An Empirical Study on the Blocking of HTTP and DNS Requests at Providers Level to Counter In-The-Wild Malware Infections

Marcus Botacin[1] , **Paulo de Geus**[2] , **André Grégio**[1]

[1] Federal University of Paraná (UFPR)

{mfbotacin, gregio}@inf.ufpr.br

[2]University of Campinas (UNICAMP)

paulo@lasca.ic.unicamp.br

Resumo. *Enquanto o bloqueio de exemplares de malware em nível de endpoint é essencial para o tratamento de ameaças recém-criadas, o controle de ameaças em larga escala só é possível através do bloqueio dos serviços de rede utilizados por estas. Este estudo se propõe a investigar como este tipo de bloqueio ocorre na prática, visando identificar os problemas existentes e oportunidades de desenvolvimento. Para tanto, consideramos a análise diária de milhares de exemplares de malware de duas bases de dados distintas (representativas de exemplares brasileiros e mundiais) e investigamos a prevalência e a forma com que os servidores HTTP e DNS contatados pelos exemplares são tornados indisponíveis. Nós identificamos que: (i) A indisponibilização de serviços para todos os exemplares ocorre de forma semelhante, com os exemplares brasileiros sendo primeiramente afetados; (ii) ameaças armazenadas em servidores de nuvem computacional são indisponibilizadas de modo diferente das ameaças armazenadas em servidores dedicados; (iii) o bloqueio da resolução de nomes de domínio tem uma escalabilidade maior do que o bloqueio de tráfego HTTP de modo individualizado; e (iv) ainda há um grande espaço para desenvolvimentos futuros visto que a maior parte das ameaças não teve nenhum serviço indisponibilizado durante o período de observação.*

Abstract. *Although blocking samples at endpoint level is still essential to respond to newly created threats, large-scale threat blocking is only possible at network level. We propose investigating how such blocking procedures take place in actual scenarios aiming to identify and bridge existing development gaps. We considered daily analysis results of two malware datasets (representatives of the Brazilian and the World scenarios) and investigate the prevalence and the methods leveraged to make the downloaded HTTP payloads and the resolved DNS domains unavailable to the malware samples. We discovered that: (i) The servers contacted by all samples are sinkholed in similar ways, but Brazilian samples were first affected; (ii) cloud-stored samples are blocked in distinct manners than the ones stored in private servers; (iii) blocking DNS resolution of malicious domains is more efficient than blocking individual HTTP-retrieved payloads; and (iv) there is still open development gaps in network security as most samples had no contacted domain sinkholed at any time.*

1. Introduction

Malware are constant threats in most user's lives and the samples are becoming each time more dependent on network resources. There are even a whole class of threats that base their operation on the download of external components. These malware downloaders [Rossow et al. 2013] are often seem in complex infection campaigns that involve the infection of multiple machines to build their distributed botnets that are leveraged to deliver malware according to attacker's demands.

The major defense line against malware, including downloaders, is the use of Antiviruses' endpoints. However, although blocking samples at endpoint level is still essential to respond to newly created threats, blocking large-scale threats such as downloaders and botnets is only possible at network level, as it allows protecting all potential victims with a single blocking action. Previous work demonstrated that entire botnets might be defused via network traffic blocking measures [Stone-Gross et al. 2009]. The measures taken to block malicious traffic are often called as *takedowns*, *takeovers*, or *sinkhole* procedures [Hunt and Zeadally 2012]. Despite slightly different according to the context, these terms are interchangeably used in this work.

Malicious traffic blocking was proposed and implemented by leveraging multiple methods, such as Antivirus blocking, shared blacklists [Sucuri 2019], or at provider level [Locaweb 2019]. We believe that before proposing developing a new solution, we need to understand what are the current gaps in existing proposals. Therefore, in this work we present an investigation on how threat is blocked at provider's level. We hope our findings might then be further used to assist on the development of new, custom blocking solutions.

In particular, we propose investigating how such blocking procedures take place in actual scenarios, considering both the stored payloads as well as the routes to reach it. In other words, we investigate the takedown of HTTP payloads by host providers and the blocking on the resolution of malicious domain names by nameservers providers. For our experiments, we considered daily analysis results of the execution of more than 30 thousand malware samples on a sandbox solution. The malware samples were collected from two malware datasets that are representative, respectively, of the Brazilian and the World scenarios.

We discovered that: (i) The servers contacted by all samples are sinkholed in similar ways, but Brazilian samples were first affected; (ii) Many Brazilian samples are stored in cloud providers and these samples are blocked in distinct manners than the ones stored in private servers; (iii) blocking DNS resolution of malicious domains is more time-efficient than blocking individual HTTP-retrieved payloads; and (iv) there is still open development gaps in network security as most samples did not have any contacted domain sinkholed at any time.

Our findings indicate that having an efficient mechanism for reporting abuse is essential to allow providers to block malicious content. Whereas such mechanism was identified in some providers [Amazon 202, Google 2020], this is not the case for all hosts, which highlights the need for improvements.

In summary, our contributions are the following:

- We present an evaluation of the blocking of HTTP and DNS malicious requests from actual malware infections.
- We discuss the best strategies to block malicious traffic at multiple levels, which encompasses payloads stored at cloud providers.
- We present recommendations to improve the report of network abuse to service providers.

This work is organized as follows: In Section 2, we present related work to better position our contributions; In Section 3, we present the methodology adopted to conduct the presented experiments; In Section 4, we present experiments results; In Section 5, we discuss our findings; Finally, we draw our conclusions in Section 6.

2. Related Work

In this section, we present related work to better position our contributions.

2.1. Malware & Networks

In this work, we evaluate domain takedown [Hunt and Zeadally 2012]—-in the sense of malicious domains being blocked. This evaluation is of particular importance because modern malware often presents itself in the form of downloaders [Rossow et al. 2013]. The downloaders are used by criminals in pay-per-install campaigns that deliver malicious payloads as a service. Shutting down these domains is essential to provide an scalable counter-measure against malware downloaders. The greater the response time to take down a domain, the more users may get infected by the distributed threats.

2.2. Malicious Traffic Analysis

Much research work is dedicated to detect malicious traffic [Liu et al. 2017, Parekh et al. 2006]. Despite its importance, our focus in this paper is a bit different. We focus on evaluating whether the malicious domains involved in malware infections are detected in practice and how long it takes for such hosts to be neutralized. In this sense, our work resembles Hao et al.'s [Hao et al. 2011], which looks to malicious domains to understand how long after creation the domains are used for launching attacks. In comparison, our observations take into account only from the moment that the attacks are launched.

2.3. Domain Sinkhole

Previous work on sinkhole were focused on tracking botnets [Wang et al. 2016], as this kind of threat is network-intensive. More specifically, previous work on botnet sinkhole focused on individual threats [Stone-Gross et al. 2009] whereas this work focuses on a broader scenario. These work pointed that modern malware may rely on the fast flux technique to change resolved DNS names and thus keep their infections active [McGrath et al. 2009]. This work aims to track the changes on the analyzed malware samples. We expect that our findings might help understanding the impact of malicious traffic sinkhole in practice, as previous work performed in other contexts, such as for traffic exchange [Yousaf et al. 2016].

2.4. Malware Traffic Detection

Many work in the literature proposed techniques to detect malware-generated traffic. Previous approaches covered mobile environments [Khatri and Abendroth 2015], the use of machine learning techniques [Yu and Harang 2017], and the classification of HTTP traffic [Aresu et al. 2015]. Whereas related to all these approaches, our work does not focus on detecting malicious traffic. Instead, we focus on assessing the effectiveness of the detection procedures potentially leveraged by the providers and hosts.

3. Methodology

In this section, we present the methodology adopted to conduct the experiments presented in this work.

3.1. Datasets

In this work, we aim to investigate how malicious domains are sinkholed in practice, which required us to look for sources of active malware samples. We had access to two distinct malware sources that provides malware samples on a daily basis: (i) a CSIRT honeypot that provides active Brazilian malware samples; and (ii) the MalShare repository [Malshare 2018] of samples collected world-wide. These two datasets were characterized in previous studies [Ceschin et al. 2018, Beppler et al. 2019] and we considered adequate to this study's goals. In total, we considered 20 thousand Brazilian malware samples and 11 thousand worldwide samples. All samples were distributed by the repositories in the form of Windows PE executables and collected between August/2017 to December/2018. All experiments were performed right after the sample's collection. A significant advantage of leveraging two distinct datasets in this study is that the results of all experiments can be compared with regards the two datasets and thus reduce the uncertainty of our findings.

3.2. Experiments Design

Our experimental approach consisted in daily collecting new samples and running them on our sandbox solution [Botacin et al. 2018]. After the first run, we identify the samples whose contacted domains were not sinkholed yet so that we can add the samples to a daily analysis queue. The samples in this analysis queue are daily re-analyzed in the sandbox until their contacted servers become unreachable, up to the limit of 30 days (a limit identified on a previous work [Botacin et al. 2020]). For each run, we collect the whole network traffic, parse and dissect the data for the DNS and HTTP protocols, and store it on a database. The database is further queried to identify data discrepancies and for metrics calculation, as following presented. The protocol dissection allows us to reason why the domains and servers become unavailable. We grouped eventual error cases in categories: network errors, such as routing problems; HTTP errors, such as `404`, when the page was effectively shut down; and other errors, like when web pages were actually displayed instead of the payload being retrieved, which happens when a `404` wrapper is displayed by the host provider or when a JavaScript page triggers an error.

3.3. Network Metrics

The sinkhole of malicious domains and servers can be analyzed from multiple perspective. We limited our evaluations to the aspects that most affect malware execution and understanding. To accomplish this goal, we defined the following Network Metrics (NM).

- **NM1. Content Sinkhole**. This metric evaluates whether the malicious payloads downloaded by given samples are removed from the hosting servers after some time or keep infecting users. This is particularly important in cases where AV solutions fail to detect a given threat and payload removal is the only defense available to protect users.

- **NM2. Content Survival Time**. This metric evaluates how long the domains contacted by the malware samples remain active before being taken down. This metric helps evaluating whether removal procedures occur in reasonable time. Early removals are desired because AV solutions might take some time to develop signatures to new threats, thus network hosting providers might help reducing the attack opportunity window.

- **NM3. Sinkhole Method**. This metric evaluates how content hosts and/or providers act to remove malicious payloads. The removal method indicates which security scope is affected/involved.

- **NM4. DNS Takeover**. This metric evaluates how many DNS records stop being resolved by the DNS requests performed by the malware samples within the evaluated period. This metric evaluates whether network administrators are blocking identified malicious traffic or not.

- **NM5. DNS/IP Rotation**. This metric evaluates how many distinct IP addresses are resolved to the same DNS queries. This metric evaluates whether attackers rotate domains to make removal harder.

- **NM6. DNS Takeover Time**. This metric evaluates for how long a given DNS query keeps being resolved until blocking. It impacts the attack opportunity of early-launched threats.

3.4. Assumptions

Making assumptions is required to make our analysis practical. These are following presented.

- **Selected Protocols.** Attacks occur on top of all protocols, but analyses should be limited to be made practical, for instance, due to the need of handling huge amounts of data. Therefore, we limited our analyses to the HTTP and DNS protocols are they are likely the most prevalent application protocols on the Internet and also in the Brazilian malware [Botacin et al. 2015].

- **Sinkhole Reasons.** Although we can identify that a domain and/or server become unreachable for a long-time, we cannot be sure who performed the takedown procedure (e..g, providers, attackers, law enforcement). Face that, we assumed the scenario most favorable to the providers. Therefore, in this scope of this work, we assume that any sinkhole effect is due to providers recognizing the maliciousness of the given traffic.

- **Blocked DNS Requests.** Each malware sample perform DNS requests in a distinct manner (e.g., some use the default server, some set custom servers). It is hard to identify at which point each malware sample was sinkholed. Therefore, in this work, we handle DNS servers as a whole. When malware samples set their own servers, these are consider. When no custom server was specified, the Google server was used. Our hypothesis is that the malicious domains will be blocked at some point and propagated to the local resolvers at some point, a scenario which resembles the Internet usage of home-users.

4. Evaluation

In this section, we present experiments results brokendown by the considered network metrics.

4.1. NM1. Content Sinkhole

To evaluate **NM1**, we considered World-Wide (WW) and Brazilian (BR) samples which attempted to perform at least one download of an Internet content. Figure 1 shows the fraction of total samples and payloads that were affected by takedown procedures at least once in any observation day.

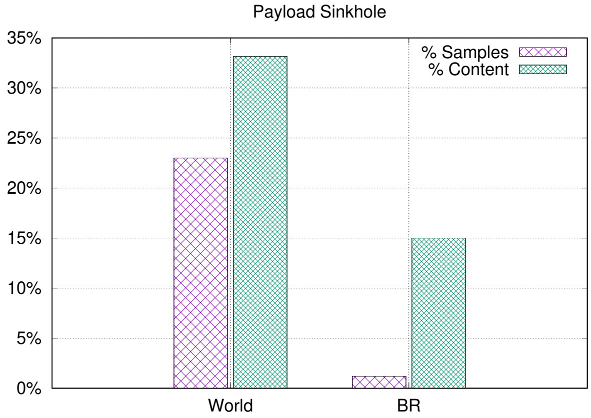

Figure 1. NM1. Takedown of sample and payload contents from World-Wide (WW) and Brazilian (BR) samples. The higher rates of affected payloads in comparison to the number of samples is explained by the fact that some samples download multiple files.

We notice that 23% of WW samples and 1% of BR ones had at least one of their contacted domain blocked, which means that the majority of the samples had their payloads available until the end of our measurement, thus affecting users for a long-term. When we look to individual payloads, such rates are about 33% and 15%, respectively, because some samples work by downloading multiple payloads during their executions, which shows that taking down entire malicious servers might be more effective than blocking individual payloads. We also notice a significant difference between the rates identified for the WW and the BR scenarios: Whereas the WW payloads are spread among multiple hosts, BR samples seem to be more concentrated in fewer hosts, thus the higher number of payloads affected when a server traffic is sinkholed. This can be explained by the fact that most BR samples rely on legitimate cloud providers to store their payloads. On the one hand, this makes legitimate/malicious traffic distinction harder. On the one hand, it reduces attacker's resilience against sinkholes if the payloads are stored in the same cloud server. In our sampling, 70% of BR samples relied on at least one known cloud provider (e.g., Google, Amazon, Locaweb, UOL host, so on) to store their files. This rate is lower than 40% on WW samples.

4.2. NM2. Content Survival Time

To evaluate **NM2**, we repeated the aforementioned analysis during a period of 30 consecutive days. Figure 2 shows the takedown evolution for the WW and BR samples that had at least one of their contacted hosts sinkholed.

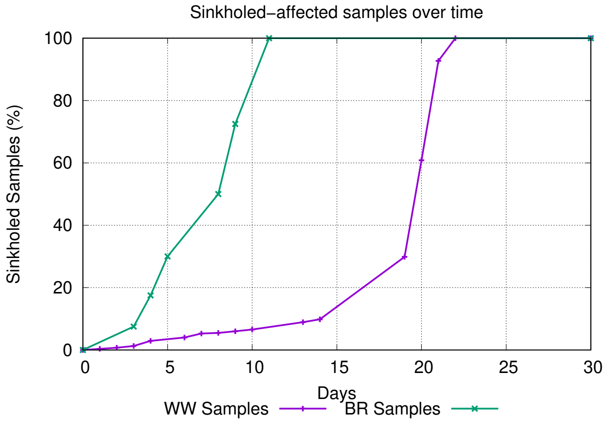

Figure 2. NM2. Payload takedown over time. Despite presenting distinct response times, WW and BR present the same characteristic curves.

We observe that the BR payloads are removed faster than WW ones, which can be explained by the fact that multiple payloads are stored in the same cloud servers, thus easing removal. However, we remark that it covers fewer cases than the WW scenario. As the two curves are very similar, we believe that the time taken would increase for BR samples, in line with the WW scenario, if more samples were considered. The two lines present an initial takedown delay followed by a ramp up, We hypothesized the initial time is taken for hosts receiving notification and performing checks, then the samples are quickly sinkholed. Reducing such initial response time is an open research question. Security-wise, payload removal procedures would help in a scenario where AV solutions had not yet deployed signatures for the new threats. However, this can help only if the removal is faster than signature generation. While this is likely the case for the few BR samples, it is not for the general case, because the takedown procedures took almost the same time to react as AV solutions to generate signatures (20 days), as shown in previous work [Botacin et al. 2020].

4.3. NM3. Sinkhole Method

To evaluate **NM3**, we checked how the samples were take down. For this analysis, we considered only the domains which were not accessible on the 30^{th} day of the observation period. The results are shown in Figure 3.

We observe that the WW scenario presents more connection fail errors—when a given server or network is unreachable—than the BR one. This comes from the fact that WW samples rely on private servers or distributed infected machines to store their payloads and these machines can be easily turned off at any time. In turn, BR samples rely more on cloud servers for payload storage, thus requiring providers and hosts to shut down each malicious page instead of a whole server or network segment. This fact is corroborated when we look to the number of 4xx and 5xx HTTP error codes—higher in the BR than in the WW scenario. The 400 - invalid request error may be related to a page redirection of a so-far malicious webserver. The 403 - forbidden and 500 - internal error messages may be related to cloud servers removing access permission for specific directories—the ones storing the malicious payloads. This is an efficient strategy when the malicious content is delivered by a shared machine, having other customers which should not be affected by a network takedown. As for the 503

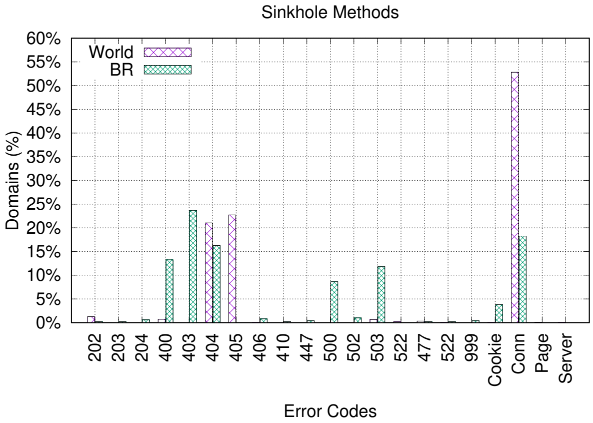

Figure 3. NM3. Takedown methods. `WW` samples present more generic connection errors whereas `BR` ones present more errors related to the use of cloud servers.

`- service unavailable` error, this may be related to a particular VM shutdown instead of quitting the whole physical server connected on the same network.

4.4. NM4. DNS Takeover

In addition to removing the content itself, another way of preventing access to malicious payloads is to remove/reroute malicious DNS entries. To evaluate this metric (**NM4**) in practice, we observed the DNS behavior of the same samples previously presented. Figure 4 shows, respectively, the fraction of samples that had at least one DNS query unresolved in any day and the number of unresolved DNS queries for each sample.

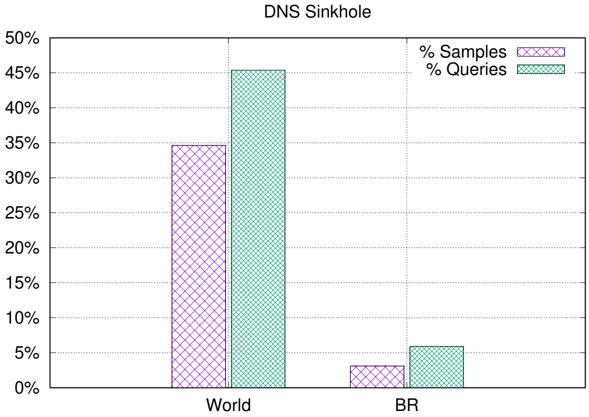

Figure 4. NM4. DNS Sinkhole. `WW` samples are more affected than `BR` ones.

We observe, as in the HTTP's case, that most queries are not met by a taken-down domain. When looking at the number of taken-down domains, the scenario is a little bit better, as each sample performs more than one query. Therefore, a sample that connects to a taken-down domain effects multiple DNS queries. By contrast, `WW` samples are more sinkholed than `BR` ones, which is related to the use of cloud servers, as previously commented. Most samples ($\approx 90\%$) contacted the default DNS server, and few servers contacted their custom servers. We did not notice any difference in the ratio of blocked requests among these, although we noticed that most of custom servers were already unreachable in the first observation day.

4.5. NM5. DNS/IP Rotation

Regarding the queries, we observed that `WW` and `BR` samples received, respectively, a mean of 3.9 and 2.9 distinct DNS answers for each query. However, DNS answers are complex and may include additional information about servers and redirection information. Therefore, to evaluate **NM5**, we considered only IP addresses on distinct network (ranges)—although we cannot guarantee they do not belong to the same company. We obtained 43% and 34% of `WW` and `BR` samples presenting DNS queries resolved to multiple IP addresses. As we were not able to identify and enumerate all legitimate providers which resolve their domain to multiple IPs, we opted to not provide the rate of samples in such category. Instead, we present a qualitative view of exhibited patterns. In summary, we can observe the following network usage patterns[1]:

1. **Content Deliver Networks (CDNs)**: this kind of provider may place their servers on distinct networks, balancing their loads and providing regionalized services. As some malware payloads are stored on cloud providers, we notice this change on resolved IPs, as in `load.s3.amazonaws.com: 54.231.AA.BBB, 52.216.CC.DD`.

2. **Dynamic IP services**: Attackers rely on dynamic domain services to implement their C&C. These services allow changing the registered IP address for the same domain name, as in `iutf.dyndns.org: 216.146.EE.FF, 91.198.GG.HH`. Changing the resolved IP for an existing name allows attackers to reuse their malicious binaries, even when the domain name is hardcoded.

3. **Registered Domains**: Attackers can also implement C&Cs mechanisms for their samples by relying on their own registered domains, such as in `xlscgpqghsxopwceausfyif.ru: 198.105.II.JJ, 104.239.KKK.L`.

4.6. NM6. DNS Takeover Time

To evaluate **NM6**, we repeated the DNS experiment for 30 consecutive days. Figure 5 shows the DNS takedown evolution for the `WW` and `BR` samples that had at least one of their DNS server's requests sinkholed.

As for HTTP, the two lines present similar patterns, with `BR` queries being sinkholed a little bit faster than `WW` ones. A significant difference is that DNS queries are sinkholed faster than HTTP payloads. This is due to the fact that DNS is managed by network providers whereas cloud storage is under attackers' control. In addition, DNS blocking is reversible whereas HTTP content deletion may be permanent if not performed through an access control method. Therefore, we conclude DNS control is a more efficient way to help on threat mitigation procedures.

5. Discussion

In this section, we discuss the implications of our findings and the limitations of our experiments.

[1] We masked IP addresses to not reveal possibly malicious hosts

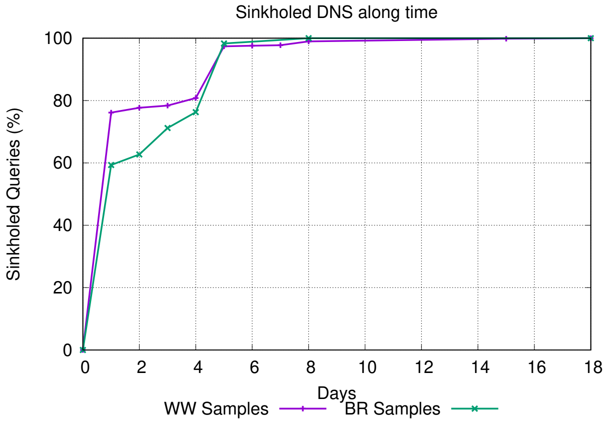

Figure 5. NM6. DNS takedown over time. WW and BR samples presented similar behavior.

Contributions. Much research work has measured Internet traffic and identified malicious traffic, but ours is the first one to categorize traffic coming out from known malicious sample's execution. We highlight the importance of measuring malicious content takedown metrics in addition to antivirus detection, because making this kind of content unavailable helps fighting malicious code in a very scalable manner.

Takedown Method's Effectiveness. Threat control at the network level may be performed by removing malicious payloads, performing server-level blocking, or by avoiding content to be reachable through DNS blocking. Our results have shown the latter to be the most effective approach, because it is under the providers' control. The biggest challenge is to identify malicious servers and automatically block DNS answers for such addresses. This is also an open research question. Currently, some providers started to offer a malware-filtered version of their DNS resolution services [CloudFlare 2020], which we believe to be a good practice.

Takedown at Cloud Servers. HTTP content removal is a hard task in the general scenario as the webservers are often under customers' control, requiring a more detailed analysis before removal, which usually takes longer. However, as the use of cloud providers is widespread, as seen in the BR scenario, we claim they should be involved in threat mitigation. Similarly for DNS, automatic identification and HTTP block procedures also need to be developed. As some providers are already using permission-based blocking, perhaps some kind of quarantine might be developed, allowing content access only after some time. This may counter the ease with which attackers can do their malicious activities, such as quickly deploying and deactivating VMs in cloud providers. Currently, a major challenge to effectively deploy these detectors in the cloud providers is how to inspect the contents stored in the cloud servers without introducing privacy risks to the cloud users.

Reporting Malicious Content is essential to make providers aware of abuse and thus block the content. For such, it is important for providers to provide mechanisms for users reporting suspicious domains and events. We looked for such mechanisms in the top accessed providers. We only found clear ways of reporting abuse in the Amazon [Amazon 202] and in the Google Cloud [Google 2020] servers. Therefore, we recommend for the other providers to make their reporting procedures more accessible.

The Importance of Context. This study revealed an expressive difference on how Brazilian and World samples were sinkholed. These differences were also observed in other scenarios, such as when comparing antivirus detection of Brazilian and World samples [Botacin et al. 2020]. The adoption of distinct strategies for payload removal reflects the distinct strategies employed by attackers to threaten users. The protection/security differences are explained by differences on the local culture, which makes some users more vulnerable to certain attacks than others (e.g., the Internet Banking scenario in Brazil is different than the rest of the world [Botacin et al. 2019]). Therefore, we highlight the importance of this type of study to understand the particularities of targeted user's population. In the Brazilian case, the high usage of cloud servers to distribute malicious payloads is an indicator that cloud-based defenses should be prioritized in this scenario.

Limitations The major advantage of this work over previous investigations is that our data analysis procedures rely on real malicious traffic, thus each reported finding was identified in practice and not only hypothesized. On the other hand, this experimentation design choice brings significant limitations, such as not knowing who effectively sinkholed the traffic. Although our hypothesis that the malicious servers were taken down by the providers is the most plausible, we cannot overlook the fact that in some cases the attackers themselves might have removed their distributed payloads. We also could not scale our analysis to all protocols, thus we limited our investigation to HTTP and DNS traffic. Whereas this choice is backed by the fact that these are likely the most used protocols on the Internet, we cannot overlooked that malicious traffic is also carried via other protocols. One can even hypothesized that the minted attention to these other protocols might facilitate attacks to be carried on top of them. Finally, our evaluation is also limited to look to standard DNS queries, i.e., non-encrypted connections binded to the standard port. Whereas this is the prevalent case, as shown in previous study [Botacin et al. 2015], we must acknowledge that advanced malware might implement their own stubs to carry custom DNS queries to their own servers in non-standard ports, which makes their services more resistant to takedown procedures.

Future Work This work is not an exhaustive analysis of sinkhole procedures. Instead, it presents our view on particular-yet-representative datasets. For future, we plan to expand our analyses to more detasets so as to present stronger evidences of the occurrences of the identified sinkhole events. We plan, for instance, to increase the number of evaluated DNS resolvers to identify whether distinct providers implement distinct blocking policies.

6. Conclusion

In this paper, we conducted an empirical evaluation of the blocking of HTTP payloads and DNS name resolvers for malicious domains in practice. We considered the daily analysis of more than 10 thousand Brazilian and world-wide samples for a period of up to 30 days. We discovered that (i) The servers contacted by all samples are sinkholed in similar ways, but Brazilian samples were first affected; (ii) cloud-stored samples are blocked in distinct manners than the ones stored in private servers. This can be explained by the fact that the malicious VM might be hosted in the same server and/or network as the legitimate VMs; (iii) blocking DNS resolution of malicious domains is more efficient than blocking individual HTTP-retrieved payloads, as many samples rotate the IP address resolved for the same name over time; and (iv) there is still open development gaps in network security, as most samples did not have any contacted domain sinkholed at any time. We expect that these findings might foster future research in the field and help guiding the development of the next-generation malware detection solutions and policies.

Acknowledgements. This project was partially funded by the Brazilian National Counsel of Technological and Scientific Development (CNPq, PhD Scholarship, process 164745/2017-3).

References

Amazon (202). How do i report abuse of aws resources? https://aws.amazon.com/pt/premiumsupport/knowledge-center/report-aws-abuse/.

Aresu, M., Ariu, D., Ahmadi, M., Maiorca, D., and Giacinto, G. (2015). Clustering android malware families by http traffic. In *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, pages 128–135.

Beppler, T., Botacin, M., Ceschin, F. J. O., Oliveira, L. E. S., and Grégio, A. (2019). L(a)ying in (test)bed. In Lin, Z., Papamanthou, C., and Polychronakis, M., editors, *Information Security*, pages 381–401, Cham. Springer International Publishing.

Botacin, M., Ceschin, F., de Geus, P., and Grégio, A. (2020). We need to talk about antiviruses: challenges & pitfalls of av evaluations. *Computers & Security*, 95:101859.

Botacin, M., Grégio, A., and de Geus, P. (2015). Uma visão geraldomalwareativo no espaço nacional da internet entre 2012 e 2015. https://www.lasca.ic.unicamp.br/paulo/papers/2015-SBSeg-WFC-marcus.botacin-visao.malware.nacional.pdf.

Botacin, M., Kalysch, A., and Grégio, A. (2019). The internet banking [in]security spiral: Past, present, and future of online banking protection mechanisms based on a brazilian case study. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ARES '19, pages 49:1–49:10, New York, NY, USA. ACM.

Botacin, M. F., de Geus, P. L., and Grégio, A. R. A. (2018). The other guys: automated analysis of marginalized malware. *Journal of Comp. Virology and Hacking Techniques*.

Ceschin, F., Pinage, F., Castilho, M., Menotti, D., Oliveira, L. S., and Gregio, A. (2018). The need for speed: An analysis of brazilian malware classifers. *IEEE Sec. & Priv.*

CloudFlare (2020). What is dns filtering? | secure dns servers. https://www.cloudflare.com/learning/access-management/what-is-dns-filtering/.

Google (2020). Report suspected abuse on google cloud platform. https://support.google.com/code/contact/cloud_platform_report.

Hao, S., Feamster, N., and Pandrangi, R. (2011). Monitoring the initial dns behavior of malicious domains. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, New York, NY, USA. ACM.

Hunt, R. and Zeadally, S. (2012). Network forensics: An analysis of techniques, tools, and trends. *Computer*, 45(12):36–43.

Khatri, V. and Abendroth, J. (2015). Mobile guard demo: Network based malware detection. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 1177–1179.

Liu, D., Li, Z., Du, K., Wang, H., Liu, B., and Duan, H. (2017). Don't let one rotten apple spoil the whole barrel: Towards automated detection of shadowed domains. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 537–552, New York, NY, USA. ACM.

Locaweb (2019). Blacklist / lista negra. `https://ajuda.locaweb.com.br/wiki/blacklist-lista-negra/`.

Malshare (2018). Malshare. `http://malshare.com`.

McGrath, D. K., Kalafut, A., and Gupta, M. (2009). Phishing infrastructure fluxes all the way. *IEEE Security Privacy*, 7(5):21–28.

Parekh, J. J., Wang, K., and Stolfo, S. J. (2006). Privacy-preserving payload-based correlation for accurate malicious traffic detection. In *Proceedings of the 2006 SIGCOMM Workshop on Large-scale Attack Defense*, LSAD '06, pages 99–106, New York, NY, USA. ACM.

Rossow, C., Dietrich, C., and Bos, H. (2013). *Large-Scale Analysis of Malware Downloaders*, pages 42–61. Springer Berlin Heidelberg, Berlin, Heidelberg.

Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C., and Vigna, G. (2009). Your botnet is my botnet: Analysis of a botnet takeover. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 635–647, New York, NY, USA. ACM.

Sucuri (2019). Website blacklist removal. `https://sucuri.net/website-security-platform/blacklist-removal-and-repair/`.

Wang, T., Hu, X., Jang, J., Ji, S., Stoecklin, M., and Taylor, T. (2016). Botmeter: Charting dga-botnet landscapes in large networks. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pages 334–343.

Yousaf, S., Iqbal, U., Farooqi, S., Ahmad, R., Shafiq, Z., and Zaffar, F. (2016). Malware slums: Measurement and analysis of malware on traffic exchanges. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 572–582.

Yu, K. F. and Harang, R. E. (2017). Machine learning in malware traffic classifications. In *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, pages 6–10.