Filters that Fight Back Revisited: Conceptualization and Future Agenda

Sampsa Rauti and Samuli Laato

Department of Computing, University of Turku, Finland
sjprau@utu.fi, sadala@utu.fi

Abstract. Online scams, unsolicited advertisements, messages containing malicious files and other forms of spam continue to be a nuisance in today's internet, wasting users' time and causing financial damage to companies and organizations. There have been many proposals on how spam should be stopped, from various kinds of spam filters to legislative measures. One of the more extreme suggestions is fighting back by bombarding spammers' servers with a deluge of HTTP requests. In the current study, we revisit this idea "filters that fight back" originally proposed by Graham in 2003, and investigate why the approach has received little attention recently. We also showcase an example solution that automatically sends false information back to spammers by filling forms on their websites or replying to mail addresses they have provided. We offer a conceptualization and future agenda of filters that fight back, and discuss the ethical and technical challenges related to this solution.

Keywords: spam filters, filters that fight back, offensive defense, cybersecurity, offensive security

1 Introduction

Unsolicited messages sent to a large number of recipients, prominently referred to as spam [8], have been a major nuisance in the internet almost ever since its conception. These messages can be, for example, commercial advertisements, attempts to obtain users' personal information or messages related to financial fraud. Spam messages regularly contain links to dubious web pages built with the goal of phishing or distributing malicious software.

Spam can be annoying to deal with and wastes recipients' time [20]. Thus, significant efforts have been devoted to prevent spam messages from reaching their destination, for example, by using spam filters and restricting spam with legislation. During the 90's these filters were mostly rule-based, but today more complex solutions such as statistical spam filters [19] and filters based on artificial neural networks [1] are widely used by prominent email service providers and sometimes also by end users. Despite these countermeasures, users still react and respond to spam messages in high enough numbers to make spamming a profitable business. Botnets such as Necurs [4] have been employed by spammers to effectively and effortlessly send large numbers of spam messages.

During the 90's, the number of spam emails increased steadily, amounting for as much as 90% of all email in 2008 [11]. More recently, unsolicited messages have gone down to 50–60% of all email traffic [23, 15]. Today, spam campaigns have been moved to other mediums besides email, for example, to instant messaging applications, phone calls and social media platforms [2].

While spam filters can analyse spam messages and use them as data for training machine learning models to better detect and block spam [9], most spam filters simply settle for deleting spam messages after they have been detected, taking no further action against the perpetrator. This paper explores the idea of punitive spam filters that aim to incur a cost to a spammer each time they send junk messages. We explore the idea of filters that put a strain on spammers' servers by bombarding their servers with HTTP requests [13]. This idea was originally proposed by Graham in 2003 [13], but contains several ethical and technical challenges which make using such techniques non-straightforward.

In this study, we first talk about Grahams' proposal in detail and go through what has been studied in the academic field on filters that fight back (FFB) back since then. Subsequently, we present an example solution created for this manuscript that illustrates the idea of filters that fight back. This solution fills forms on spammers' websites with fake information or alternatively provides automated bogus answers to spam messages. We follow this example with discussion on the technical, ethical and legal challenges of FFB as well as the benefits of such solution. We conclude the work by providing a future agenda for researchers and engineers interested in FFB.

2 Filters that fight back - the current view

The original concept of FFB which Paul Graham talked about on his website [13] proposed the idea of punishing spammers by sending bogus data back to them. The solution would be implemented by adding a "punish mode" feature to spam filters [13, 24]. When turned on, this mode launches a counterattack on spammers by opening all the URLs in a spam message N times (N is 0 or greater, chosen by the user). The web pages are crawled, that is, all the links on the found pages are followed (which can be repeated for k levels of links) [12]. Consequently, sending huge amount of spam messages now works against spammers, flooding their servers with HTTP requests increasing the bandwidth usage and inflating the costs of maintaining a webpage. The deluge of requests can also make the spammer's servers unavailable to those users who would otherwise have responded to the spammer in good faith and fallen into the scam. If the spammer churns out a million messages an hour, they will potentially receive millions of hits an hour on their servers. This would make operating scams through spamming unremunerative.

Although a URL sent to millions of people is likely to be an address of a spam page, it is important to ensure that HTTP requests are only launched against spam pages. Graham [12] suggested only crawling sites that are on a special blacklist. Web pages are blacklisted only after being inspected by humans. As

a spam message has a lifetime of couple of hours at least, the blacklist can be updated in time to ensure counter-spam measures can be activated against the adversary.

The challenge from the spammers' perspective is the fact that to reach a few gullible recipients who will reply, the spammer needs to send messages to tens of thousands of recipients if not more. FFB has the benefit of enabling the non-gullible majority to make it more difficult for most ductile users to fall for scams. Here one additional potential positive consequence for users is, that in order for spammers to protect their servers against a deluge of HTTP requests, they could be prompted to provide their victims an "unsubscribe" option. This would free spammers from counter-spam and enable recipients to free themselves from the spammers' mailing list. [12]. The problem here of course is that as the tech-savvy spam respondents unsubscribe, the more gullible victims would remain in spammers' mailing lists.

An idea similar to Graham's scheme was used in practice when Lycos Europe web portal launched a screensaver that sent HTTP requests to websites that were known to be promoted in unsolicited mail messages [14]. An advertisement on Lycos Europe's urged the users to "annoy a spammer now!". As the percentage of spam mails has decreased over the years, it appears these kinds of campaigns have became less commonplace. One of the reasons for this could be that modern machine learning-based spam filters have evolved to be so efficient in screening and deleting spam emails [9] that no further action is needed. An additional reason could be that spammers are choosing to host their websites and spamming operation on servers with a flat rate charge. As an extreme example, if an FFB solution would be implemented on an web hosting company X email service, we could see a case of web hosting company X filters sending counter-spam to a web hosting company X server hosted by spammers. Therefore, it is not in the interest of web hosting company X to use FFB.

One of the most recent examples of FFB technical implementations comes from a bachelor project carried out at the Delft University of Technology. In this project, Bansagi et al. [3] created a system that recommends replies that are sent back to spammers, making it easy to waste a scammer's time and money. The authors developed a Google Chrome plugin to enable quick replies to spam emails. This solution, however, does not include offensive defense in the form of crawling the scammer's website.

Spam emails are typically sent using botnets consisting of infected machines. That is why it is often difficult to directly target this infrastructure built by spammers. By targeting the spammers' dubious websites and feeding them fabricated information, their operations can be disrupted. Although tech companies such as Microsoft have managed to take down huge botnets such as Necurs [4], smaller scale operations can still be beneficial. Hence, here we focus on attacking spammers' websites.

3 Our example solution

For the purpose of illustrating how FFB could work in the modern online ecosystem, we created our own conceptual solution that wastes spammers' time by automatically sending fake information to them. After all, spammers usually aim to collect information about their victims. In what follows, we discuss a conceptual solution for sending fallacious data through web forms and email. Figure 1 shows the general idea of our solution.

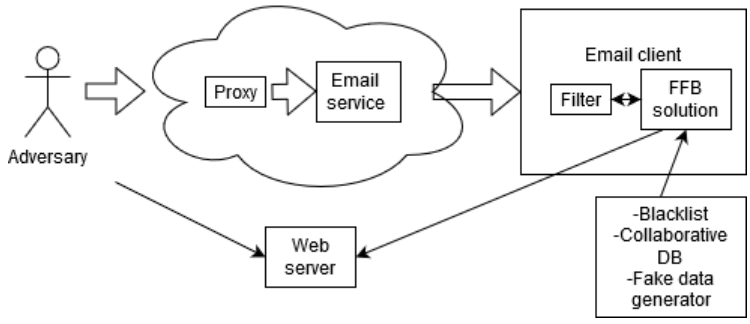

Fig. 1. An overview of how FFB operates. The FFB module responds by targeting the malicious website to which the scammer tries to lure gullible victims.

3.1 The algorithm

A high abstraction level skeleton of a general algorithm for FFB implementation that is invoked when an email is identified as spam, and which crawls weblinks given in spam messages, is described below in pseudocode:

```
If the mail is a spam message
  If the mail contains URLs
    For each URL
      If the URL is on the blacklist
        Crawl each subpage of the website K levels deep
          Load the subpage N times
          If the subpage contains a form
            Fill in fake information M times
  If the mail contains a form
    Respond with fake information
```

If the spam filter classifies a received mail as spam, the mail is checked for URLs. Each URL is then tested against a blacklist, and if the URL is on the

list, it is chosen for further inspection. The subpages under the main address are crawled and loaded N times, following Graham's scheme. This is done K levels deep, meaning all possible link chains (with the length of *K-1* or smaller) from the main page are followed. However, if there are a huge number of subpages, this process could be stopped after a specific number of pages to avoid needlessly wasting our own time and resources. Also, the punitive functionality suggested by Graham could be completely turned off by setting N to 0.

Each crawled subpage is also checked for forms. If a form for collecting a user's information is found, it is filled with fake information. This can be made several (M) times, but depending on the checks implemented on the spammer's server, the form might only be accepted once. The process of generating fake information and filling in the form is further discussed in the next section.

Finally, the body of the received email can also be checked for forms. This is not usually a HTML form but a list of details that the spammer wants the user to fill in and reply to an email address. Using the same fake data generation functionality as previously on the spammer's website, fallacious information is created and sent to the email address provided by the scammer.

It is worth noting that the conceptual solution we have presented employs many other components: an email client, a spam filter, a blacklist, and a fake data generator. Still, our scheme is independent of how these other components have been implemented. The solution could be added to an email client like Mozilla Thunderbird as an extension. The spam filter and the blacklist can use any of the currently available approaches as long as they are accurate enough so that they do not produce a significant number of false positives. The fake data generator can be a part of the implementation or a component implemented by a third party. It has to be able to generate believable bogus names, addresses, phone numbers etc. Our solution can also be applied to other types of spam messages such as SMS spam or spam in Facebook or Twitter.

3.2 Filling in the forms

An important part of the discussed conceptual solution is filling in the forms on spammers' websites. Web forms are typically included in phishing websites which aim to steal victims' credit card information or other details. The entities in the form (such as name address, phone number) can be recognized by using relatively simple rules such as looking at the descriptions and names of the form fields seeing if they match known entity names. If the meaning some form field cannot be recognized, we can simply fill it with random content. The filters could also use human assistance for finding the type of some specific field and then collaboratively share this information to other filters that are also completing the same form.

After the form has been found and the types of the fields have been decided, the fake data generator will generate an appropriate input for each field. This kind of fake data generating component has to use a large list of possible values in order to make the fake details convincing [16, 17]. It has to be able to generate

wide variety of valid values such as addresses, phone numbers and social security numbers.

If the form submission fails, we can retry submitting it a certain number of times. In many cases, the red text indicating an error near a form field can be used to guess which field was not accepted, and a different value can be chosen for that field. The filters can also share information on what kinds of values were successfully accepted for a specific form.

One challenge is the fact that many forms which require credit card information. A seemingly valid fake credit card number can be generated but if the spammer's system immediately attempts to charge the credit card, deceiving the fraudster will not succeed. Checking the validity of the provided card number wastes spammer's resources (computational power or human effort).

There is also an interesting side effect when feeding false information to spammers. If some unique piece of fake information – also called honeytoken [5, 22] – is included in the data given to spammers, it could later resurface somewhere else. Consequently, planting honeytokens can help in tracking and attributing spammers [21, 18], as well a finding out where they sell the information.

4 Discussion

Supplying spammers with fake information poisons their database and wastes their time. Perpetrators can no longer be sure which entries fake are and which are not. This resembles scambaiting, which aims to waste the scammers time and resources by exchanging messages with them, but in our scheme the whole process is automatic. We summarize the key findings of the potential damage FFBs cause spammers below:

- wastes computational resources and bandwidth
- wastes spammer's time (when the obtained information is processed manually)
- prevents gullible users from becoming victims of phishing
- may cause software development related costs when the spammers have to fix their website or information gathering model so that poisoning becomes more difficult
- damages the spammer's reputation as a business partner if poisoned low-quality data is sold

4.1 Technical challenges

A potential challenge with our scheme is the fact that clicking links in spam mails and replying to spammers often causes them to send more spam mail. However, when this happens, the punitive filter will punish them even more, and as long as the spam filter works, the user does not see increasing amount of spam. The increased bandwidth consumption should not be a problem with modern broadband connections.

Another potential weakness of the solution is that blacklists are prone to abuse. While our scheme is independent of the implementation of blacklist, it is important to ensure that untrustworthy individuals or spammers themselves cannot easily poison the blacklist with entries that should not be there. Also, simply being on the blacklist does not cause a website any problems, it only gets hit when it is already blacklisted and a new spam message arrives with a link to the site arrives [12].

4.2 Ethical and legal issues

One can argue that this kind of offensive defense and "striking back" is immoral or illegal [6]. It is not completely clear whether loading spammer's webpages a few times means participating an organized denial of service attack or whether automatically filling spammers' web forms constitutes any kind of offense. In some jurisdictions, however, the user might be rendered legally liable.

Spammers may not be likely to take the matter to court, but if some innocent party was accidentally targeted, things might be different. Still, the solution we propose is very different from "hacking back", that is, which would mean tracing back to the attacker and invading their system. There are probably very few, if any, precedents pertaining to this kind of offensive defense. Of course, what is permissible from a legal or ethical points of view, also depends on who is doing it. Some kind of authority could also take care of striking back against spammers.

4.3 Benefits

Our solution has some additional benefits compared to Graham's original solution. It not only increases the load for spammers' servers, it also deteriorates the quality of the data they receive. Several spammers and other cybercriminals may try to use the poisoned data if the data is sold. Moreover, the data is potentially made traceable with honeytokens. Our solution is also more likely to waste time of human perpetrators, as the information spammers receive may be manually examined.

If the functionality of loading webpages repeatedly is turned off, our version of a FFB also does not have the problem of launching denial of service attacks in the same sense Graham's solution does. For example, If the spammers website shares a host with some other innocent customers, continuously bombarding the spammer's website can cause needless collateral damage. Our solution, when used without repeated page loads, avoids this problem. Then again, if this functionality is turned off, then the filter does not protect gullible users from falling for the scam as effectively.

The conceptual solution we have presented does not depend on the environment where spam needs to be combated. Along with the email system, our solution can also be used for counter spam messages in social media or text messaging (SMS) spam. Another application is fighting search engine spam [10]. For all these types of spam, filters have been build and our solution could be combined with those filters.

5 Conclusions and Future Agenda

In this paper, we revisited the idea of FFBs introduced by Paul Graham in 2003 [13] that has since been dormant and seen minimal attention in the scientific community. To see whether FFBs are still a viable security measure, we developed Graham's idea further and provided an example case of a spam filter that provides a form of offensive defense by replying to spammers with fake information and wasting their time. The solution works on individual cases, but large-scale application of this approach remains untested. In addition to this example, other versions of FFB adjusted to the modern online ecosystems have been presented (e.g. [3]) but these filters have not been widely adopted in practice either. With regards to evidence as to why FFBs are currently not used, the following main reasons emerge:

- Ethical concerns related to the justification of offensive defense and its large scale operation.
- Security concerns related to misuse (intentional or unintentional) of FFBs.
- Advances made in other spam filters and other technologies for curbing rampant spam messages.
- Spammers' utilization of 3rd party flat rate online services for their schemes, where FFBs would not in fact cause major damage to the spammer.
- Other concerns related to the feasibility and effectiveness of FFBs such as FFBs alerting the scammer that their operation has been detected, and FFBs enabling scammers to reverse engineer ML-based filters.

While these certainly seem to explain the lack of use of FFBs, it might still be early to throw away the idea of offensive defense against online spam. In fact, recently we have seen the rise of scambaiting, that is, online streamers and content creators making fun of scammers and wasting their time [25]. This activity takes a harmful activity (scamming) and turns it into popular entertainment. While scambaiting also has obvious ethical concerns, in principle it serves to educate people about scamming, cause harm on scamming as a business and protecting potential scam victims, in addition to being entertaining. For these reasons, we believe that FFB should also deserve further attention from the scientific community to see whether it can be applied to make the internet a safer environment. Here we would like to provide a future agenda related to interesting research topics and questions in the domain of FFB.

First, an empirical longitudinal analysis of the consequences of using FFBs in the large scale could be carried out. This analysis needs to focus on both the impact FFBs have on spammers and scammers, and the impact FFBs have on service providers. To this end, a FFB implementation of our solution and field work testing of it is required. Here we identify legislative and ethical challenges in addition to the technical. Second, further analysis is required on whether FFBs could be abused by making them target innocent or trusted parties. Third, the possibility of sending honeytokens to the scammers' system and being able to track information that the scammers have is another promising future research

avenue. For example, this could enable discovering which scammers are connected to one another. Fourth, the ethics and lawfulness of offensive defense deserves attention with regards to FFBs, scambaiting and other forms of means to retaliate. Online vigilantism, sometimes discussed as digilantism [7] which includes FFBs, remains in many regards problematic. While in the ideal situation authorities would take care of cybercrime, there are many reasons as to why this is not currently the case. These reasons include lack of technical skills of the authorities, the global scale of the cyberworld where people operate in the same environment under different sets of laws and the rapidly changing and evolving nature of the internet.

References

1. Alghoul, A., Al Ajrami, S., Al Jarousha, G., Harb, G., Abu-Naser, S.S.: Email classification using artificial neural network. (2018)
2. Almeida, T.A., Silva, T.P., Santos, I., Hidalgo, J.M.G.: Text normalization and semantic indexing to enhance instant messaging and sms spam filtering. Knowledge-Based Systems **108** (2016) 25–32
3. Bansagi, A., Bes, R., Garama, Z., Gosshalk, L.: The scam filter that fights back. https://repository.tudelft.nl/islandora/object/uuid\%3A6099061a-4ca7-469b-b9c0-86d7bc0f52e3 (2016) Accessed: 2020-11-05.
4. Bapat, R., Mandya, A., Liu, X., Abraham, B., Brown, D.E., Kang, H., Veeraraghavan, M.: Identifying malicious botnet traffic using logistic regression. In: 2018 Systems and Information Engineering Design Symposium (SIEDS). (2018) 266–271
5. Bercovitch, M., Renford, M., Hasson, L., Shabtai, A., Rokach, L., Elovici, Y.: Honeygen: An automated honeytokens generator. In: Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics, IEEE (2011) 131–136
6. Bradbury, D.: Offensive defence. Network Security **2013**(7) (2013) 9–12
7. Byrne, D.N.: 419 digilantes and the frontier of radical justice online. Radical History Review **2013**(117) (2013) 70–82
8. Cormack, G.V.: Email spam filtering: A systematic review. Now Publishers Inc (2008)
9. Crawford, M., Khoshgoftaar, T.M., Prusa, J.D., Richter, A.N., Al Najada, H.: Survey of review spam detection using machine learning techniques. Journal of Big Data **2**(1) (2015) 23
10. Egele, M., Kolbitsch, C., Platzer, C.: Removing web spam links from search engine results. Journal in Computer Virology **7**(1) (2011) 51–62
11. Farrell, N.: Cisco says that 90 percent of email is spam. https://www.theinquirer.net/inquirer/news/1050056/cisco-90-percent-email-spam (2008) Accessed: 2019-03-17.
12. Graham, P.: FFB FAQ. http://www.paulgraham.com/ffbfaq.html (2003) Accessed: 2020-09-13.
13. Graham, P.: Filters that fight back. http://www.paulgraham.com/ffb.html (2003) Accessed: 2020-09-13.
14. Hines, M.: Lycos Europe: 'Make love not spam'. https://www.cnet.com/news/lycos-europe-make-love-not-spam/ (2004) Accessed: 2020-09-16.
15. M. Vergelis, N. Demidova, T.S.: Spam and phishing in Q3 2018. https://securelist.com/spam-and-phishing-in-q3-2018/88686/ (2018) Accessed: 2020-09-13.

16. Papalitsas, J., Rauti, S., Tammi, J., Leppänen, V.: A honeypot proxy framework for deceiving attackers with fabricated content. In: Cyber Threat Intelligence. Springer (2018) 239–258
17. Papalitsas, J., Tammi, J., Rauti, S., Leppänen, V.: Recognizing dynamic fields in network traffic with a manually assisted solution. In: World Conference on Information Systems and Technologies, Springer (2018) 208–217
18. Rauti, S.: Towards cyber attribution by deception. In: International Conference on Hybrid Intelligent Systems, Springer (2019) 419–428
19. Rusland, N.F., Wahid, N., Kasim, S., Hafit, H.: Analysis of naïve bayes algorithm for email spam filtering across multiple datasets. In: Proceedings of the IOP Conference Series: Materials Science and Engineering. (2017)
20. Siponen, M., Stucke, C.: Effective anti-spam strategies in companies: An international study. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06). Volume 6., IEEE (2006) 127c–127c
21. Spitzner, L.: Honeypots: Tracking Hackers. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2002)
22. Spitzner, L.: Honeytokens: The other honeypot. http://www.symantec.com/connect/articles/honeytokens-other-honeypot, (2003)
23. Statista: Spam: share of global email traffic 2014-2018. https://www.statista.com/statistics/420391/spam-email-traffic-share/ (2018) Accessed: 2020-09-13.
24. Zdziarski, J.A.: Ending spam: Bayesian content filtering and the art of statistical language classification. No starch press (2005)
25. Zingerle, A.: Scambaiters, human flesh search engine, perverted justice, and internet haganah: Villains, avengers, or saviors on the internet? In: ISEA Conference. (2015)