 32 (76 %) | 4 (10 %) | 6 (2 %) | 6 (1 %) | 3 (50 %) | 1 (17 %) | 2 (33 %) | 8 (6 %) | 6 (75 %) | 1 (13 %) | 1 (13 %) |
| Total | 1208 (100 %) | 830 (69 %) | 51 (4 %) | 327 (27 %) | 303 (25 %) | 148 (49 %) | 13 (4 %) | 142 (47 %) | 139 (12 %) | 73 (53 %) | 5 (4 %) | 61 (44 %) |

**Table 5: Quarantine outcomes per release mechanism**

month. If this option is disabled, end users can contact help desk employees or abuse desk employees to get out of the quarantine or to ask for more help. However, before releasing the connection back to normal, employees might require evidence of the cleanup action, such as log files of the antivirus software that was used to remove the infection that triggered the notification.

Is there a relationship between the release mechanism and cleanup success? Since self-release is the fastest and easiest option, one might expect poorer cleanup rates. In the worst case, users simply release themselves without doing anything. To analyze the influence of the release mechanism, we compared the cleanup rates across the first three quarantine actions for all users. As shown in Table 5, the first quarantine action ended with 805 users self-releasing, 361 users following assisted release by abuse staff and 42 users were released when the quarantine period expired after 30 days. Of the 805 self-releasing end users, 67% managed to clean the infection. Another 4% also got cleaned, but was later reinfected. In other words, around 71% of all users managed to perform cleanup. Compare this to the cleanup rate of the users who were released by abuse staff after providing evidence of successful cleanup: 75%. These cleanup rates are very close together. Remarkably, self-release does not invite lax security behavior.

Another surprising finding relates to the 3% of users who remained in quarantine until it expired. They had an even higher success rate: around 86%. We do not have an explanation for this. Perhaps these users were fine with only using the white-listed webmail services and, while remaining in quarantine, automated cleanup tools – e.g., Microsoft's Malicious Software Removal Tool, which is downloaded as part of Windows updates – kicked in at some point.

Users who experienced a second quarantine event chose the self-release option in almost the same proportion (64% versus 67% in the first quarantine event). That being said, cleanup rates are not as high as during the first quarantine. In the self-release group, 48% cleaned up successfully (though 5% later got reinfected). In the provider-assisted release, the cleanup rate is 63%.

During the third walled garden notification period, 82% of the remaining end users ask ISP employees to get them out of the quarantine environment. At this stage, most users no longer get the self-release option, because they were quarantined twice already in one month. Of the users going through assisted release, 54% managed to clean up.

The drop in cleanup rates over successive quarantine events is not large, but might still suggest that perhaps users become habituated and try to get out faster, potentially spending less effort on cleaning and more on getting released. An alternative, and arguable more likely, explanation is that this is caused by selection bias. The users who end up in a second and third quarantine event are likely to be more at risk and perhaps less technically competent. This fits with the fact that with successive quarantine events, the cleanup effectiveness of the assisted-release users become slightly higher compared to the self-release group.

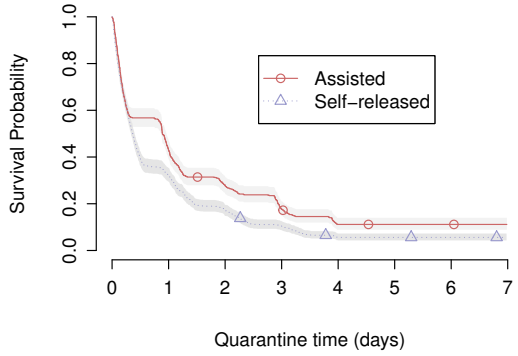



**Figure 7: Survival probabilities per release mechanism**

Figure 7 shows the duration of all infections per release mechanism in the form of Kaplan-Meier survival curves. As expected, users that needed assistance to cleanup their infections left the walled garden at a slower rate than the users that self-released. Looking at the speed at which they got removed from the quarantine, we can observe significant differences between these two groups (Gehan-Wilcoxon test, $\chi^2 = 23.1$ with $p - value = 1.5e-06$). For instance, within the first 2 days in quarantine, 84% of the users that self-released left the walled garden while only 71% of users that needed assistance did so.

## 5.4 Time spent in the walled garden

We now take a closer look at the time users spend in quarantine. Figure 8 displays the distribution of the duration of the quarantine events. The majority of quarantine events lasted less than one day and only 25% of them lasted more than 3 days. A small fraction (57 events) last until they automatically expire.

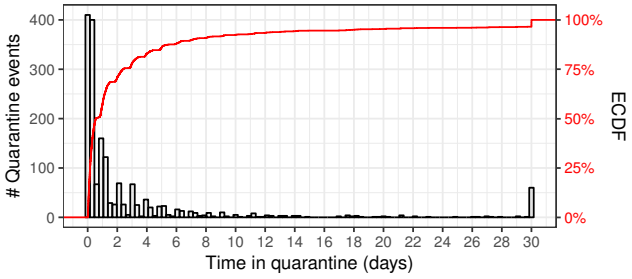



**Figure 8: Histogram and cumulative density function of the quarantine period**

Figure 9 displays the survival probability curves of users in terms of time spent in the quarantine environment for the first three quarantine events and the rest. As demonstrated in Figure 9, end users spent more time in quarantine during their first time than the second time. This might be due to being unfamiliar with the environment or with the process to clean up the infection.

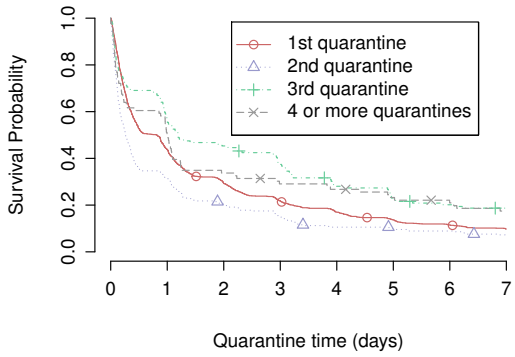



**Figure 9: Survival probabilities over different quarantine events**

To further investigate, Table 6 shows the median time spent in quarantine during the first three quarantine events. We compare them across the different release mechanisms and cleanup outcomes. End users that managed to remove the infection, stayed longer in the walled garden than those who remained infected, regardless of which release mechanism was used. Take a look at the median time of the assisted end users in the first quarantine event, for example. Those who managed to clean up spent 24 hours in quarantine, while users who remained infected took just around 7 hours. In the self-release group, successful cleanup also took longer, though for the first quarantine event, the difference is surprisingly small with the group that remains infected or got re-infected at a later stage (roughly 11 versus 10 hours).

During the second and third quarantine event, the differences become more pronounced. Longer time spent in quarantine is now clearly related to cleanup success. Users who remain infected spend about half as long in quarantine as the other two groups. It seems a certain group of users is becoming habituated to the walled garden notification and environment. They self-release very quickly and it seems unlikely that they made a serious attempt to perform cleanup.

It is important to note, though, that the self-releasing users that do succeed in cleaning up also leave the walled garden faster over successive quarantine events. The median time drops from 11 hours during the first quarantine to 8 hours (second quarantine) and then to just 3 hours (third quarantine). In other words, it seems there is not just habituation going on, but also actual positive learning effects in terms of how to perform cleanup and navigate the release from the walled garden.

# 6. END USER REACTIONS

To get a better sense of the actual experience of the end users, we qualitatively analyzed the communication of the quarantined users with the abuse and support staff at the ISP. Each communication channel was used for different of reasons. Generally, emails were sent to inform abuse desk employees about the cleanup efforts and possible causes of the infection. Interaction with the support staff, on the other hand, were more often asking for more information about the quarantine and how to resolve the situation. The content of the submitted walled garden forms often contained more specific information on the cleanup actions taken by the quarantined users. For instance, some users pasted the output of the antivirus scans in these forms to prove that the infection was no longer present.

First, we manually analyzed a sample of 200 walled garden forms, 200 help desk logs and 50 emails to the abuse desk. We saw five recurring themes that speak to the user experiences of the walled garden: (i) asking for additional help to resolve the infection and leave the walled garden; (ii) requesting a paid technician to visit the user; (iii) expressing distrust of the walled garden notification; (iv) complaining about the disruption of service; and (v) threatening to terminate the contract with the ISP. To get a sense of how many users were associated with these types of communications, we collected keywords from the manual analysis of the sample and then searched the full communication data for their presence. Table 7 shows the number of unique users associated with each topic. For 51% of the users who communicated with the ISP, their messages did not fit any of these topics and we categorized them as 'Miscellaneous'.

## 6.1 Requesting additional help

Almost 27% of the users at some point contacted the ISP to ask for additional help to cleanup the infection. The users wanted to solve the problem, but they were unable to understand the notification or to follow the steps towards quarantine release. The type of help that is requested varies widely. Some of this is driven by differences in the type of infection and the operating system of the user. Cleanup software and materials provided in the notification content would not work on all OS types, OS versions and patch levels. Some customers in our study downloaded the requested software to remove the infection, only to find out that it would not install correctly. Some users could not download the software at all from the links provided by the ISP. In those cases, they requested to be released from the quarantine environment so that they could download additional software.

One of the malware families was Mirai – the infamous botnet made up of Internet-of-Things devices. Not surprisingly, users with these infections asked for help in identifying which of their many devices was the problem and how to then secure it from future infections. Not to put too fine a point on it, but from a usability perspective the cleanup of compromised IoT is a world of pain for which we have very little practical guidance. In these cases, ISP staff would ask users additional questions about what devices they had connected to their home network. Based on the replies, staff would try to identify the offending device and more specific cleanup actions. In one case, after contacting the ISP, a user disconnected his IP camera from the network so as to prevent future infections and quarantine events, while the actual problem later turned out to be a DVR. The user ended up getting infected and quarantined again.

## 6.2 Requesting a paid technician

About 7% of the users in our study were not capable of removing the infection by themselves and requested the ISP to send a paid technician to their home. In a handful of cases, end users mentioned taking their computer to technicians at local computer repair shops. The ISP's technicians are typically people who also have a background in abuse handling. Some of the communications we analyzed were from these technicians themselves who contacted their colleagues at the ISP abuse department from the customer premises and provided detailed information about their cleanup actions. This way, the abuse desk employees got the required proof of cleanup and could release the connection from the walled garden. Interestingly, in a few rare cases, we found that the paid technician could not actually find the infection. They then referred the end users back to abuse desk employees to communicate the occurrence of a false positive. Unfortunately, as a result of this process, users remained in the walled garden environment longer.

| Release mechanism | 1st Quarantine event | | | | 2nd Quarantine event | | | | 3rd Quarantine event | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Total # users | Median quarantined time (hours) | | | Total # users | Median quarantined time (hours) | | | Total # users | Median quarantined time (hours) | | |
| | | Cleaned, not seen again | Cleaned, later reinfected | Still Infected | | Cleaned, not seen again | Cleaned, later reinfected | Still Infected | | Cleaned, not seen again | Cleaned, later reinfected | Still Infected |
| Self release | 805 | 11.16 | 10.69 | 10.24 | 195 | 8.07 | 7.04 | 3.74 | 17 | 3.15 | 11.42 | 3.29 |
| Assisted | 361 | 24.25 | 4.90 | 6.82 | 102 | 24.87 | 25.28 | 3.30 | 114 | 69.72 | 49.60 | 22.51 |

**Table 6: Summary statistics on the time to cleanup for self released and ISP assisted released mechanisms**

## 6.3 Distrust of the notification

Around 2% of the users contacted ISP employees to confirm the veracity of the email and walled garden notifications. They did not expect that their ISP would notify them about an infection and were worried that this could be a phishing attack to install ransomware or steal personal information. Users mainly contacted help desk employees to confirm the veracity of the notifications. One user replied directly to the notification email, i.e., using the very channel that he did not trust, and voiced his concerns this way to the recipient at the abuse department.

## 6.4 Complaints over disruption of service

Placing a customer in a walled garden environment is a strong incentive for end users to clean up, but also an intrusive measure. During in our study period, around 10% of the users complained in some shape or form. Some reported that their business was disrupted due to having no Internet to work with. Usually, these turned out to be users that run small businesses over their consumer broadband connection: shops, restaurants and even a small medical clinic. They claimed that they could not provide services to their customers and, as a consequence, lost customers. Some mentioned, for example, that the payment terminals did not work and so their customers could not complete their purchases. In two cases, the owner of the shop stated he had to close the shop until the problem was fixed. Several of these users provided a calculation of the monetary loss they suffered and demanded a reimbursement from the ISP.

## 6.5 Threats to terminate the contract

Around 3% of the users were so unhappy about their connection getting quarantined that they threatened to terminate their subscription and move to one of the ISP's competitors. Some of the users pointed to the losses they had incurred, others to the fact that they had to pay for the subscription even though they no longer were provided with Internet access. Also several users threaten to leave the ISP because the user could not, even with their best effort, identify and remove the infection. These users were quarantined multiple times and they spent quite a bit of time in the walled garden environment.

| Topics | # of users |
|---|---|
| Request additional help | 323 (27 %) |
| Request paid technician | 80 (7 %) |
| Distrust of the notification | 19 (2 %) |
| Complain over disruption of service | 126 (10 %) |
| Threaten to terminate the contract | 39 (3 %) |
| Miscellaneous | 621 (51 %) |

**Table 7: User issues raised in communication with ISP**

# 7. ETHICAL CONSIDERATIONS

Access to data about the user's experience upon abuse notifications is extremely limited and cooperation with an ISP is essential to enable otherwise impossible research. For this study we leverage secondary data that was originally collected by an ISP for business purposes. This data was pre-processed by a coauthor of this paper while working for this ISP and with the consent of the ISP's abuse desk manager. Moreover, the data was processed on the ISP premise and within the ISP privacy policies.

Unavoidably, the processed dataset was not fully anonymized as the high dimensionality of the data did not allow for a robust anonymization, i.e., the anonymization would have led to an unacceptable level of data loss. To ensure confidentiality, the raw dataset was stored in a secure server to which only authorized users could access. Moreover, the data was analyzed while preserving the privacy of the ISP's customers and ensuring that it is not possible to identify them from any of our results. Both the processed and anonymized data were removed after the publication. The original data remains in the ISP systems, allowing for replication if needed.

# 8. LIMITATIONS

We underline four limitations relevant to the findings of our study. First, we based our study on a single ISP with a relatively strict implementation of the walled garden notification system. The generalizability of our results to other implementations and ISPs is a matter for further studies. Second, our study uses data collected as part of the operational process of the ISP. As such, the study lacks an experimental design and a control group. This means we cannot compare the effectiveness of the walled garden notification to the cleanup rate of a mere email or no notification whatsoever. Third, our dataset on infections is limited to what has been reported in the Shadowserver feeds. As a result of this, we lack visibility into notifications triggered by other feeds and infections that are not reported by Shadowserver. This makes our coverage of malware infections biased towards those that are sinkholed and reported by Shadowserver. Malware that has escaped these defender efforts might also be harder to clean. Fourth, the cleanup outcomes are also based on the Shadowserver feeds. It is possible that an infection might not show up in the Shadowserver feeds right away. This is partly driven by user behavior, such as temporarily turning off the infected device or disconnecting it from the Internet, and partly by other factors, such as the properties of the malware families. Some are less aggressive in terms of scanning for victims or contacting the command-and-control server for commands. This absence in the feed may cause us to overestimate the cleanup rate. For this reason we chose a conservative time frame. We only counted a machine as cleaned up if we did not see it for 30 days after release from the walled garden.

# 9. CONCLUSION

In this study, we explored the effectiveness of walled garden notifications and quarantining in terms of helping users in residential networks to perform malware cleanup. Based on data on 1,736 quarantining actions involving 1,208 unique users, collected from April 2017-October 2017 by a medium-sized European ISP, we found that roughly half to three quarters of the quarantined users had managed to clean their machine. There is no clear point of reference for this success rate. When we look at prior work on abuse and vulnerability notifications, it seems to be quite high. Most of

those studies find rates well below 50%. That being said, comparison is difficult as the typical recipient of those notifications is a server admin or webmaster, not a home user.

Most users are quarantined only once, so the effort of cleanup kept them clean for months, if not longer. Perhaps the quarantine experience made users adapt their online behavior or improve their system's security defaults, like automatic patching and the installation of antivirus tools. This suggests there may also be long-term benefits to quarantining, beyond mitigating the immediate threat posed by the infection.

Users could self-release easily and quickly for the first two quarantine events in a month. Remarkably, this easy way out does not incite lax security behavior. Cleanup rates are either as high, or just a bit lower, than users who have to submit proof of cleanup to the provider and wait for the abuse staff to release them. We see a bit of evidence for habituation among a small group of users who learn how to release themselves from quarantine, rather than clean the infection. We also saw evidence, however, of a positive learning effect: successful cleanup also became faster for users going through successive quarantining events.

All in all, we found substantial support for the effectiveness of this best practice for ISPs in the fight against botnets. Since effectiveness of the other recommended best practices has been questioned, this suggests more ISPs should be considering to adopt a walled-garden solution. In light of the rising problem with IoT malware, this might become a critical line of defense. That being said, IoT malware remediation methods will differ from traditional cleanup strategies and, thus, walled garden implementations will have to be revisited to accommodate the cleanup requirements for IoT malware.

On the downside: setting up and maintaining a walled garden environment is a significant investment for an ISP. Furthermore, providing support to users in their attempts to clean up also imposes a significant cost. Around one out of four quarantined users posed a question for help to a staff member. These costs could perhaps be reduced by allowing self-release more broadly, since it seems to be more or less equally effective as the more labor-intensive form of provider-assisted release. Some of this assistance might provide a business opportunity, as we found that around 7% of the quarantined users asked for a paid technician.

A fraction of the users, around 10% of them, voiced complaints over the disruption. Around 3% even threatened to terminate the contract. We do not know how many users actually terminated their subscription, but the threat alone might, unfortunately, be enough to scare off some ISPs from investing in a walled garden. In competitive broadband markets with high penetration rates, customer acquisition is very expensive. In these situations, a prisoner dilemma might appear as not having a walled garden might be a competitive advantage. This could push ISPs to not deploy it, even though it is effective. On the other hand, if all ISPs adopted it simultaneously, it would generate collective benefits, though these would not necessarily flow back to the ISP, except through lower customer churn rates.

We did notice that the group which seemed the most negative about the quarantining actions were small businesses operating on a consumer broadband connection. ISPs could prevent them from being affected in the future by providing an easy transition to a comparatively-priced business subscription, which would take them out of the consumer market – and thus keep them away from the walled garden. This would reduce the pushback over time and allow the walled garden to do what it does best: protecting home users from further damage caused by their infection, and protecting the rest of the Internet from the infected home user.

## 10. ACKNOWLEDGMENT

This publication was supported by a grant from the Netherlands Organisation for Scientific Research (NWO), under project number 628.001.022. Also, we would like to thank the anonymous reviewers, Folkert Visser and Dennis van Beusekom for their helpful comments.

## 11. REFERENCES

[1] Do malware reports expedite cleanup? an experimental study. In *Presented as part of the 5th Workshop on Cyber Security Experimentation and Test*, Bellevue, WA, 2012. USENIX.

[2] H. Almuhimedi, A. P. Felt, R. W. Reeder, and S. Consolvo. Your reputation precedes you: History, reputation, and the chrome malware warning. In *Symposium on Usable Privacy and Security (SOUPS)*, volume 4, page 2, 2014.

[3] H. Asghari, M. Ciere, and M. J. Van Eeten. Post-mortem of a zombie: conficker cleanup after six years. In *USENIX Security Symposium*, pages 1–16. USENIX Association, 2015.

[4] H. Asghari, M. J. van Eeten, and J. M. Bauer. Economics of fighting botnets: Lessons from a decade of mitigation. *IEEE Security & Privacy*, 13(5):16–23, 2015.

[5] P. Black, I. Gondal, and R. Layton. A survey of similarities in banking malware behaviours. *Computers & Security*, 2017.

[6] C. Bravo-Lillo, L. Cranor, S. Komanduri, S. Schechter, and M. Sleeper. Harder to ignore. *Revisiting pop-up fatigue and approaches to prevent it, USENIX Association*, pages 105–111, 2014.

[7] C. Bravo-Lillo, S. Komanduri, L. F. Cranor, R. W. Reeder, M. Sleeper, J. Downs, and S. Schechter. Your attention please: designing security-decision uis to make genuine risks harder to ignore. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, page 6. ACM, 2013.

[8] D. Canali, D. Balzarotti, and A. Francillon. The role of web hosting providers in detecting compromised websites. In *Proceedings of the 22nd international conference on World Wide Web*, pages 177–188. International World Wide Web Conferences Steering Committee, 2013.

[9] O. Cetin, C. Ganán, M. Korczynski, and M. van Eeten. Make notifications great again: learning how to notify in the age of large-scale vulnerability scanning. In *16th Workshop on the Economics of Information Security (WEIS 2017)*, 2017.

[10] O. Cetin, M. Hanif Jhaveri, C. Gañán, M. van Eeten, and T. Moore. Understanding the role of sender reputation in abuse reporting and cleanup. *Journal of Cybersecurity*, 2(1):83–98, 2016.

[11] ECO Internet industry association. Botfree. `https://www.botfree.eu/en/aboutus/information.html`, 2013.

[12] European Network and Information Security Agency (ENISA). Involving Intermediaries in Cyber-security Awareness Raising. `https://www.enisa.europa.eu/publications/involving-intermediaries-in-cyber-security-awareness-raising`, 2012.

[13] M. Fagan and M. M. H. Khan. Why do they do what they do?: A study of what motivates users to (not) follow computer security advice. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 59–75, 2016.

[14] A. Forget, S. Pearman, J. Thomas, A. Acquisti, N. Christin, L. F. Cranor, S. Egelman, M. Harbach, and R. Telang. Do or do not, there is no try: user engagement may not improve security outcomes. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 97–111, 2016.

[15] International Telecommunication Union (ITU). ITU Botnet Mitigation Toolkit. http://www.itu.int/ITU-D/cyb/cybersecurity/projects/botnet.html, 2007.

[16] International Telecommunication Union (ITU). ITU Botnet Mitigation Toolkit. https://www.itu.int/ITU-D/cyb/cybersecurity/projects/botnet.html, 2018.

[17] Jilani, Umair. The ACMA and Internet providers working together to combat malware. https://www.acma.gov.au/theACMA/engage-blogs/engage-blogs/Cybersecurity/The-ACMA-and-internet-providers-working-together-to-combat-malware, 2015.

[18] S. Kim and M. S. Wogalter. Habituation, dishabituation, and recovery effects in visual warnings. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 53, pages 1612–1616. Sage Publications Sage CA: Los Angeles, CA, 2009.

[19] K. Krol, M. Moroz, and M. A. Sasse. Don't work. Can't work? Why it's time to rethink security warnings. In *Risk and Security of Internet and Systems (CRiSIS), 2012 7th International conference on*, pages 1–8. IEEE, 2012.

[20] M. Kührer, T. Hupperich, C. Rossow, and T. Holz. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In *USENIX Security Symposium*, 2014.

[21] F. Li, G. Ho, E. Kuan, Y. Niu, L. Ballard, K. Thomas, E. Bursztein, and V. Paxson. Remedying Web Hijacking: Notification Effectiveness and Webmaster Comprehension. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1009–1019. International World Wide Web Conferences Steering Committee, 2016.

[22] J. Livingood, N. Mody, and M. O'Reirdan. Recommendations for the Remediation of Bots in ISP Networks (RFC 6561). *Internet Eng. Task Force*, 2012.

[23] A. Mathur, J. Engel, S. Sobti, V. Chang, and M. Chetty. They Keep Coming Back Like Zombies: Improving Software Updating Interfaces. In *SOUPS*, pages 43–58, 2016.

[24] Messaging Anti-Abuse Working Group and others. Abuse Desk Common Practices. https://www.m3aawg.org/sites/default/files/document/MAAWG_Abuse_Desk_Common_Practices.pdf, 2007.

[25] Messaging Anti-Abuse Working Group and others. M3AAWG best practices for the use of a walled garden. https://www.m3aawg.org/documents/en/m3aawg-best-common-practices-use-walled-garden-version-20, 2015.

[26] National Institute of Standards and Technology. Models To Advance Voluntary Corporate Notification to Consumers Regarding the Illicit Use of Computer Equipment by Botnets and Related Malware. https://www.nist.gov/itl/upload/SANS_BotNet-FRN-Comment-11-4-11.pdf, 2011.

[27] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 51–65. IEEE, 2007.

[28] B. Stock, G. Pellegrino, F. Li, M. Backes, and C. Rossow. Didn't You Hear Me?–Towards More Successful Web Vulnerability Notifications. In *The Network and Distributed System Security Symposium (NDSS)*, 2018.

[29] B. Stock, G. Pellegrino, C. Rossow, M. Johns, and M. Backes. Hey, you have a problem: On the feasibility of large-scale web vulnerability notification. In *USENIX Security Symposium (Aug. 2016)*, 2016.

[30] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of ssl warning effectiveness. In *USENIX security symposium*, pages 399–416, 2009.

[31] R. Wash, E. Rader, K. Vaniea, and M. Rizor. Out of the loop: How automated software updates cause unintended security consequences. In *Symposium on Usable Privacy and Security (SOUPS)*, pages 89–104, 2014.

# APPENDIX
# A. WALLED GARDEN LANDING PAGE

**Secure environment**

A safe Internet is in everyone's interest. We strongly care about protecting your (confidential) information.

We have received information from one of our partners that a security issue has been detected on your Internet connection. You probably have not noticed anything yet.

Don't worry. To protect you against the security risks we have placed your Internet connection in our secure environment. In this environment you can safely solve the security issues. We are willing to help you to do so.

**What is the problem and how can you solve it?**
One or more computers connected to your Internet connection are infected with a virus.

We kindly ask you to follow the steps to remove viruses on all computers/laptops as described on:

https://address.com

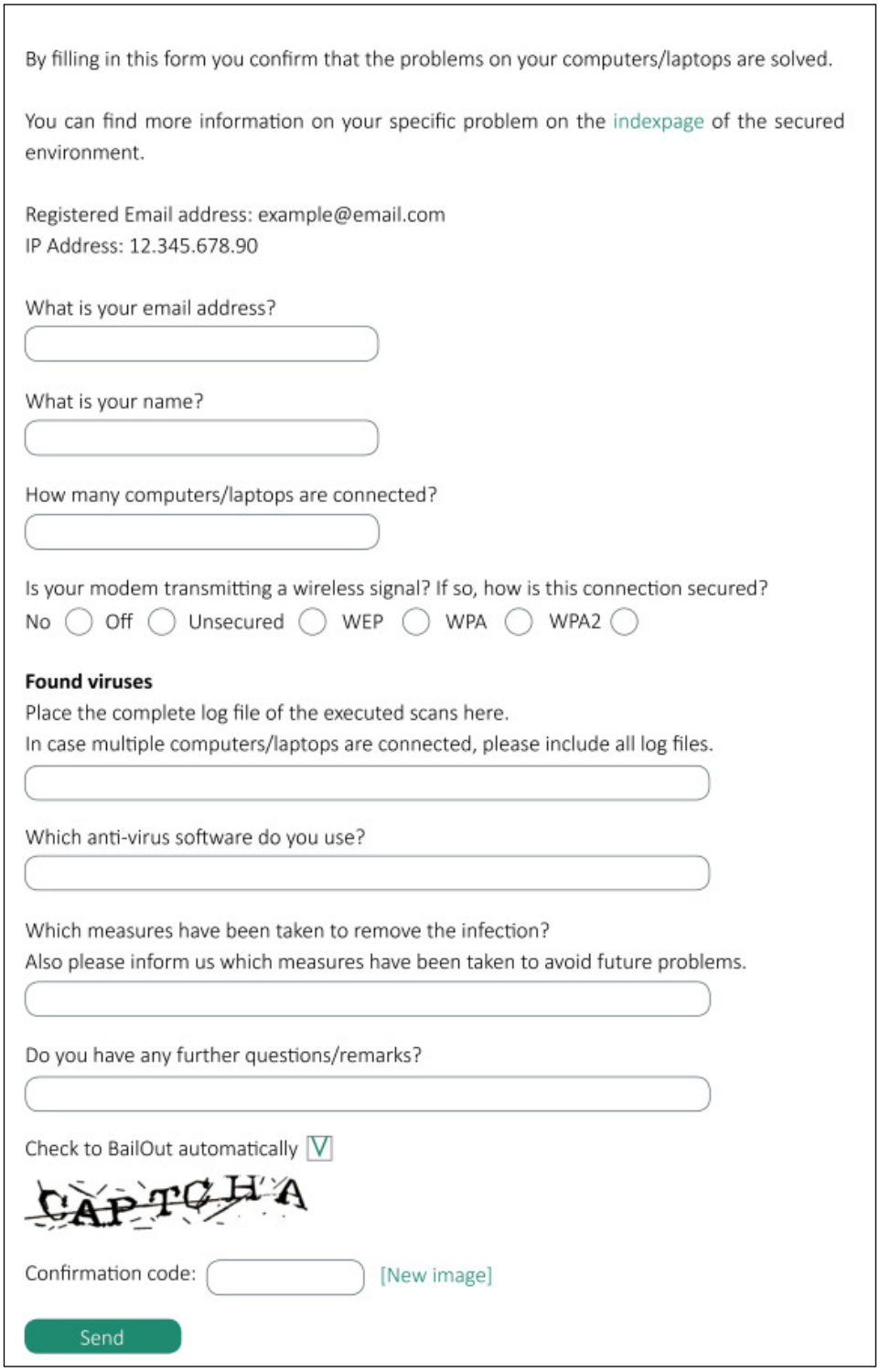
When the scan has finished the program will create a log file with the scan results. Please send us the content of this log file(s).

We would like to be informed what measures have been taken to make sure this abuse will not take place again.

**Necessary steps**
1. Take the measures stated above
2. Fill in our form (and restore your Internet connection)

**General security tips**
* Use an up-to-date virus scanner to keep out potential hazards
* Keep computer software, like your operating system, up to date
* Do not open messages and unknown files that you do not expect or trust
* Secure your wireless connection with a unique and strong password

## B. WALLED GARDEN RELEASE FORM



By filling in this form you confirm that the problems on your computers/laptops are solved.

You can find more information on your specific problem on the indexpage of the secured environment.

Registered Email address: example@email.com
IP Address: 12.345.678.90

What is your email address?

What is your name?

How many computers/laptops are connected?

Is your modem transmitting a wireless signal? If so, how is this connection secured?
No ◯ Off ◯ Unsecured ◯ WEP ◯ WPA ◯ WPA2 ◯

**Found viruses**
Place the complete log file of the executed scans here.
In case multiple computers/laptops are connected, please include all log files.

Which anti-virus software do you use?

Which measures have been taken to remove the infection?
Also please inform us which measures have been taken to avoid future problems.

Do you have any further questions/remarks?

Check to BailOut automatically ☑

CAPTCHA

Confirmation code: [New image]

Send