TRAWL: Protection against rogue sites for the masses

Antonia Nisioti, Mohammad Heydari, Alexios Mylonas, Vasilios Katos
Department of Computing and Informatics
Bournemouth University, Bournemouth, UK
{anisioti, mheydari, amylonas, vkatos}@bournemouth.ac.uk

Vahid Heydari Fami Tafreshi
Faculty of Computing, Engineering and Sciences
Staffordshire University, Stoke, UK
v.heydari@staffs.ac.uk

Abstract— ***The number of smartphones reached 3.4 billion in the third quarter of 2016 [1]. These devices facilitate our daily lives and have become the primary way of accessing the web. Although all desktop browsers filter rogue websites, their mobile counterparts often do not filter them at all, exposing their users to websites serving malware or hosting phishing attacks. In this paper we revisit the anti-phishing filtering mechanism which is offered in the most popular web browsers of Android, iOS and Windows Phone. Our results show that mobile users are still unprotected against phishing attacks, as most of the browsers are unable to filter phishing URLs. Thus, we implement and evaluate TRAWL (TRAnsparent Web protection for alL), as a cost effective security control that provides DNS and URL filtering using several blacklists.***

Keywords— ***Phishing, Mobile web browsers, Android, iOS, Windows Phone, Smartphone, Security***

I. Introduction

Nowadays, the use of mobile devices has become pervasive and smartphones are now the primary way of accessing the web [2]. Smartphones contain a plethora of valuable data, a fact that constitutes motivation for an attacker who seeks to violate their confidentiality, integrity or availability. According to McAfee [3], in 2016 we witnessed an increase in mobile malware that is predicted to continue in 2017, as well. In the same year, Kaspersky detected almost 3 million malware installations on mobile devices [4]. The growing number of smartphone targeting malware is inevitable due to the lack of security mechanisms [5]. For the past year, the main mobile threats were Trojans and banking malware, which were distributed through malicious sites delivering advertisements [6].

Consequently, accessing the web presents, more than ever, the risk of downloading malicious software. A user could accidently download malware, which is commonly known as “drive by download”, just by visiting a web page with active content. According to authors in [7], 1.3% of Google’s search engine queries return at least one link to a malicious site. These web pages can be present not only in ‘nefarious’ websites (*e.g.*, gambling sites, pirated software, adult sites), but also in ‘benign’ sites, such as social media, search engines, news sites, which may have been compromised [8]. Web-based malware attacks are on the rise due to the increasing numbers of online applications leveraging the use of a browser [9]. Moreover, in the third quarter of 2016, it was reported that Internet browsers are the most targeted application by cybercriminals [10].

In addition, web users are exposed to attacks which aim to deceive them with the ultimate goal of stealing their personal and/or financial information, as well as their login details. This type of attack is commonly known as phishing and may lead to even more serious crimes, such as identity theft, fraud, hacking, etc. Phishing is typically initiated through email, SMS, web articles or social media, by including a link to a compromised or rogue website. A successful phishing attack consists of three main phases [11]: 1) the attacker lures a victim to a malicious webserver through an email, SMS, advertisement, etc., 2) the victim follows the rogue URL; 3) the attacker successfully captures the user’s data. In most cases, a successful phishing attack involves both social engineering and technological methods [12].

To combat the aforementioned threat of rogue web sites, *i.e.*, those serving malware or hosting phishing attacks, the browser provides warnings to the user as to whether access to the requested webpage constitutes a threat. Also, security indicators are provided by the browser’s graphical interface. For instance, a padlock or link to SSL certificate is available next to the address bar for the user to inspect whether a given website or webpage can be trusted [13]. However, users often ignore these security indicators, as they may 1) lack the knowledge regarding phishing related threats; 2) be social engineered by a webpage or graphics; or 3) not pay attention to the browser’s security indicators [14]. Moreover, these security indicators are available for desktop browsers, but not for their mobile counterparts [5].

One anti-phishing security measure is DNS-filtering. Both Google and OpenDNS offer free public DNS services that block rogue web sites. Consequently, a user does not need to have her own anti-rogue site detection system. However, the user has to sacrifice her privacy for this added security, as all the traffic is redirected through those companies’ DNS servers. Moreover, DNS filtering is able to block malicious URLs only on the domain level. For instance, *www.evilsite.com* will be blocked assuming that this is a known rogue site, while *www.legitsite.com/photo.php?code=evilcode* will not, assuming that the domain www.legitsite.com is a 'benign' domain that is compromised.

A security countermeasure that overcomes the previously mentioned obstacle of DNS filtering, is the use of URL filtering. URL filtering facilitates blacklists, which either come with third–party software of the web browser (*i.e.*, security extensions) [15] or are pre-installed, such as Google's Safe browsing. In addition, browser extensions, such as AdBlock [16] and Ghostery [17] that block advertisements (both malicious and benign), nowadays also include blacklists for malicious content. Google provides Safe Browsing [18], a blacklist which offers protection against both malware and phishing. Currently, Mozilla Firefox, Google Chrome and Apple Safari implement Safe Browsing. Internet Explorer uses Microsoft's blacklist called SmartScreen [19]. Finally, Opera uses a combination of PhishTank[20] and Netcraft[21] blacklists against phishing and TRUSTe [22] against malware. However, our past work proved that these technologies are absent from web browsers in Android and iOS [23], [25], [26].

In this paper, we revisit the anti-phishing protection that is provided by mobile browsers on the three most popular OS for mobile devices, namely: Android, iOS and Windows Phone. We also compare this protection to our previous findings. Our results show that mobile users are still vulnerable against phishing. To raise the bar of the anti-phishing protection that is currently offered, we implement TRAWL (TRAnsparent Web protection for alL) as an instance of Secure Proxy [25], offering a cost-efficient security control against rogue sites. In summary, the paper makes the following contributions:

- Compares the anti-phishing protection that is provided by mobile web browsers in Android, iOS and Windows Phone. This protection is compared to our previous findings regarding anti-phishing protection in Android and iOS, highlighting the evolution of the security mechanisms in mobile devices.
- It implements and evaluates TRAWL as a cost effective security control against rogue sites. TRAWL is based on our previous work, Secure Proxy that aggregates multiple blacklists to provide web users with protection against rogue sites.

The remainder of the paper is structured as follows. Section 2 presents related work. Section 3 describes the methodology and Section 4 presents our results. Section 5, introduces TRAWL. Finally, Section 6 provides a discussion of our work and suggestions for future work.

II. Related Work

Mylonas et al. [23] compared the availability and manageability of security controls offered by both mobile and desktop web browsers. The results highlighted that web browsers for mobile devices lack the necessary security controls, which are typically found in desktops. According to their results, Safari and Opera Mini had several serious security issues such as unpatched vulnerabilities and no protection against invalid digital certificates. The protection provided by Firefox Mobile was comparable to the protection offered by desktop browsers. The evaluation also revealed that web users are exposed to third-party advertising due to the out of the box protection that is offered by most desktop browsers. Wu et al. [24] analysed 115 android mobile browser applications in terms of protection against four types of attacks on Android devices. Based on their results, more than half of the tested browsers were found to be vulnerable.

Virvilis et al. in [25] and [26] examined the anti-phishing and anti-malware protection that is offered by web browsers on the two most popular smartphone operating systems, iOS and Android, as well as Windows for desktops. The most popular web browsers for desktops and mobile devices were tested against 2800 rogue URLs (1400 phishing and 1400 malicious URLs). The tests revealed that the level of protection offered by desktop browsers is higher than the one in smartphones, even when these browsers implement the same technology (*e.g.*, Safe Browsing). Specifically, iOS's default browser offers no protection against malicious URLs and limited protection against phishing. Likewise, the default Android browser offers no protection against rogue web sites. To address these problems the authors proposed the use of a secure proxy for analysing URLs with the help of several blacklists and AV engines. Their proposed system significantly raised the level of protection against rogue sites for both mobile and desktop web browsers.

Amrutkar et al. [27] present a comparison of the security indicators available on mobile web browsers and the ones available on their traditional desktop version. Their work indicates that mobile browsers fail to implement all the recommended desktop indicators and show inconsistency regarding their availability across the different applications.

Akhawe et al. [28] utilized both Mozilla Firefox and Google Chrome to collect data regarding effectiveness of warning impressions in security events. The authors showed that security warnings can have an immense impact on user behaviour depending on the demographic group of the user.

Furthermore, the authors in [29], used statistical data to compare a website against known malicious sites in order to identify similarities. Antonakakis et al. [30] developed Kopis, a system which operates at the upper DNS level and attempts to detect malware related domains based on global DNS resolution patterns. In [31], authors propose an architecture that heuristically selects candidate URLs and determines if they exhibit malicious behavior via execution in a virtual machine. Finally, the authors in [32] propose Zozzle, a static in-browser detector for malicious JavaScript.

Table I. Applications' versions used for evaluation

Application	Android	iOS	Windows Phone
Firefox	50.0	5.3	N\A
Opera	37.0	14.0	9.1.0
Chrome	54.0	54.0	N\A
Default browser	2.1.34	N\A	N\A
Internet Explorer	N\A	N\A	11.0
Safari	N\A	version in iOS 10.1.1	N\A

III. Methodology

A. *Testing Enviroment*

Virvilis et al. [25] and Mylonas et al. [23] evaluated the build-in protection mechanisms offered by various web browsers against rogue websites. Their results highlighted that mobile users are exposed to this threat. In this work, we revisit their experiments on mobile devices to evaluate the anti-phishing protection of mobile browsers. We evaluate the popular browsers that are available in the three most popular operating systems for mobile devices, namely Android, iOS and Windows Phone [33] (see Table I).

Similarly to Virvilis et al. [25] and Mylonas et al. [23], to evaluate the protection of the mobile browsers against malicious websites a webpage containing 100 links to malicious websites was created on a local webserver. The devices that were used for the evaluation were running the most popular version of Android, iOS and Windows Phone, at the time that our experiments were conducted (October 2016 – December 2016). Specifically, we used: a) a Samsung Note 3 with Android Lollipop for testing the Android browsers, as it is the dominant Android version with 35% of the user share [34], b) an iPhone 4s with iOS 10.1.1, and c), a Microsoft Lumia 540 with Windows 8.1. Moreover, we tested the most popular browsers that are available in these operating systems, namely Chrome, Firefox, Internet Explorer, Opera, Safari and the default browser in Android (referred as Internet or Browser in Android). The availability of the web browsers in Android, iOS and Windows Phone is summarized in Table I.

B. *Mobile browser test*

To evaluate the anti-phishing protection offered by mobile browsers we randomly selected 100 confirmed online phishing URLs that were reported in PhishTank the day before our anti-phishing experiments. These URLs were added as links in a web page that was hosted on a local web server. Then, with each mobile web browser we attempted to visit each of these URLs and classified them in one of the following categories:

i. **Blacklisted**: the URL is successfully detected as a phishing attack by the browser and a warning is displayed to the user.

ii. **False negative**: a phishing URL that is not blocked by the browser and therefore exposes the user to a phishing attack.

iii. **Non-phishing**: a URL that is not blocked by the browser but has been suspended/taken down and therefore does not expose the user to a phishing attack anymore.

IV. Experimental Results

This section presents the experimental results regarding the anti-phishing protection of mobile browser in Android, iOS and Windows Phone. It also provides a comparison of with our previous findings in [23], [25], [26].

Android. Our results suggest that despite the improvements with regards to the anti-phishing protection that is offered on some of the mobile browsers (namely Firefox and Opera), Android users still remain exposed to this threat (see Table II). This holds true, as the default, pre-installed Android browser and Chrome provide by default no anti-phishing protection. Specifically, as shown in Table II the browsers failed to block any of the phishing sites. On the other hand, Firefox and Opera blocked most of the attacks.

iOS. As illustrated in Table III, during our experiments none of the browsers was able to block any phishing attack. This is surprising especially in the case of Firefox and Safari, as both of their desktop counterparts use Safe Browsing.. Moreover, Firefox for Android blocked 80% of the phishing attacks. Consequently, this suggests that Firefox for iOS did not implement or does not have enabled by default Safe Browsing in version 5.3.

Windows Phone. As discussed earlier, at the time of our experiments Chrome and Firefox were not available in Windows Store for Windows Phone 8.1. The results suggest that Windows Phone users are also exposed to phishing attacks (see Table 4). This holds true as IE only blocked half of the phishing sites, while Opera did not block any of them.

A. *Comparison with previous evaluation*

Our results confirm the results in [23] and [25]. Most of the findings from the experiments that were conducted in 2014 are still valid, *i.e.*, web browsers for mobile devices still fail to

Table II. Results from Samsung Note3 running Android Lollipop (n=100)

Browser	False Negative	Blacklisted	Non-phishing
Default Browser (v2.1.34)	57	0	43
Firefox (v50.0)	16	80	4
Opera (37.0)	14	75	11
Chrome (54.0)	52	0	48

Table III: Results from iPhone 4s running iOS version 10.1.1 (n=100)

Browser	False Negative	Blacklisted	Non-phishing
Safari	50	0	50
Firefox (5.3)	56	0	44
Opera (14.0)	62	0	38
Chrome (54.0)	62	0	38

Table IV. Results from Microsoft Lumia 640 running Windows Phone 8.1 (n=100)

Browser	False negative	Blacklisted	Non-phishing
IE (11)	28	52	20
Firefox	No app available for Windows		
Opera (9.1.0)	61	0	39
Chrome	No app available for Windows		

protect their users from phishing sites. Specifically, only Firefox and Opera for Android offer improved anti-phishing protection. However, they are third-party web browsers that are not necessarily used by the owner of the mobile device. The only pre-installed browser that partially protects users against phishing is Internet Explorer 11 for Windows Phone 8.1. Neither Safari for iOS, nor the default pre-installed browser for Android offers anti-phishing protection. Consequently, users have to install third party browsers in order to have a filtering mechanism against phishing sites – as well as malicious sites as the absence of the filtering mechanism exposes the users to both malware and phishing. An Android user can use a third-party browser, such as Firefox or Opera - assuming that he has the knowledge to do so. However, this does not hold for iOS users as our results suggest that still neither the default browser (*i.e.,* Safari) nor any third-party browser offers any protection against phishing attacks.

V. TRAWL

This work uncovers that most mobile browsers do not filter rogue websites. Therefore, as discussed in the previous subsection, their users are exposed to websites hosting phishing attacks and serving malware. In this context, we implement TRAWL (TRAnsparent Web protection for alL) as an extension of Secure Proxy, a security countermeasure that was proposed by Virvilis et al. [25]. This section provides the architecture and details about its implementation and evaluation.

A. Architecture

Secure proxy [25] was design as a cross platform security control, which raises the bar of protection against rogue web sites. To this end, it queries VirusTotal in order to deduce if a given http request should be blocked or not. However, this introduces two considerable limitations: a) the online queries to VirusTotal leak the users' browsing history to Google servers, thus violating their privacy and b) VirusTotal has imposed a restriction of four queries per minute by its public API, making secure proxy's deployment impractical. To overcome these limitations, TRAWL uses multiple local blacklists from various sources instead of online queries. TRAWL acts as a DNS filtering server and a web proxy to blocks both rogue domains and URLs, respectively, by utilizing multiple local blacklists (see Figure 1).

TRAWL's architecture is depicted in Figure 1 and its operational flow in Figure 2. Once a client is configured to use TRAWL to browse the web, each http request is firstly redirected to the DNS filtering server. The server checks whether the domain of the queried URL exists on its own blacklist. If the domain is blacklisted, then the request is rejected and a message is displayed to the user to inform her that the requested website is considered a threat and should be blocked. In case the URL is not blacklisted by the DNS filtering server, the request is examined by the proxy server. The proxy server will check the requested URL against its blacklists to determine whether it should be blocked or not. If the URL is found in the blacklist the request is rejected. Otherwise, the request is forwarded.

B. Hardware and Software considerations

To implement the module for URL blocking a proxy server was selected as: (a) it works not only at the domain level but also at URL level, (b) it can speed up the user's browsing experience as it has a good caching mechanism that reduces network latency [35] and (c) it has the ability of masking the client and therefore improving her anonymity.

Although, the proxy server is sufficient for filtering malicious websites, DNS filtering is a faster solution. Its disadvantage is that it cannot detect malicious URLs and therefore is limited to domain filtering. However, by combing the two approaches we can exploit both their advantages.

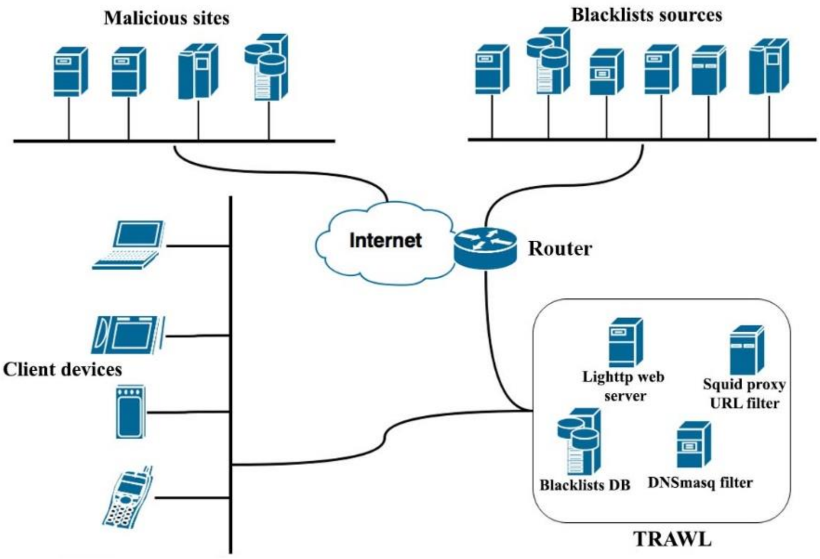

Fig. 1. Architecture of TRAWL

We envisage TRAWL as a security control that is free and available for everyone. To this end, we used open-source software and low cost hardware in order to provide an affordable solution. TRAWL relies on bash and python scripting and therefore any Unix/Linux operating system compatible with the selected hardware can be used as the device running TRAWL. Squid [36] was chosen as the proxy server as it is a well-known, stable and multi-functional software. DNSmasq [37] was chosen as the DNS server because of its ability to work with low cost systems. In terms of hardware, Raspberry Pi 3 was chosen because of its low operational consumption and low price.

C. Black List Sources

Using a set of up-to-date lists of rogue websites is essential for creating a comprehensive blacklist. In this regard, we have aggregated a list of 25 different sources that are used by TRAWL to filter rogue websites, namely:
{Abuse.ch, Amazon, Anti-Adblock Killer, AutoShun, Blueliv, Camelon, Comodo Site Inspector, DNS-BH – Malware Domain Blocklist, ,Easylist, Fanboy, hpHosts, Malc0de, Malekal, Malware Domain List (MDL), Malware patrol, Malwared, Open BL organization, OpenPhish, PhishTank, Ransomware tracker, Safe browsing, Secure Mecca, Spam404, Streans, Zeus tracker}

The aforementioned list provides protection against a) malicious sites, b) phishing sites, and c) advertisements.

D. Power consumption test

As stated earlier, one of the goals of TRAWL is to offer to the user a cost-efficient solution against rogue websites. For this reason, we have conducted three experiments to measure the power consumption of the proposed solution.

During the first experiment the Raspberry Pi was idle with no client connected to it. Therefore, TRAWL was active but no client device was trying to access any domain. For the second experiment we assumed that TRAWL is used in a home network in which 5 clients were using it, namely: 3 iPhones, 1 Samsung Note 3, and a Mac laptop. Users of the connected devices were utilizing different browsers to access various websites on the Internet. Finally, throughout the last experiment the five devices were still connected and in use and TRAWL was downloading and updating its blacklists at the same time. Each of the experiments was monitored over a three-hour period in order to get a consistent record.

To conduct the experiments an energy power meter [38] was used, which essentially acts as a bridge between the mains and the Raspberry Pi’s power adaptor. The kit has its own battery to operate and measure the electric consumption of TRAWL.

E. Experimental Results

Figure 3 shows the increase in power consumption when clients are connected to TRAWL as well as when its blacklist databases were updating. The device consumes 2.2 Watts when it is in idle mode and an average of 2.7 Watts when it is in use by five clients. However, when the blacklists are updated and TRAWL used by five clients, the power consumption almost doubles (3.6 Watts). This is expected as the greater the computational requirements, the greater the amount of power required.

Figure 4 presents the accumulated energy consumption of the Raspberry Pi during the tests. Over the course of three hours for each test, there was an average increase of 0.003 kWh every hour. Getting an average reading, the idle mode provokes the least energy consumption at 0.002/kWh; when the proxy is in use, the average reading grows to 2.67 Watts per hour. Finally, during the last test the reading further increases to an average of 4.67/Wh.

To evaluate how cost-efficient and environment-friendly is the TRAWL, the cost of electric consumption and the carbon footprint in the United Kingdom is computed. For this we considered the cost for a pay-as-you-go tariff (£0.14 per hour) [39] and the yearly carbon footprint based on the online computation provided by the National Energy Foundation [40]. Based on the aforementioned, Table 5 illustrates that if the proxy server is running in a household or an SME, it will cost a maximum of £6 per year (based on £0.14p/hour electric tariff) with a carbon footprint of 3 kgCO2.

VI. Conclusion and Future Work

Nowadays, smartphones have become a vital part of our everyday life. The number of malware targeting mobile devices is growing day by day and the majority is being distributed through phishing attacks. Therefore, this paper evaluates the anti-phishing and anti-malware security mechanisms offered by mobile web browsers. We consider the most popular web browsers for the three most popular mobile operating systems, namely Android, iOS and Windows Phone.

Our results reveal that the majority of mobile browsers still do not offer the necessary protection against rogue websites. Specifically, our experimental results suggest that the pre-installed browsers offer no protection against phishing in iOS and Android and inadequate protection (50%) in Windows Phone. Firefox and Chrome on Android are the only browsers offering an adequate protection against phishing (approximately 75-80%). It is worth noting that our results are not generalizable, as we used a limited number of rogue websites to test the protection that is offered by web browsers. However, our results highlight the absence of the security control in a number of web browsers for mobile devices, which leaves their users exposed to websites serving malware and phishing attacks.

In this context, we present TRAWL, an extension of our secure proxy that as we demonstrate is an energy efficient security control against rogue web sites. TRAWL offers DNS and URL filtering based on multiple blacklists. We decided not to implement our proposed countermeasure on client side (i.e. on the mobile phone, tablet, desktop etc.) for two reasons: a) not to consume the client's resources (battery, processing power etc.); b) to provide a cross-platform solutions, independent from the device's OS. One of our goals was to create a cost-efficient domestic security control and therefore we chose Raspberry Pi as the hardware that hosts TRAWL. Finally, we designed TRAWL having domestic or SME usage in mind. Therefore, we have not considered any performance or scalability requirements that might arise in a more complex corporate environment, which we consider as out of our scope. Our results prove that TRAWL is efficient in terms of power consumption.

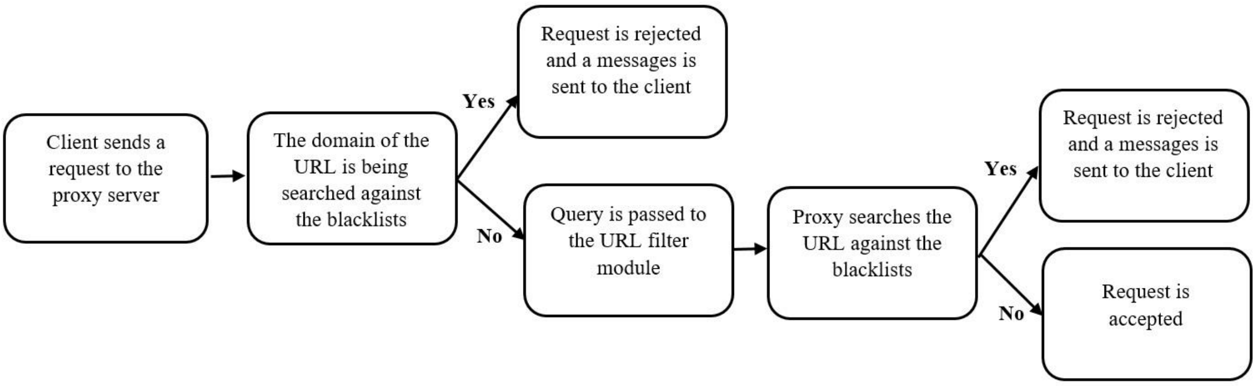
Client sends a request to the proxy server
The domain of the URL is being searched against the blacklists
Yes
Request is rejected and a messages is sent to the client
No
Query is passed to the URL filter module
Proxy searches the URL against the blacklists
Yes
Request is rejected and a messages is sent to the client
No
Request is accepted

Fig. 2. Operational process of TRAWL

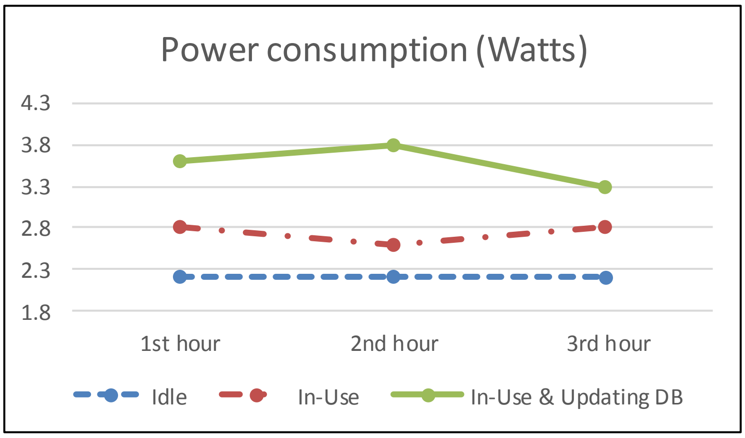

Fig. 3. Power consumption test for TRAWL that is idle, normal and heavy load

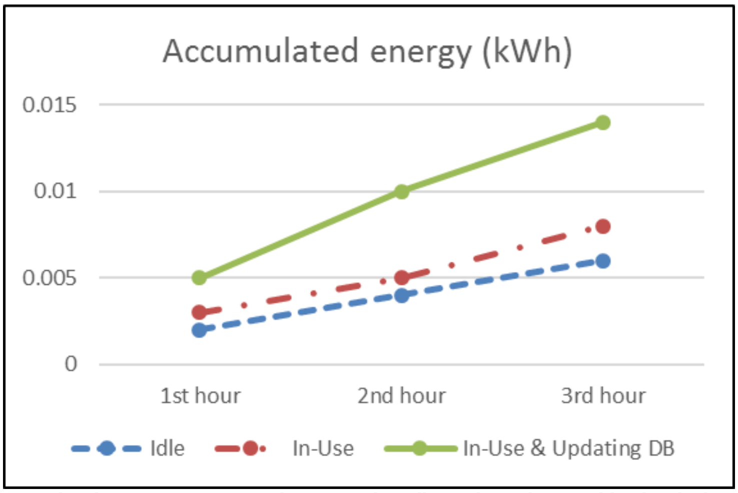

Fig. 4. Accumulated energy test over a 3-hour period in idle mode, with normal load and when it is busy

Manual evaluation is a time-consuming process which requires a significant amount of effort and so our corpus was limited to 100 rogue URLs only. Furthermore, browsers' frequent updates require constant re-evaluation of the security mechanisms in use. Finally, new rogue webpages are created every minute, as there are many tools online that can generate or clone a legitimate website in a few minutes. TRAWL utilizes several blacklists for detecting malicious URLs and therefore is based on static knowledge. Such a system is inseparably linked with its databases and its performance depends on their quality. Consequently, though filtering of rogue URLs is a well-used method as it produces a high rate of true positive as its efficiency depends on the blacklists in use.

For future work we plan to extend TRAWL to include Google's Safe Browsing. Moreover to overcome the manual evaluation obstacle we intent to create an architecture for automated evaluation of browsers on different platforms (mobile and desktop).

ACKNOWLEDGMENT

The authors would like to express their appreciation to R. Delmo for his contribution.

REFERENCES

[1] Ericsson Mobility Report, June 2016

[2] Rob van der Meulen, Christy Pettey, "Gartner Survey Highlights Top Five Daily Activities on Media Tablets," 2012. [Online]. Available: https://www.gartner.com/newsroom/id/2070515. [Accessed 23 01 2017

[3] M. L. Report, "McAfee Labs 2017 Threats Predictions," MacAfee Labs, 2016.

[4] Kaspersky, "IT threat evolution Q3 2016. Statistics," Kaspersky, [Online]. Available: https://securelist.com/analysis/quarterly-malware-reports/76513/it-threat-evolution-q3-2016-statistics/. [Accessed 25 1 2017].

[5] La Polla, M., Martinelli, F., & Sgandurra, D. (2013). A survey on security for mobile devices. *IEEE communications surveys & tutorials*, *15*(1), 446-471.

[6] David Emm, Roman Unuchek, Kirill Kruglov, "Kaspersky Security Bulletin 2016, REVIEW OF THE YEAR," Kaspersky Lab, 2016.

[7] Mavrommatis, N. P. P. and Monrose, M. A. R. F.,, "All your iframes point to us," in *USENIX security symposium*, 2008.

[8] CISCO, Cisco annual security report. [Online]. Available at: http://www.cisco.com/c/en/us/products/security/annual_security_report.html [Accessed 27.01.2017]

[9] Perdisci, R., Ariu, D., & Giacinto, G. (2013). Scalable fine-grained behavioral clustering of http-based malware. *Computer Networks*, *57*(2), 487-500.

[10] Unuchek, R., Garnaeva, M., Makrushin, D., Sinitsyn, F. and Liskin, "IT threat evolution Q3 2016, Kaspersky Lab," 2016.

[11] J. Hong, "The state of phishing attacks," *Communications of the ACM*, vol. 55, 2012.

[12] Chaudhry, J. A., Chaudhry, S. A. and Rittenhouse, "Phishing Attacks and Defenses," *International Journal of Security and Its Applications*, vol. 10, 2016.

[13] Akhawe, D. and Felt, A. P., "Alice in warningland: A large-scale field study of browser security warning effectiveness," in *22nd USENIX Security Symposium (USENIX Security 13)*, 2013.

[14] Dhamija, R., Tygar, J. D. and Hearst, M.,, "Why phishing works," in *Proceedings of the SIGCHI*

conference on Human Factors in computing systems, ACM, 2006.

[15] Kirda, E. and Kruegel, C., "Protecting users against phishing attacks with antiphish," in *IEEE 29th Annual International Computer Software and Applications Conference*, 2005.

[16] AdBlock, "AdBlock, 2016," [Online]. Available: https://getadblock.com/ [Accessed 27.01.2017]

[17] "Ghostery," [Online]. Available: https://www.ghostery.com/about-us/about-ghostery. [Accessed 25.1.2017]

[18] Google, "Safe Browsing API". [Online]. Available at: https://developers.google.com/safe-browsing/ [Accessed 27.01.2017]

[19] Microsoft, "SmartScreen Filter". [Online]. Available at: https://blogs.msdn.microsoft.com/b8/2011/09/15/protecting-you-from-malware/ [Accessed 27.01.2017]

[20] PhishTank, "Join the fight against phishing". [Online]. Available at: https://www.phishtank.com/ [Accessed 27.01.2017]

[21] Netcraft,"Anti-phishing Services". [Online]. Available at: https://www.netcraft.com/anti-phising/ [Accessed 27.01.2017]

[22] Abrams, R., O. Barrera, and J. Pathak. "Browser Security Comparative Analysis." *NSS Labs* (2013).

[23] Alexios Mylonas, Nikolaos Tsalis, Dimitris Gritzalis, "Evaluating the manageability of web browsers controls," *Springer Security and Trust Management*, 2013.

[24] Wu, D., & Chang, R. K. (2014, October). Analyzing Android browser apps for file://vulnerabilities. In *International Conference on Information Security* (pp. 345-363). Springer International Publishing.

[25] Virvilis, N., Mylonas, A., Tsalis, N. and Gritzalis, D, "Security Busters: Web Browser security vs. rogue sites," *Computers & Security*, vol. 52, 2015.

[26] Virvilis, N., Tsalis, N., Mylonas, A., & Gritzalis, D. (2014, August). Mobile devices: A phisher's paradise. In *Security and Cryptography (SECRYPT), 2014 11th International Conference on* (pp. 1-9). IEEE.

[27] Amrutkar, C., Traynor, P., & Van Oorschot, P. C. (2015). An empirical evaluation of security indicators in mobile Web browsers. *IEEE Transactions on Mobile Computing*, *14*(5), 889-903.

[28] Devdatta Akhawe, Adrienne Porter Felt, "Alice inWarningland: A Large-Scale Field Study of Browser SecurityWarning Effectiveness," in *22nd USENIX Security Symposium*, 2013.

[29] Vadrevu, P., Rahbarinia, B., Perdisci, R., Li, K. and Antonakakis, M, "Measuring and detecting malware downloads in live network traffic," in *European Symposium on Research in Computer Security*, 2013.

[30] Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou II, N. and Dagon, D., "Detecting Malware Domains at the Upper DNS Hierarchy," in *USENIX security symposium*, 2011.

[31] Niels Provos, Dean McNamee, "The ghost in the browser analysis of web-based malware," in *Proceeding HotBots'07 Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, 2007.

[32] Curtsinger, C., Livshits, B., Zorn, B. G. and Seifert, C, "ZOZZLE: Fast and Precise In-Browser JavaScript Malware Detection," in *USENIX Security Symposium*, 2011.

[33] International Data Corporation (IDC) Smartphone OS Market Share, 2016 Q3. (Online). Available at: http://www.idc.com/promo/smartphone-market-share/os;jsessionid=0335F30E5F14899C5B1D8DB3EB09D02D [Accessed 27.01.2017]

[34] Statista, 2016a. *Android version market share distribution among smartphone owners as of September 2016* [online]. Statista. Available from: https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/ [Accessed 15 October 2016].

[35] Barish, G., & Obraczke, K. (2000). World wide web caching: Trends and techniques. *IEEE Communications magazine*, *38*(5), 178-184.

[36] Squid, 2016. About Squid Proxy [online]. Squid-Cache.org. Available at http://www.squid-cache.org/Intro

[37] Kelly, S., 2016. Dnsmasq [online]. Available from: http://www.thekelleys.org.uk/dnsmasq/doc.html

[38] Energenie, 2016. *Energy Saving Power Meter* [online]. Energenie. Available from: https://energenie4u.co.uk/catalogue/product/ENER007 [Accessed 1 December 2016].

[39] ScottishPower, 2016. *Tariff Information Label Lookup* [online]. Scottish Power. Available from: https://www.scottishpower.co.uk/tariff-information.process?execution=e1s2 [Accessed 2 December 2016].

[40] NEF, 2016. *Simple Carbon Calculator* [online]. National Energy Foundation. Available from: http://www.carbon-calculator.org.uk/ [Accessed 2 December 2016].