Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Security Busters: Web browser security vs. rogue sites

Nikos Virvilis [a], Alexios Mylonas [a,b], Nikolaos Tsalis [a], Dimitris Gritzalis [a,*]

[a] Information Security & Critical Infrastructure Protection (INFOSEC) Laboratory, Dept. of Informatics, Athens University of Economics & Business, 76 Patission Ave., Athens GR-10434, Greece
[b] Faculty of Computing, Engineering and Sciences, Staffordshire University, Beaconside, Stafford ST18 0AD, United Kingdom

ARTICLE INFO

Article history:
Received 27 January 2015
Received in revised form
14 April 2015
Accepted 20 April 2015
Available online 29 April 2015

Keywords:
Android
iOS
Malware
Mobile device
Phishing
Smartphone
Security
Windows
Web browser

ABSTRACT

URL blacklists are used by the majority of modern web browsers as a means to protect users from rogue web sites, i.e. those serving malware and/or hosting phishing scams. There is a plethora of URL blacklists/reputation services, out of which Google's Safe Browsing and Microsoft's SmartScreen stand out as the two most commonly used ones. Frequently, such lists are the only safeguard web browsers implement against such threats. In this paper, we examine the level of protection that is offered by popular web browsers on iOS, Android and desktop (Windows) platforms, against a large set of phishing and malicious URL. The results reveal that most browsers – especially those for mobile devices – offer limited protection against such threats. As a result, we propose and evaluate a countermeasure, which can be used to significantly improve the level of protection offered to the users, regardless of the web browser or platform they are using.

1. Introduction

Nowadays, people spend more than 60 h per week accessing online content, either through their mobile devices or personal computers (Nielsen, 2014). The use of web applications and services is continuously increasing, with smartphones now being the primary device used to access the web (Gartner, 2014). However, while browsing the web users might visit *rogue web sites*, namely sites that serve malicious software (malware) and/or host phishing scams. It is worth clarifying, that users might be exposed to such threats not only when they visit nefarious web sites, such as adult websites, ones hosting pirated software or gambling sites. Benign sites (e.g. social media websites, search engines, news sites, etc.) have been misused in the past to deliver phishing/malware attacks, after being compromised (i.e., watering hole attacks) (CISCO, 2014). As a result, the likelihood that users will be exposed

* *Corresponding author.*
E-mail addresses: nvir@aueb.gr (N. Virvilis), alexios.mylonas@staffs.ac.uk, amylonas@aueb.gr (A. Mylonas), ntsalis@aueb.gr (N. Tsalis), dgrit@aueb.gr (D. Gritzalis).

http://dx.doi.org/10.1016/j.cose.2015.04.009
0167-4048/© 2015 Elsevier Ltd. All rights reserved.

to such threats, should not be neglected. In fact, Symantec states that 38% of mobile users have experienced mobile cybercrime in past 12 months (Symantec).

Phishing is one of the most popular and profitable attacks, as almost 450,000 attacks happened in 2013 with an estimated loss of over $5.9 billion (RSA, 2014). In addition, one in every 392 emails contained a phishing attack in 2013 (Symantec), and to make things worse, 80% of business users were unable to detect phishing attacks effectively (McAfee, 2014). If we take into account the increased exchange of emails on mobile devices (Gartner, 2070), users are very likely to access phishing pages on such a device.

Malware attacks are also increasing, with Kaspersky Labs reporting that over 3 billion malware attacks were detected in 2013, with a total of 1.8 million malicious or potentially unwanted programs used in these attacks (Funk and Garnaeva, 2013). McAfee Labs Threat Report highlighted that there was a 167 percent growth in mobile malware between 2013 and 2014 (McAfee, 2014).

Our previous work (Mylonas et al., 2013) uncovered the lack of security awareness of mobile users, which is confirmed by a recent report revealing that 57% percent of adults are unaware of the existence of security solutions for mobile devices (Symantec). These users rely on web browsers (or ‘browsers’) to protect them from web sites (or ‘sites’) that serve malware (hereinafter referred as ‘malicious sites’) or phishing attacks (hereinafter referred as ‘phishing sites’).

In this paper, we evaluate the level of protection against rogue web sites, which is offered by the most popular web browsers on the desktop and smartphone platforms. Our work focuses on Android and iOS for the smartphone platform and Windows for the desktop platform. Our results reveal that most smartphone browsers offer very limited (if any) protection against rogue sites. On the other hand, most desktop browsers offer an adequate level of protection against phishing sites, but not against malicious sites. As a consequence, in order to raise the level of protection that is offered to users, we propose and implement a new countermeasure that can be used by any web browser or platform.

The paper makes the following contributions:

- It provides a comparison of the phishing and malware protection offered by the popular desktop browsers (Windows) and mobile browsers (Android and iOS).
- It highlights the limited efficacy – and in specific cases the total lack – of protection mechanisms on mobile browsers.
- It compares the anti-phishing browsers' detection rate with our previous results (Virvilis et al., 2014).
- It proposes *Secure Proxy*, a new countermeasure that is based on the aggregation of multiple blacklists and AntiVirus (AV) engines. The countermeasure is implementated and evaluated, demonstrating that it offers enhanced protection against rogue sites.

The rest of the paper is structured as follows: Section 2 presents related work. Section 3 describes our methodology. Section 4 presents our experimental results. In Section 5 we present and evaluate *Secure Proxy*, our proposed countermeasure against rogue sites. The paper continues with a discussion of our work in Section 6 and presents our conclusions and future work in Section 7.

2. Background

2.1. *Phishing*

The main defense against phishing/malware attacks is based on blacklists, which are used by browsers to identify if a requested URL has been reported as malicious. The most popular blacklist is Google's Safe Browsing (Google, 2014), which protects from both phishing and malicious web sites. Safe Browsing is currently used by Chrome, Firefox and Apple Safari browsers. Internet Explorer uses SmartScreen, Microsoft's proprietary blacklist (Microsoft, 2014). Other browsers use their own proprietary lists and/or aggregate data from third parties. For instance, Opera uses phishing blacklists from Netcraft (Netcraft) and PhishTank (Phishtank) and a malware blacklist from TRUSTe (Abrams et al., 2014).

A number of approaches has been proposed by the research community to protect users from phishing attacks, which vary from user awareness surveys to experiments of the effectiveness of current security mechanisms and proposals of novel ones. More specifically, the work in Banu et al. (2013), Rosiello et al. (2007) and Rani and Dubey (2014) focuses on phishing with regards to its properties, characteristics, attack types, and available counter-measures. Authors in Rani and Dubey (2014), Jansson and Von Solms (2013) present a survey on user training methods, as well as their effectiveness against phishing attacks.

Literature has also focused on visual indicators that protect users from phishing. An overview of the warning indicators is presented in Bian (2014). Also, Darwish and Bataineh (2012) has surveyed users' interaction regarding security indicators in web browsers. A study on the effectiveness of browser security warnings was published in Akhawe and Felt (2013), focusing on Chrome and Firefox browsers. A similar study in Egelman and Schechter (2013) analyzed the impact on the users' decision based on the choice of background color in the warning and the text descriptions that were presented to them.

The authors in Sheng et al. (2009) focused on the effectiveness of phishing blacklists, and more specifically on their update speed and coverage. They found that less than 20% of phishing sites were detected at the beginning of their test. Similarly, in Kirda and Kruegel (2005) the authors proposed the use of ‘Anti-Phish’, an anti-phishing extension for Firefox. Zhang et al. (2011a) used a text classifier, an image classifier, and a fusion algorithm in order to defend against known properties of phishing attacks.

The authors in AV (2012) analyzed 300 phishing URLs and measured the detection effectiveness of desktop browsers. Opera browser offered the highest level of protection by blocking 94.2% of the phishing sites. Mazher et al. (2013) tested the effectiveness of anti-phishing add-ons for Internet Explorer, Chrome, and Firefox, finding that Chrome outscored the other browsers. Finally, Abrams et al. (2014) examined the time required for Firefox, Chrome, Opera, IE, and Safari to block a malicious site (from the creation of the malicious domain until the time it was blocked by the browsers).

The authors in Vidas et al. (2013) investigated the viability of QRishing (i.e. QR-code-initiated phishing attacks). Similarly, Xu and Zhu (2012) examined how notification customization may allow an installed Trojan application to launch phishing attacks or anonymously post spam messages. Our previous work on browser security (Mylonas et al., 2013) revealed that security controls, which are typically found on desktop browsers, are not provided by their smartphone counterparts. Finally, our work in Virvilis et al. (2014) revealed significant differences in the effectiveness of phishing blacklists of Android, iOS, and desktop browsers.

2.2. *Malware*

Although the majority of browsers rely solely on blacklists to detect malicious URLs, on Windows, Internet Explorer and Chrome perform further analysis to detect malicious downloads. Both browsers analyze the metadata of the downloaded file (i.e. digital signature, hash, file's popularity, etc.) to warn users about potential risks (Rajab et al., 2013; Microsoft, 2014; Colvin, 2011). Furthermore, non-browser specific models have been proposed for the detection of malicious domains through DNS monitoring (Antonakakis et al., 2011), while Bilge et al. (2014) discusses large-scale, passive DNS analysis techniques to detect domains that are involved in malicious activity. A zero-day anti-malware solution is proposed in Shahzad et al. (2013), which uses a combination of whitelists and blacklists. The authors discuss that such whitelists do not need signature updates and provide protection against sophisticated zero-day malware attacks by enforcing software restriction policies, which allow only legitimate executables, processes and applications to be executed.

In addition, a number of models have been suggested in the literature focusing on malware detection, namely: (a) the AMICO project (Vadrevu et al., 2013), which detects malware downloads in live web traffic using statistical analysis to identify characteristics of malware distribution campaigns, (b) the ZOZZLE (Curtsinger et al., 2011), which detects and prevents JavaScript malware from been deployed in the browser, and (c) the EFFORT system (Shin et al., 2012), which focuses on the detection of malware serving bots. Furthermore, multiple models have been proposed that rely on machine learning techniques: (a) for malware detection (Kolter and Maloof, 2006; Perdisci et al., 2008; Antonakakis et al., 2011; Perdisci et al., 2008) and (b) for detection of drive-by downloads (Caballero et al., 2011; Lu et al., 2010a; Lu et al., 2010b; Provos et al., 2007; Provos et al., 2008; Zhang et al., 2011b).

The industry offers a large number of content filtering solutions, ranging from software-based solutions, to Cloud services and hardware appliances. Multiple software solutions exist, such as McAfee's Site Advisor and Symantec's Safe Web (McAfee, 2014b; Symantec, 2014b). They are usually offered for free or at low cost. However, they require the installation of third-party software/browser extensions and are browser and platform dependent, with limited support for mobile platforms. Commercial content filtering appliances and Cloud Services (e.g. OpenDNS (OpenDNS)) are popular in enterprises and are usually platform and browser agnostic. However, their effectiveness is hard to measure due to the use of proprietary technologies. Also, their significant cost limits their use.

Lastly, a number of online services exists offering file analysis. One of the most popular is VirusTotal (VirusTotal), which utilizes a large number of popular antivirus engines to analyze (suspicious) files (c.f. Table 17 in the Appendix for a list of AV engines). Users can upload and analyze their files, or query the service for files that have already been analyzed by searching their hash. VirusTotal also supports URL scans, querying a URL against a large number of blacklists (c.f. Table 18 in the Appendix).

3. Methodology

3.1. *Test environment*

Our experiments were conducted in June and July 2014 and focused on the most popular browsers in the desktop and smartphone platforms (c.f. Appendix, Tables 9–10), namely:

- **Desktop browsers.** Internet Explorer 11, Chrome v35, Firefox v29, and Opera v22, which were installed on a Windows 7 64-bit system.
- **Mobile browsers.** Safari Mobile (built-in on iOS 7.1.1), Chrome Mobile v35, Opera Mini v7.0, "Browser" or "Internet" (i.e. the default browser for Android 4.0.4), Firefox Mobile v30, and Opera Mobile v22, which were installed on an iPhone 5S (iOS v7.1.1) and a Sony Xperia Tipo (Android v4.0.4). Although some of the desktop browsers have mobile counterparts, their availability in the two smartphone platforms is heterogenous, as shown in Table 1.

Our effort in the smartphone platform focused on iOS and Android, as these are the two most popular operating systems, comprising almost 90% of the global smartphone market share (Bradley, 2013).

To evaluate the protection that is offered to users against rogue web sites, we accessed 1400 phishing and 1400 malicious URLs and marked whether the browsers warned us about the risk of our action. As technology that can be used to fully automate this process is not currently available, a security savvy user manually verified if a web page that had not been reported by the browser, was indeed a rogue or a benign

Table 1 – Browser availability on tested platforms.

	iOS 7.1.1	Android 4.0.4	Windows 7
Safari Mobile	X		
Chrome Mobile	X	X	
Opera Mini	X	X	
Browser[a]		X	
Firefox Mobile		X	
Opera Mobile		X	
Chrome			X
Firefox			X
Internet Explorer			X
Opera			X

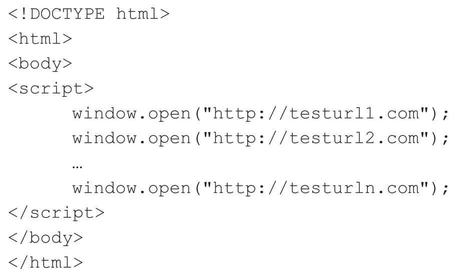
[a] 'Browser' (or Internet in newer versions) is the pre-installed browser on Android.

web page. Since this was a cumbersome task, we set up the architecture that is presented in Fig. 1.

The URL Collection included the URLs that were used in our evaluation. The URL Container parsed daily the URL Collection and selected the URLs that had been reported in the last 24 h. Then, two HTML files were created, one for phishing URLs and one for malicious URLs, which were formatted as Snippet 1. Finally, a researcher used each browser installed on the test devices, to access each HTML file and collect the number of: a) blocked URLs, b) false negatives, and c) non-phishing/malicious URLs.

3.2. *Phishing tests*

To evaluate the anti-phishing protection that is offered by the aforementioned web browsers, we used 1400 phishing URLs collected from PhishTank (Phishtank, 2014). PhishTank was selected as it is a popular online service that lists phishing URLs, which are verified by an active community. PhishTank publishes every day a list of verified phishing URLs, i.e. ones that have been confirmed as fraudulent and online. However, as the state of a phishing URL is dynamic, a confirmed URL might be cleaned or be taken down shortly after its submission.



```
<!DOCTYPE html>
<html>
<body>
<script>
      window.open("http://testurl1.com");
      window.open("http://testurl2.com");
      …
      window.open("http://testurln.com");
</script>
</body>
</html>
```

Snippet 1. HTML content

Although part of the experiments could have been automated (e.g. when the request returned an HTTP Error Code or the browser raised a warning), manual review was required in order to measure the false negatives (actual phishing URLs, which were not blocked by the browser). In specific, we manually examined each URL that was not blocked by the browser and classified it as (a) benign or not responsive/non accessible site (i.e. inactive phishing site) or (b) false negative, i.e. an active phishing site that was not blocked by the browser.

Thus, in the end of the experiment every URL in the phishing collection was manually classified as:

a) ***Blacklisted***: The phishing URL was blocked by the web browser, i.e. the user received a warning indicating a phishing site.
b) ***False Negative***: An active phishing URL that was manually verified as fraudulent, but was not blocked by the browser.
c) ***Non-Phishing/Timeout/Error***: The phishing URL had been suspended/taken down/cleaned when we accessed it.

It should be noted that our data set included only verified phishing URLs (i.e. a human operator from PhishTank has manually verified them as fraudulent), as our main objective was to examine the efficacy of browsers' anti-phishing blacklists. Therefore, the false positive rate of all browsers, which is out of the scope of this work, was zero. Finally, we compared our results with our previous work in Virvilis et al. (2014).

3.3. *Malware tests*

An online, well-known service that reports on a daily basis verified malware-hosting websites, offering strong community support (comparable to PhishTank), is not currently available. Therefore, to gather the malicious URL collection we used the open source *Collective Intelligence Framework* (CIF) (CIF, 2014). CIF allows the collection and analysis of malicious threat information from a large number of trusted sources (c.f. Appendix for a list of these sources), which is used for incident response and intrusion detection and mitigation.

Similarly to the anti-phishing experiment, our tests included manual browsing to 1400 URLs that hosted malicious software. Every URL was categorized as follows:

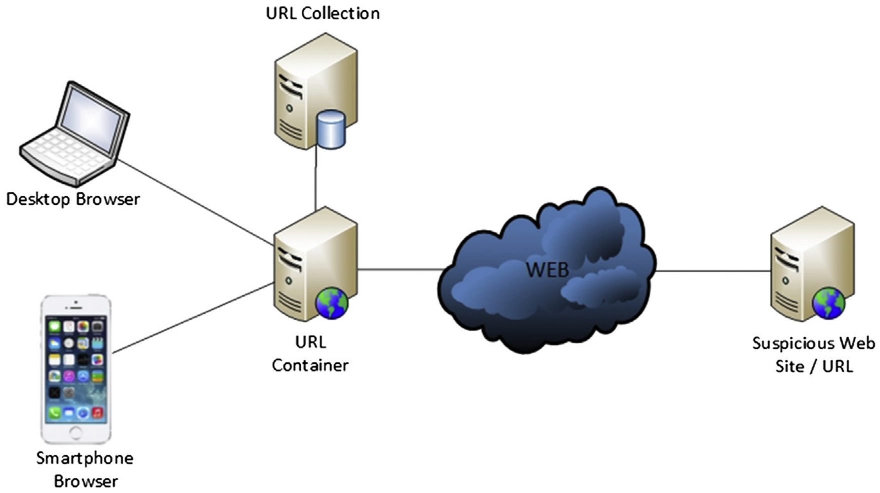


Fig. 1 – Laboratory setup.

a) ***Blacklisted***: The browser blocked either the URL or the file that was downloaded (or issued a warning that the file could be potentially dangerous).
b) ***False Negative***: A URL that was not blocked by the browser and triggered the download of a potentially malicious file, without any further alert being raised by the browser.
c) ***Non-Malicious/Timeout/Error***: The URL had either been cleaned or suspended/taken down when we accessed it. This category also included the URLs that did not trigger a download.

Similarly, the dataset for this test included only verified malicious URLs as this work examines the efficacy of browser blacklists in blocking such attacks. As a result, the false positive rate of all browsers, which is out of the scope of this work, was zero.

4. Experimental results

4.1. *Protection against phishing sites*

4.1.1. *iOS browsers*

Mobile Safari, which is the pre-installed iOS web browser, uses Google's Safe Browsing to provide anti-phishing protection. Our evaluation revealed that the implementation of this anti-phishing control suffers from a significant design weakness, as the Safe Browsing blacklist is only updated when the iOS device is synchronized with iTunes. Considering that some iOS users may not synchronize their devices frequently, they may end up with an outdated blacklist. Thus, any phishing site that has been created in the meantime – even if it has been reported to Safe Browsing list – will not be blocked. As a result, iOS users receive considerably limited protection against phishing attacks. In fact, during our experiments Safari Mobile did not block any phishing URL when we skipped this synchronization step. Therefore, the iOS test device was synced daily, right before starting our evaluation.

Chrome Mobile offers phishing protection since January 2014. However, this option is not enabled by default, but requires the user to enable the “Reduce Data Usage” option, which uses Google's servers as a proxy to fetch the requested URL. When this option is enabled, the contents of the web page are downloaded and compressed and the URL is checked against the Safe Browsing list. This feature is privacy intrusive – as all traffic is transferred through Google's servers – and does not work for SSL/TLS pages or in Incognito mode (private browsing). We have excluded this browser from our evaluation as: a) we regard that it is less likely that normal users (i.e. not security and savvy ones) would enable security controls, as smartphone users tend to be oblivious about their security and privacy (Mylonas et al., 2013) and (b) the control is not ‘easily configurable’ (Mylonas et al., 2013), i.e. the label of the control is not intuitive and confusing even for security savvy users. Finally, Opera Mini did not support anti-phishing (See Table 2).

Table 2 – Phishing protection statistics on iOS.

	Blacklisted	False negatives	Non-phishing
Safari Mobile	542	370	488
Chrome Mobile	N/A	N/A	N/A
Opera Mini	N/A	N/A	N/A

N/A: Browser does not offer anti-phishing mechanisms.

4.1.2. *Android browsers*

Android users also receive limited protection against phishing attacks, as the default Android browser (known as “Browser” or “Internet”) does not offer phishing protection. The same holds true for Chrome Mobile and Opera Mini. It must be noted that these are the most popular browsers on Android, according to the number of downloads on Google Play (c.f. Table 10 in the Appendix).

On the other hand, Firefox Mobile and Opera Mobile offer anti-phishing protection. Our results suggest that both browsers offer similar protection with their desktop counterparts. Specifically, Firefox and Opera on Windows blocked 86.7% and 77.9% of the phishing URLs and Firefox Mobile and Opera Mobile blocked 85.4% and 75.9%, respectively (c.f. Fig. 6) Nevertheless, if one considers that: (a) not all users feel the need and/or are capable to install a third-party browser on their devices (Mylonas et al., 2013) and (b) the pre-installed browser offers no anti-phishing protection, then a large number of Android users is not protected from phishing attacks (See Table 3).

4.1.3. *Desktop browsers*

Our analysis revealed that all desktop browsers offered anti-phishing protection using either Safe Browsing list (Chrome and Firefox) or their own proprietary blacklists (Opera and Internet Explorer). The most phishing URLs were blocked by Chrome and Firefox. Although their results are similar – which is expected as they use the same blacklist – Chrome outperforms Firefox, as in our experiments it blocked roughly 5% more phishing sites and has a lower false negative rate.

During our experiments we encountered another issue with the synchronization of blacklists in Firefox, which was also raised by Abrams et al. (2014). Specifically, each day Safe Browsing blacklist in Firefox was not updated, unless Firefox was executed for a few minutes before our evaluation, which resulted to a large number of false negatives. This stems from the way the Safe Browsing protocol updates its local database (Sobrier, 2014). Interestingly, this problem did not appear in Chrome or any smartphone browsers that use Safe Browsing. To avoid this synchronization issue, we executed

Table 3 – Phishing protection statistics on Android.

	Blacklisted	False negatives	Non-phishing
Firefox Mobile	1196	48	156
Opera Mobile	1062	110	228
Chrome Mobile	N/A[b]	N/A	N/A
Opera Mini	N/A	N/A	N/A
Android Browser[a]	N/A	N/A	N/A

[a] ‘Browser’ (or Internet) is the pre-installed Android browser.
[b] N/A: Browser does not support anti-phishing mechanisms.

Table 4 – Phishing protection statistics on Windows 7.

	Blacklisted	False negatives	Non-phishing
Firefox	1215	83	102
Chrome	1302	18	80
Opera	1090	118	192
Internet Explorer	678	138	584

Firefox for at least 10 min to allow the browser to update its blacklist.

Opera blocked roughly 10% less phishing sites than Firefox and had slightly more false negatives. Finally, Internet Explorer offered the lowest level of protection among the desktop browsers, having the smallest percentage of blocked URLs (less than 50%) (See Table 4).

4.1.4. *Comparison with previous evaluation*

Herein, we compare the anti-phishing protection that the popular desktop and mobile browsers offer in Q2 2014 with the results of our previous work in Virvilis et al. (2014), which was conducted in Q1 2014. Since the phishing 'ecosystem' is dynamic, i.e. phishing sites are short-lived with an average life expectancy of 23 h (Abrams et al., 2014), our aim is to examine how this dynamic nature is reflected in the browsers' anti-phishing protection over this period of time.

Our results are summarized in Fig. 2 (c.f. Tables 12–14 in the Appendix for detailed results). The browsers blocked fewer phishing URLs in Q2 with the exception of Firefox Mobile on Android. Safari Mobile's (iOS) detection dropped almost by half. This stresses again the problematic implementation of the Safe Browsing protocol on iOS. Furthermore, our analysis showed a small decrease in the performance of the desktop versions of Firefox and Opera, with respect to the blacklisted URLs and false negatives. Finally, Internet Explorer blacklisted less URLs and was prone to more false negatives, and Opera Mobile had more false negatives.

4.2. *Protection against malicious sites*

4.2.1. *iOS browsers*

Our results revealed that none of the iOS browsers offered any protection against malicious sites, leaving their users exposed to this threat. In the case of Mobile Safari this was rather surprising, as the browser uses Safe Browsing to provide anti-phishing protection, but it does not provide detection of malicious sites. Opera Mini did not utilize any blacklist for the detection malicious sites and neither did Chrome Mobile (not enabled by default and excluded due to the shortcomings that were mentioned previously) (See Table 5).

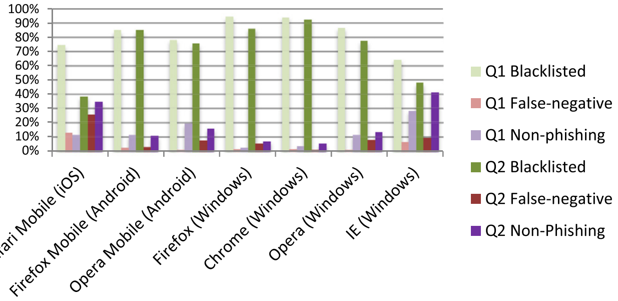


Fig. 2 – Comparison of anti-phishing protection (Q1 2014 – Q2 2014).

Table 5 – Malware protection statistics on iOS.

	Blacklisted	False negatives	Non-malware
Safari Mobile (iOS)	N/A	N/A	N/A
Chrome Mobile	N/A	N/A	N/A
Opera Mini	N/A	N/A	N/A

N/A: Browser does not support anti-malware mechanisms.

4.2.2. *Android browsers*

Our results suggest that Android users are also unprotected against malicious sites. This finding is very worrying if one considers the increasing number of attacks against Android and the exponential growth of Android malware (Funk and Garnaeva, 2013; Zhou and Jiang, 2012; Zhou et al., 2012). Specifically, our results revealed that the pre-installed web browser ("Browser" or "Internet"), Chrome Mobile and Opera Mini offered no protection against malicious sites. As summarized in Table 6, only Firefox Mobile and Opera Mobile utilized malware blacklists. Nonetheless, our results suggest that the level of offered protection was very limited, as they blocked only 10–12% of the malicious URLs.

4.2.3. *Desktop browsers*

The results suggest that popular desktop browsers offer poor protection against malicious sites. Internet Explorer (IE) blocked the most malicious sites and was the least prone browser to false negatives, which confirms findings from similar research conducted by the industry (Abrams et al., 2014). Nevertheless, even though IE outperformed all the other browsers, it only blocked ~41% of the malicious URLs and had a 30% of false negatives. This highlights that the application reputation mechanism that is used by IE does offer an extra line of defense against malware, but is far from perfect.

Chrome had more false negatives than IE, even though it offers a similar mechanism against malicious downloads. Opera ranked third in terms of blocking malicious sites, but had 58% of false negatives, the highest in the experiments. Firefox

Table 6 – Malware protection statistics on Android.

	Blacklisted	False negatives	Non-malware
Firefox Mobile (Android)	139	641	620
Opera Mobile (Android)	166	683	551
Chrome Mobile	N/A	N/A	N/A
Opera Mini	N/A	N/A	N/A
Browser[a]	N/A	N/A	N/A

N/A: Browser does not support anti-malware mechanisms.

[a] 'Browser' (or Internet in newer versions) is the pre-installed browser on Android.

Table 7 – Malware protection statistics on Windows.

	Blacklisted	False negatives	Non-malware
Firefox	70	729	601
Chrome	280	552	568
Opera	180	816	404
Internet Explorer	573	420	407

offered the poorest protection, blocking only 5% of the malicious URLs and having a ~43% of false negatives (see Table 7).

During our experiments Firefox and Opera blocked malicious sites only by examining their URLs, without analyzing the downloaded files. Newer versions of Firefox now analyze the downloaded files using the same technology as Chrome (Mozilla). This raises the level of anti-malware protection and we assume that its efficacy would be similar to Chrome's, since the same reputation-based mechanism for downloaded files is used.

5. Secure proxy

Our results uncovered three problems that must be addressed to protect users from rogue sites:

a) The limited effectiveness of blacklists against malicious sites and phishing sites.
b) The limited effectiveness of reputation based mechanisms (e.g. in Internet Explorer and Chrome) to block malicious downloads.
c) The unavailability of the relevant security controls in popular mobile browsers.

In this context, we propose and implement *Secure Proxy*, as a proof-of-concept security mechanism, which proves the efficacy of aggregating multiple data sources in the detection of rogue sites. Currently, and due to the heterogeneity and restrictions of smartphones and their security models, a different security control might be incompatible (e.g. as on iOS) or infeasible to be implemented due to resource restrictions (e.g. on older Android smartphones). The proposed proxy is browser and platform agnostic (i.e. does not require the installation of third party software) and can protect the user, regardless of the browser she is using. Finally, and in contrast with the commercial and closed source content filtering solutions, we are proposing an open architecture which can be built with a fraction of the cost.

5.1. *Architecture*

We have implemented a secure forward HTTP proxy, which uses VirusTotal's public API to analyze the requested URLs and downloaded files. VirusTotal was selected due to (a) its popularity, (b) the number of AV engines and blacklist providers that it offers, and (c) the availability of free API. Nevertheless, any other similar service could be used.

The proxy queries VirusTotal for each requested URL to identify if the URL is blacklisted by any of the blacklist providers. If the URL is blacklisted, the request is blocked and a warning message is returned to the user. Otherwise, the proxy returns the HTTP response (i.e. page contents) to the browser. If a download is triggered, the proxy calculates the SHA-256 hash of the file and queries VirusTotal. Once more, if the hash is known and is reported as malicious by any AV vendor, then the download will be blocked and a warning will be raised. If the hash is unknown (i.e. the file has not been analyzed beforehand), then the proxy uploads the contents of the file for analysis and allows the user to download the file if it does not get flagged as malicious by the AV engines. These steps are summarized in Fig. 3.

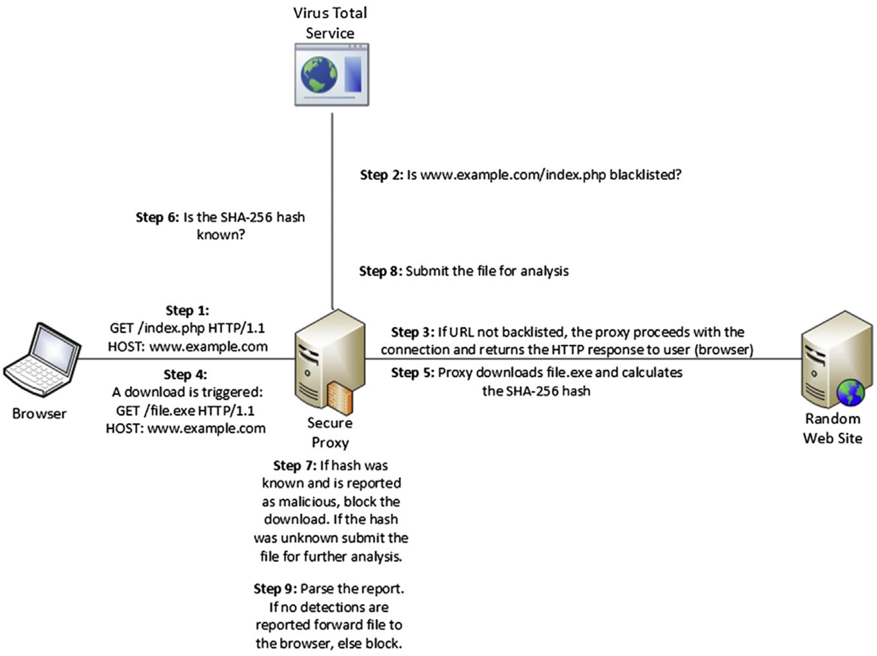

Virus Total Service
Step 2: Is www.example.com/index.php blacklisted?
Step 6: Is the SHA-256 hash known?
Step 8: Submit the file for analysis
Step 1: GET /index.php HTTP/1.1 HOST: www.example.com
Step 3: If URL not backlisted, the proxy proceeds with the connection and returns the HTTP response to user (browser)
Step 5: Proxy downloads file.exe and calculates the SHA-256 hash
Browser
Step 4: A download is triggered: GET /file.exe HTTP/1.1 HOST: www.example.com
Secure Proxy
Random Web Site
Step 7: If hash was known and is reported as malicious, block the download. If the hash was unknown submit the file for further analysis.
Step 9: Parse the report. If no detections are reported forward file to the browser, else block.

Fig. 3 – Proposed architecture.

Table 8 – Percentage of blacklisted malicious sites from Secure Proxy vs. Internet Explorer (n = 1400).

Secure proxy	Best browser (Internet Explorer)
53.2%	40.9%

5.2. *Secure proxy evaluation*

The evaluation of the *Secure Proxy* focused on the detection of malicious sites, as the majority of browsers provide only weak protection (or hardly any) against them. The proxy was also tested with the complete collection of phishing URLs from PhishTank. However, this test was only a verification of the correct operation of the *Secure Proxy*, as PhishTank is one of the anti-phishing providers that is used by VirusTotal.

To evaluate the *Secure Proxy* we accessed every day the same list of malicious URLs that the browsers were tested against, redirecting the requests through it. We used a script that simulated the web requests instead of using any browser, to make sure that no browser specific countermeasure would interfere with our results, as well as to automate this process. We collected the number of blocked URLs and compared them with the results of the browser that achieved the highest blocking rate in our evaluation. The *Secure Proxy* was configured to block downloaded files (either based on the hash or the actual file analysis), when the number of detections reported by VirusTotal were at least one. This parameter is configurable and it can be set to block a request with a different blocking threshold, according to the existing security policy or risk appetite.

We also collected statistics regarding: (a) the number of URLs that were blocked due to URL-only analysis, (b) the number of downloads that were blocked due to hash analysis, and (c) the number of downloads that were blocked due to file based (content) analysis.

5.2.1. *URL-only and hash-based analysis*

As summarized in Table 8, the use of multiple blacklists enabled the *Secure Proxy* to block almost half of the malicious URLs in our experiment – thus outperforming Internet Explorer by 12.3% – which was the browser that blocked the most malicious URLs. This proves that while the aggregation of multiple blacklists provides higher protection than any individual browser, it still fails to detect almost half of the malicious URLs in our collection (i.e. 46.8% false negatives).

Browsing to these URLs triggered the download of 460 unique files (based on their SHA-256 hash), all of which were PE executables. *Secure Proxy* downloaded these files and queried VirusTotal for their hashes. Our results revealed that 57.3% of the submitted hashes (i.e. 264 out of 460) were unknown to VirusTotal, i.e. the files had not been submitted for analysis beforehand. The detection rate of the rest of the files is summarized in Fig. 4 (for detailed results see Table 15 in the Appendix).

The number of AV engines that are available during the analysis of a file in VirusTotal ranges between 49 and 54 antivirus engines. Fig. 4 summarizes the results based on the detection ratio for each file. This ratio was calculated with z/n, where z represents the number of antivirus engines that flagged the file (hash) as malicious and n is the number of antivirus engines that were used. The results indicated that the detection rate of approximately half of the malicious files was in the range of 6–38% of the antivirus engines. Moreover, only the 27.4% of the malicious files were detected by the majority of the antivirus engines.

5.2.2. *File-based analysis*

During our experiments, *Secure Proxy* uploaded 264 (57.3%) of the downloaded files to VirusTotal for analysis, as their hashes were unknown. All of them were reported as malicious and the majority of them (~71.0%) were detected by 46–50% of the antivirus engines (c.f. Fig. 5 and Table 16 in the Appendix). This suggests that with the absence of file-based analysis from the *Secure Proxy*, there is ~50% likelihood that malicious files

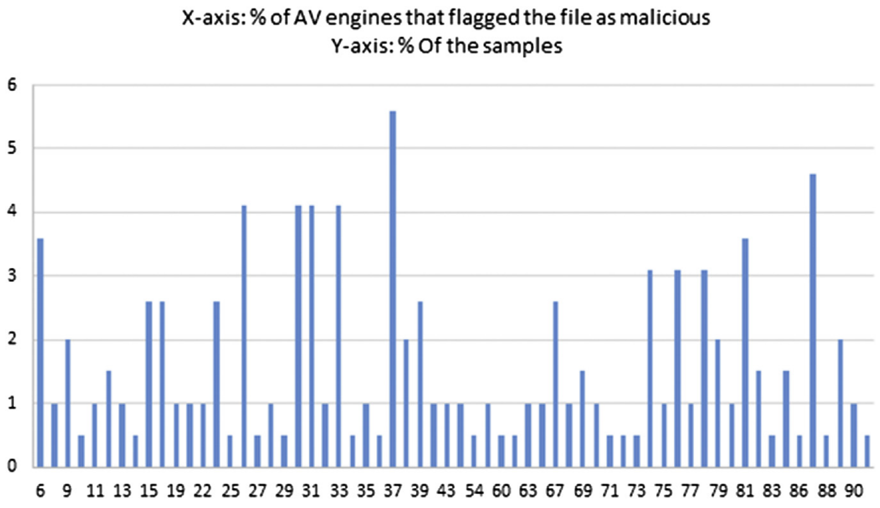


Fig. 4 – Detection percentage of hashes (X-axis lists the percentage of AV engines that flagged the file as malicious, Y-axis lists the percentage of the samples).

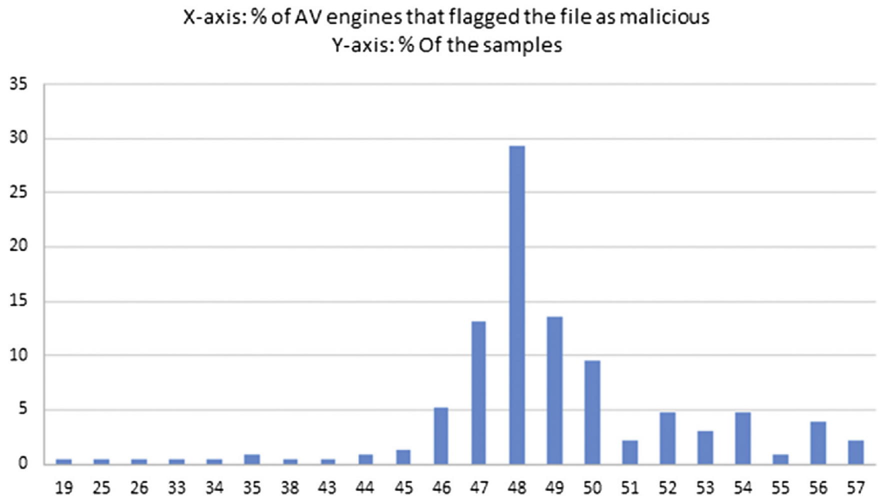


Fig. 5 – Detection percentage of executables (X-axis lists the percentage of AV engines that flagged the file as malicious, Y-axis lists the percentage of the samples).

will not be detected from the user's AV (assuming that the user has installed only one antivirus program).

Finally, the abovementioned detection rates refer to desktop antivirus engines. Mobile antivirus applications are more constrained due to the sandboxed environment of iOS and Android and also lack advanced features that are available at desktop antivirus engines (i.e. advanced heuristic analysis).

5.2.3. *Performance evaluation*

As mentioned beforehand, *Secure Proxy* performs three types of queries to VirusTotal, which incur delays: (a) URL Query to identify if the URL is reported as rogue by any of the blacklist providers, (b) Hash Query to identify if the hash of the file is known to be malicious, and (c) File Query in which the actual file is uploaded to VirusTotal for analysis.

Our analysis revealed that for URL Queries the proxy received a response from VirusTotal and allowed or blocked the request on average in 648 ms. The average Hash Query time for each triggered download was 516 ms. The longest delay occurs when the hash is unknown as the file needs to be uploaded for further analysis (File Query). The delay depends on various parameters, e.g. the size of the file, the network speed, and the load on the VirusTotal service – with the latter often being the most time consuming parameter. In our evaluation, the average size of the collected malicious executables was 848 KB and our Internet connection was a 20Mbit (2048Kbit upload) ADSL line. On average, our file queries were completed in under 41sec, including the time required to upload a file and get the detection report.

6. Disscussion

6.1. *Limitations*

Our corpus was limited to 2800 rogue URLs (1400 phishing and 1400 malicious URLs), due to the significant manual effort that was required to test different browsers on Windows, iOS and Android. This introduced a potential bias in our results, which describe the level of protection in the period that the tests took place, namely June–July 2014. However, even though our results are not generalizable, we consider that they provide adequate indications about the level of protection offered to the users for two reasons: a) our findings for desktop browsers are in accordance with the results in Abrams et al. (2014) and b) the results of this work regarding the effectiveness of anti-phishing protection are similar to our previous evaluation of a much larger data set of 5651 URLs (Virvilis et al., 2014).

Our work focuses only on the popular desktop browsers (Windows) and their smartphone counterparts that are available on iOS and Android. While there are other browsers that we did not examine, such as Safari on Mac OS X and Internet Explorer Mobile on Windows Phone, we consider our results as representative. This holds true as Windows is the most popular operating system for desktops and laptops, as well as Android and iOS users constitute the 94% of the smartphone users (78.4% and 15.6% respectively) (Gartner, 2013). In addition, as iPads and Android tablets use a similar operating system (iOS, Android), and in most cases the exact same browser versions, our findings are considered to reflect the protection that is offered on a larger user base.

In this work we proposed *Secure Proxy*, as a countermeasure against rogue sites. It is worth noting that our implementation is a proof-of-concept, highlighting the benefits that the aggregation of multiple blacklists and AV engines offers against rogue websites. Our evaluation regards issues such as privacy or performance as out of scope of this work. However, as discussed in Section 6.3, both issues can be addressed in a real-world implementation.

The implementation of *Secure Proxy* is based on the public version of VirusTotal's API, which introduces limitations. Firstly, VirusTotal is a service which was not designed to support semi-real time queries, as the ones used by the proposed control. A dedicated service optimized for such use, such as CloudAV (Oberheide et al., 2008), might achieve better performance and as it can be hosted locally, it avoids privacy concerns. Also, *Secure Proxy* – similarly to VirusTotal – does not weight differently the responses from the various antivirus engines or URL blacklists. It can be extended to assign different weights to these responses according to organization's security policy.

Finally, our results are affected by the dynamic nature of the web ecosystem. This is due to the dynamic nature of the threats and the new evasion techniques that attackers create. This is reflected on the comparison of the anti-phishing protection that is offered by the examined browsers in Q1 and Q2 (2014). Moreover, browsers add to the complexity of the evaluation due to their frequent updates, which might include new security controls (e.g. analysis of downloaded files is now supported in newer versions of Firefox).

6.2. *Protection of desktop and mobile browsers*

Overall, our results revealed that desktop browsers performed better in comparison to their smartphone counterparts, both against phishing and malicious sites. This is a worrisome finding if we consider the proliferation of smartphones, as well as the increased web browsing with these devices. One could argue that this is expected, as smartphones lack the processing capabilities of desktops and laptops. Nonetheless, this is only partly true today, as most smartphones have similar resources as a 3–4 year old laptop (e.g. dual core CPU, 1–2 GB or RAM, etc.). In addition, our previous work has shown that the unavailability of important security controls – such as blacklists for phishing and malicious sites in which this paper focuses – does not stem from the (API) restrictions that are imposed from the smartphones operating system (i.e. sandbox profile) (Mylonas et al., 2013). The reason that this happens is still unclear; however, it falls out of the scope of this work.

More specifically, our results revealed that only a subset of the mobile browsers offer anti-phishing protection and thus, their users are not protected from such attacks. This is particularly true for Android users, where the pre-installed browser does not offer anti-phishing protection. On iOS, the pre-installed browser offers anti-phishing protection, but its effectiveness is questionable (c.f. Section 4.1.1). On the contrary, all desktop browsers provided anti-phishing protection, even though their effectiveness was significantly different and blacklist synchronization issues were identified for Firefox on Windows. Fig. 6 summarizes the results of our anti-phishing experiments.

Contrary to anti-phishing protection, the results suggest that both desktop and mobile browsers offer very limited protection against malicious URLs (Fig. 7). While Internet Explorer and Chrome, which used application reputation mechanisms, blocked more malicious URLs/resources than other browsers, their detection rate was low. Moreover, only a subset of the browsers on iOS and Android offer any protection against malicious URLs – the browsers that did not block phishing URLs also did not block malicious URLs.

Interestingly, Safari Mobile did not block malicious URLs, even though it uses Safe Browsing to offer anti-phishing protection. Apple may have assumed that using such a blacklist would not increase the level of protection for iOS users, based on the fact that iOS devices only execute code

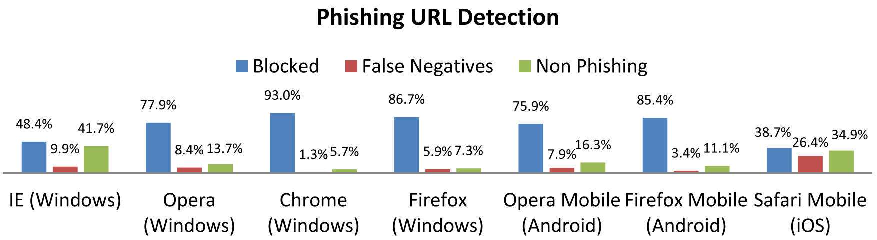


Fig. 6 – Percentage of: a) blocked phishing URLs, b) false negatives, i.e. active phishing URLs that were not blocked, c) phishing URLs not blocked but not hosting a phishing attack at the time of the analysis (i.e. cleaned, domain taken down or inaccessible) (n = 1400).

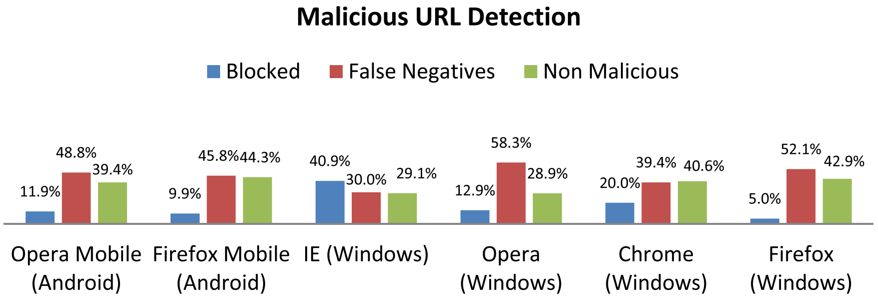


Fig. 7 – Percentage of: a) blocked malicious URLs, b) false negatives, i.e. malicious URLs that were not blocked, c) malicious URLs not blocked not hosting an attack at the time of the analysis (i.e. the domain/URL had been cleaned/taken down, or a download was not triggered) (n = 1400).

signed by Apple. This assumption seems flawed as: (a) a significant percentage of users jailbreak their iOS devices, thus the device can also execute unsigned code (Love, 2013), (b) if iOS users do not receive any warning when visiting a malicious site, they can unwillingly put other users at risk by forwarding/sharing the URL, and (c) files that are downloaded to an iPhone might be synchronized to a computer, resulting to its infection.

6.3. *Proposed countermeasure*

To raise the level of protection that is offered to users against rogue sites, we proposed, implemented as a proof of concept and evaluated, *Secure Proxy*. *Secure Proxy* is a HTTP forward proxy that uses multiple anti-phishing and anti-malware blacklists. Our work suggests that such browser agnostic architectures are currently the only solution for protecting normal (i.e. not security savvy) smartphone users, as they are less likely to install third-party browsers or security products on their devices (Mylonas et al., 2013).

The evaluation showed that *Secure Proxy* blocks more rogue URLs and is less prone to false negatives than any individual browser. Our work focused on reducing false negatives, i.e. active phishing or malicious URLs that were not blocked by the browsers, as they could result in a successful attack. *Secure Proxy* however, may be prone to false positives as some sites may have erroneously been added to a blacklist without proper verification. In this case, a URL might be blocked until it is removed from all blacklists causing a temporal nuisance to users. Nonetheless, popular blacklists (e.g. Safe Browsing) allow site administrators to request their site's removal from a blacklist when it has been cleaned, which would reduce the inaccessibility time. Furthermore, *Secure Proxy* can be configured, based on the user's or organization's risk appetite, to block a URL if the number of blacklists that identify it as malicious exceeds a threshold, thus reducing potential false positives. However, specifying this threshold falls outside the scope this work.

Similarly to Internet Explorer's and Chrome's metadata analysis of the downloaded files, *Secure Proxy* offers hash-based analysis with the aggregation of multiple AV engines. In specific, it queries the file's hash on VirusTotal and blocks its download if it is reported as malicious by at least one AV engine – similarly to URL-only analysis this threshold is configurable. Our results prove that hash-based analysis adds an extra layer of protection against malicious sites and – as URL-only analysis – does not introduce significant delays.

The benefit of online queries is that the URL or hash will always be checked against an up-to-date blacklist, avoiding any blacklist synchronization issues. In addition, using a browser agnostic countermeasure, such us the proposed countermeasure, enables the end user to use a browser that does not offer any build-in countermeasures and still be protected. Finally, it also allows devices with limited resources (e.g. smartphones) to avoid resource intensive operations and thus reduces energy consumption.

The inherent drawback of any online, centralized architecture is user privacy. Each visited URL is submitted to a third party for analysis, thus exposing the users' browsing history/profiles. Even though this falls outside the scope of this work, it can be mitigated by maintaining a local blacklist/whitelist, thus avoiding the need for a central architecture (e.g. as in our case, a proxy server) – similarly to the way Safe Browsing protocol works. The browsers that implement the protocol, keep a local database of reported URLs, which is updated frequently while the browser is running. As a result, all lookups use the local database, thus avoiding unnecessary delays and privacy issues. Based on the fact that VirusTotal is owned by Google, it seems fairly easy to include the aggregated results from all blacklist providers to a single list (e.g. imported into Safe Browsing list). Still, this will require the browser vendors to implement/adopt the Safe Browsing protocol and make sure that they avoid any synchronization issues. Another potential solution could be to host a service similar to VirusTotal locally i.e. CloudAV (Oberheide et al., 2008), which would also address the privacy issues.

Our tests have also revealed the significant benefit of aggregating multiple AV engines for the detection of malicious files. *Secure Proxy* uploads for further analysis, all downloaded files for which a hash-only query returned no results. This analysis introduces delays (41 s on average in our experiments), which may not be acceptable for some users. However, *Secure Proxy* can be configured to upload these files according to a policy, e.g. by default deny access to these files, move the files in a sandbox or ask the user to decide whether the files will be submitted for further analysis. The last option however assumes users' security awareness, which might not be the case for all users, as they are known to click through security messages (Akhawe and Felt, 2013; Egelman and Schechter, 2013; Mylonas et al., May 2013; Mylonas et al., August 2013).

Similarly to other instances in the security domain, there is a tradeoff between security and usability. A combination of a whitelist and reputation based system will further limit the number of files that have to be submitted for analysis – as only files that are not included in the whitelist and are not blocked by the reputation system have to be analyzed. Nevertheless, the benefits of such detailed analysis are significant. This holds true as our results suggest that even if the user runs an AV engine, it will fail to identify all malicious files (in our experiments on average an AV detected only half of the malicious files). On the contrary, *Secure Proxy* offered better anti-malware protection due to the aggregation of multiple AV engines and blocked all the malicious files in our corpus.

7. Conclusions

This paper provides an evaluation of the build-in protection mechanisms that are offered by web browsers against rogue web sites, namely phishing sites and sites hosting malware (‘malicious sites’). Our work focuses on the most popular desktop (Windows) browsers, as well as their Android and iOS counterparts. The browsers were tested against a data set of 2800 rogue URLs (1400 phishing and 1400 malicious URLs).

Our results uncover the shortcomings of the current security controls and highlighted the substantial security gap, between desktop and mobile browsers regarding the protection from rogue URLs. Mobile users are significantly exposed, as the default browser on iOS offers limited protection against rogue URLs, while the default browser on Android offers no protection. The performance of Windows browsers also differs significantly, especially between phishing and malicious sites. Furthermore, we highlighted synchronization problems for some of the browsers using the Safe Browsing protocol, which limit the level of protection that is offered.

To address these threats, we implemented and evaluated - as a proof of concept – *Secure Proxy*, a forward HTTP proxy, which is based on the aggregation of multiple blacklists and AV engines. *Secure Proxy* performs URL analysis using multiple blacklists and significantly increases the level of protection against malicious downloads by performing (a) hash based lookups and (b) content scanning.

Our work has proved that the aggregation of multiple blacklists and AV engines can raise significantly the level of protection against rogue sites, regardless of the user's device (smartphone or desktop). We regard that our results are useful both to the users and browser vendors. The former can be informed about the availability of protection against rogue web sites, which might help them choose a web browser based on an informed security decision. For the latter, this work may stimulate browser vendors to adjust their current anti-phishing and anti-malware controls and/or implement additional controls, which would eliminate the privacy and performance limitations of our work and offer increased protection compared to their current deployed solutions. In the meantime, we envisage that the proposed security control can be used as the basis of a forward proxy, which protects both smartphone and desktop users in an organization.

For future work, we plan to further examine the protection that browsers offer against phishing and malicious sites. To this end, we plan to revisit and extend our experiments, increasing our URL corpus, as well as including new browsers (e.g. Internet Explorer on Windows Phone and Safari browser on Mac OS X). Furthermore, we plan to investigate if the performance is affected when a local service is used, which combines multiple AV engines (i.e. CloudAV) and URL reputation lists, instead of using an online service (i.e. VirusTotal). We assume that this approach, apart from the expected performance improvement, would also avoid the potential privacy issues that have been discussed in this work.

Appendix

Table 9 – Desktop browser popularity (June–July 2014). Source: http://gs.statcounter.com/.

Browser	Use percentage
Chrome	46.03%
Internet Explorer	25.87%
Firefox	20.04%
Safari	4.93%
Opera	1.3%
Other	1.84%

Table 10 – Browser popularity on Android based on the number of Installs from Google Play (as of Jul 2014).

Browser	Million installs
Opera Mini	100–500
Chrome Mobile	500–1000
Firefox Mobile	50–100
Opera Mobile	50–100
Android Browser	In all browsers

Table 11 – Default CIF feeds.

http://aper.svn.sourceforge.net/svnroot/aper/phishing_reply_addresses
http://data.phishtank.com/data/online-valid.json.gz
http://malc0de.com/rss
http://mirror3.malwaredomains.com/files/bulk_registrars.zip
http://mirror3.malwaredomains.com/files/domains.zip
http://mirror3.malwaredomains.com/files/url_shorteners.zip
http://reputation.alienvault.com/reputation.data
http://s3.amazonaws.com/alexa-static/top-1m.csv.zip
https://feodotracker.abuse.ch/blocklist/?download=badips
https://feodotracker.abuse.ch/blocklist/?download=domainblocklist
https://feodotracker.abuse.ch/blocklist/?download=ipblocklist
https://spyeyetracker.abuse.ch/blocklist.php?download=domainblocklist
https://spyeyetracker.abuse.ch/blocklist.php?download=ipblocklist
https://spyeyetracker.abuse.ch/monitor.php?rssfeed=binaryurls
https://spyeyetracker.abuse.ch/monitor.php?rssfeed=configurls
https://spyeyetracker.abuse.ch/monitor.php?rssfeed=dropurls
https://zeustracker.abuse.ch/blocklist.php?download=domainblocklist
https://zeustracker.abuse.ch/blocklist.php?download=ipblocklist
https://zeustracker.abuse.ch/monitor.php?urlfeed=binaries
https://zeustracker.abuse.ch/monitor.php?urlfeed=configs
https://zeustracker.abuse.ch/monitor.php?urlfeed=dropzones
http://www.malwaredomainlist.com/updatescsv.php
http://www.mirc.com/servers.ini
http://www.spamhaus.org/drop/drop.lasso
http://www.spamhaus.org/drop/edrop.txt
http://dragonresearchgroup.org/insight/sshpwauth.txt
http://dragonresearchgroup.org/insight/vncprobe.txt
http://www.openbl.org/lists/date_all.txt

Table 12 – Percentage of URLs that were blacklisted.

Browser	Blacklisted	
	Results Q2 2014 (n = 1400)	Results Q1 2014 (n = 5651)
Safari Mobile (iOS)	38.7%	75%
Firefox Mobile (Android)	85.4%	85.3%
Opera Mobile (Android)	75.9%	78.7%
Firefox (Windows)	86.7%	94.9%
Chrome (Windows)	93%	94.5%
Opera (Windows)	77.9%	87.1%
IE (Windows)	48.4%	64.6%

Table 13 – Percentage of false negatives.

Browser	False negatives	
	Results Q2 2014	Results Q1 2014
Safari Mobile (iOS)	26.4%	13.3%
Firefox Mobile (Android)	3.4%	3%
Opera Mobile (Android)	7.9%	1.5%
Firefox (Windows)	5.9%	2%
Chrome (Windows)	1.3%	1.7%
Opera (Windows)	8.4%	1.4%
IE (Windows)	9.9%	6.7%

Table 14 – Percentage of URLs that were manually verified as non-phishing.

Browser	Non-phishing	
	Results Q2 2014	Results Q1 2014
Safari Mobile (iOS)	34.9%	11.5%
Firefox Mobile (Android)	11.1%	11.7%
Opera Mobile (Android)	16.3%	19.8%
Firefox (Windows)	7.3%	3%
Chrome (Windows)	5.7%	3.8%
Opera (Windows)	13.7%	11.5%
IE (Windows)	41.7%	28.7%

Table 15 – Malicious file detection based on the hash of the samples.

AV Engine detection %	% of malware	Cumulative percent	AV Engine detection %	% of malware	Cumulative percent
6	3.6	3.6	44	1	57.1
7	1	4.6	54	0.5	57.7
9	2	6.6	56	1	58.7
10	0.5	7.1	60	0.5	59.2
11	1	8.2	61	0.5	59.7
12	1.5	9.7	63	1	60.7
13	1	10.7	65	1	61.7
14	0.5	11.2	67	2.6	64.3
15	2.6	13.8	68	1	65.3
17	2.6	16.3	69	1.5	66.8
19	1	17.3	70	1	67.9
20	1	18.4	71	0.5	68.4
22	1	19.4	72	0.5	68.9
24	2.6	21.9	73	0.5	69.4
25	0.5	22.4	74	3.1	72.4
26	4.1	26.5	75	1	73.5
27	0.5	27	76	3.1	76.5
28	1	28.1	77	1	77.6
29	0.5	28.6	78	3.1	80.6
30	4.1	32.7	79	2	82.7
31	4.1	36.7	80	1	83.7
32	1	37.8	81	3.6	87.2
33	4.1	41.8	82	1.5	88.8
34	0.5	42.3	83	0.5	89.3
35	1	43.4	85	1.5	90.8
36	0.5	43.9	86	0.5	91.3
37	5.6	49.5	87	4.6	95.9
38	2	51.5	88	0.5	96.4
39	2.6	54.1	89	2	98.5
40	1	55.1	90	1	99.5
43	1	56.1	91	0.5	100

Table 16 – Malicious file detection based on file analysis (submission of the file).

AV Engine detection %	% of malware	Cumulative percent
19	0.4	0.4
25	0.4	0.9
26	0.4	1.3
33	0.4	1.7
34	0.4	2.2
35	0.9	3.1
38	0.4	3.5
43	0.4	3.9
44	0.9	4.8
45	1.3	6.1
46	5.2	11.4
47	13.1	24.5
48	29.3	53.7
49	13.5	67.2
50	9.6	76.9
51	2.2	79
52	4.8	83.8
53	3.1	86.9
54	4.8	91.7
55	0.9	92.6
56	3.9	96.5
57	2.2	98.7
61	0.9	99.6
62	0.4	100

Table 17 – VirusTotal AV engines.

AVG
AVware
Ad-Aware
AegisLab
Agnitum
AhnLab-V3
AntiVir
Antiy-AVL
Avast
Baidu-International
BitDefender
Bkav
ByteHero
CAT-QuickHeal
CMC
ClamAV
Commtouch
Comodo
DrWeb
ESET-NOD32
Emsisoft
F-Prot
F-Secure
Fortinet
GData
Ikarus
Jiangmin
K7AntiVirus
K7GW
Kaspersky
Kingsoft
Malwarebytes
McAfee
McAfee-GW-Edition
MicroWorld-eScan
Microsoft
NANO-Antivirus
Norman
Panda
Qihoo-360
Rising
SUPERAntiSpyware
Sophos
Symantec
Tencent
TheHacker
TotalDefense
TrendMicro
VBA32
VIPRE
ViRobot
Zillya
Zoner
nProtect

Table 18 – VirusTotal URL reputation providers.

ADMINUSLabs
AegisLab
AlienVault
Antiy-AVL
AutoShun
Avira
BitDefender
C-SIRT
CLEAN
CRDF
Comodo
CyberCrime
Dr.Web
ESET
Emsisoft
Fortinet
FraudSense
G-Data
Google
K7AntiVirus
Kaspersky
Malc0de
Malekal
Malware
MalwareDomainList
MalwarePatrol
Malwarebytes
Malwared
Netcraft
OpenPhish
Opera
PalevoTracker
ParetoLogic
Phishtank
Quttera
Rising
SCUMWARE.org
SecureBrain
Sophos
Spam404
SpyEyeTracker
StopBadware
Sucuri
Tencent
ThreatHive
Trustwave
URLQuery
VX
Web
Websense
Webutation
Wepawet
Yandex
ZCloudsec
ZDB
ZeusTracker
malwares.com
zvelo

REFERENCES

Abrams R, Pathak J, Barrera O, Ghimire D. Browser security comparative analysis. NSS Labs; 2014 [Online]. Available: https://www.nsslabs.com/reports/browser-security-comparative-analysis-report-socially-engineered-malware [Accessed: 06.08.14].

Akhawe D, Felt AP. Alice in Warningland: a large-scale field study of browser security warning effectiveness. In: Proc. of the 22nd USENIX Security Symposium; 2013.

Antonakakis M, Perdisci R, Lee W, Vasiloglou II N, Dagon D. Detecting malware domains at the Upper DNS Hierarchy. In: Proc. of the 20th USENIX Conference on Security (SEC'11). Berkeley, CA, USA: USENIX Association; 2011. p. 16.

AV. Anti-phishing protection of popular web browsers. AV Comparatives; Dec 2012 [Online]. Available: http://www.av-comparatives.org/images/docs/avc_phi_browser_201212_en.pdf [Accessed: 05.01.14].

Banu M, Nazreen S, Munawara Banu. A comprehensive study of phishing attacks. Proc. of the International Journal of Computer Science and Information Technologies 2013;4(6):783–6.

Bian R. M., Alice in battlefield: an evaluation of the effectiveness of various UI phishing warnings. [Online]. Available: https://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/termpapers/725mbian13.pdf [Accessed 02.02.14].

Bilge L, Kirda E, Kruegel C, Balduzzi M. EXPOSURE: finding malicious domains using passive DNS analysis. ACM Trans Inf Syst Secur ACM April 2014;16(4). USA.

Bradley, T., Android dominates market share, but Apple makes all the money. [Online]. Available at: http://www.forbes.com/sites/tonybradley/2013/11/15/android-dominates-market-share-but-apple-makes-all-the-money/ [Accessed: 12.04.14].

Caballero J, Grier C, Kreibich C, Paxson V. Measuring pay-per-install: the commoditization of malware distribution. In: Proc. of the 20th USENIX Conference on Security, SEC'11; 2011.

CIF. Collective Intelligence Framework [online]. Available: https://code.google.com/p/collective-intelligence-framework/ [Accessed: 06.08.14].

CISCO, Cisco annual security report. [Online]. Available at: http://www.cisco.com/c/en/us/products/security/annual_security_report.html [Accessed 10.10.14].

Colvin R. SmartScreen application reputation – building reputation. [Online]. Available: http://blogs.msdn.com/b/ie/archive/2011/03/22/smartscreen-174-application-reputation-building-reputation.aspx [Accessed: 04.08.14].

Curtsinger C, Livshits B, Zorn B, Seifert C. ZOZZLE: fast and precise in-browser JavaScript malware detection. In: Proc. of the 20th USENIX Conference on Security (SEC'11). Berkeley, CA, USA: USENIX Association; 2011. p. 33–48.

Darwish A, Bataineh E. Eye tracking analysis of browser security indicators. In: Proc. of Computer Systems and Industrial Informatics Conference; 2012. p. 1–6.

Egelman S, Schechter S. The importance of being Earnest [In security warnings]. In: Proc. of financial cryptography and data security. Springer; 2013. p. 52–9.

Funk C., Garnaeva M. Kaspersky security bulletin 2013. Overall statistics for 2013. [Online]. Available: http://securelist.com/analysis/kaspersky-security-bulletin/58265/kaspersky-security-bulletin-2013-overall-statistics-for-2013/ [Accessed: 04.08.14].

Gartner, Gartner Says annual smartphone sales surpassed sales of feature phones for the first time in 2013, [Online]. Available: https://www.gartner.com/newsroom/id/2665715 [Accessed: 10.08.14].

Gartner, Top 10 strategic technology trends For 2014. [Online]. Available at: http://www.forbes.com/sites/peterhigh/2013/10/14/gartner-top-10-strategic-technology-trends-for-2014/ [Accessed: 02.08.14].

Gartner, Gartner survey highlights top five daily activities on media tablets. [Online]. Available: https://www.gartner.com/newsroom/id/2070515 [Accessed: 10.03.15].

Google, Safe browsing API. [Online]. Available at: https://developers.google.com/safe-browsing/ [Accessed: 08.09.14].

Jansson K, Von Solms R. Phishing for phishing awareness. In: Proc. of Behavior & Information Technology Conference, vol. 32, issue 6; 2013. p. 584–93.

Kirda E, Kruegel C. Protecting users against phishing attacks with antiphish. In: Proc. of Computer Software and Applications Conference, vol. 1; 2005. p. 517–24.

Kolter J, Maloof M. Learning to detect and classify malicious executables in the wild. In: Proc. of the Journal of Machine Learning Research, 7; 2006. p. 2721–44.

Love D. The latest Jailbreak statistics are Jaw-dropping. [Online]. Available: http://www.businessinsider.com/jailbreak-statistics-2013-3 [Accessed: 04.08.14].

Lu M, Leita C, Thonnard O, Keromytis A, Dacier M. An analysis of rogue av campaigns. In: Proc. of the 13th International Conference on Recent Advances in Intrusion Detection, RAID'10; 2010.

Lu L, Yegneswaran V, Porras P, Lee W. Blade: an attack-agnostic approach for preventing drive-by malware infections. In: Proc. of the 17th ACM Conference on Computer and Communications Security, CCS '10; 2010.

Mazher N, Ashraf I, Altaf A. Which web browser work best for detecting phishing. In: Proc. of Information & Communication Technologies Conference; 2013. p. 1–5.

McAfee. McAfee labs threats report. 2014 [Online]. Available: http://www.mcafee.com/mx/resources/reports/rp-quarterly-threat-q1-2014.pdf [Accessed: 12 Mar 2015].

McAfee. Site Advisor [online]. Available: https://www.siteadvisor.com/ [Accessed: 06.08.14].

McAfee. McAfee labs report highlights success of phishing attacks with 80 percent of business users unable to detect scams. 2014 [Online]. Available: http://www.mcafee.com/us/about/news/2014/q3/20140904-01.aspx [Accessed: 10.03.15].

Microsoft, SmartScreen filter. [Online]. Available at: http://windows.microsoft.com/en-us/internet-explorer/products/ie-9/features/smartscreen-filter [Accessed: 08.03.14].

Mozilla, Mozilla support, [Online]. Available: https://support.mozilla.org/en-US/kb/how-does-phishing-and-malware-protection-work#w_how-does-phishing-and-malware-protection-work-in-firefox [Accessed: 20.03.15].

Mylonas A, Tsalis N, Gritzalis D. Evaluating the manageability of web browsers controls. In: Proc. of the 9th International Workshop on Security and Trust Management. UK: Springer (LNCS 8203); 2013. p. 82–98.

Mylonas A, Gritzalis D, Tsoumas B, Apostolopoulos T. A qualitative metrics vector for the awareness of smartphone security users. In: Proc. of the 10th International Conference on Trust, Privacy & Security in Digital Business (TRUSTBUS-2013). Czech Republic: Springer (LNCS 8058); August 2013. p. 173–84.

Mylonas A, Kastania A, Gritzalis D. Delegate the smartphone user? Security awareness in smartphone platforms. Comput Secur May 2013;34:47–66.

Netcraft, Phishing site feed. [Online]. Available at: http://www.netcraft.com/anti-phishing/phishing-site-feed/ [Accessed: 08.03.14].

Nielsen, The digital consumer, The Nielsen Company. [Online]. Available at: http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014%20Reports/the-digital-consumer-report-feb-2014.pdf [Accessed: 02.08.14].

Oberheide J, Cooke E, Jahanian F. CloudAV: N-Version antivirus in the network cloud. In: USENIX Security Symposium; 2008. p. 91–106.

OpenDNS [online]. Available: http://www.opendns.com/ [Accessed: 18.08.14].

Perdisci R, Lanzi A, Lee W. Mcboost: boosting scalability in malware collection and analysis using statistical classication of executables. In: Proc. of the 2008 Annual Computer Security Applications Conference, ACSAC '08; 2008. p. 301–10.

Perdisci R, Lanzi A, Lee W. Classification of packed executables for accurate computer virus detection. Pattern Recognit Lett October 2008;29(14):1941.

Phishtank, Phishtank [online]. Available: https://www.phishtank.com/ [Accessed: 06.08.14].

Provos N, McNamee D, Mavrommatis P, Wang K, Modadugu N. The ghost in the browser analysis of web-based malware. In: Proc. of the 1st Conference on first workshop on hot topics in understanding Botnets, HotBots'07. Berkeley, CA, USA: USENIX Association; 2007. p. 4.

Provos N, Mavrommatis P, Rajab M, Monrose F. All your iframes point to us. In: Proc. of the 17th Conference on Security Symposium, SS'08; 2008.

Rajab MA, Ballard L, Lutz N, Mavrommatis P, Provos N. CAMP: content-agnostic malware protection. In: Proc. of the network and distributed system security symposium (NDSS); 2013.

Rani S, Dubey J. A survey on phishing attacks. In: Proc. of the International Journal of Computer Applications, vol. 88, issue 10; 2014.

Rosiello AP, Kirda E, Kruegel C, Ferrandi F. A layout-similarity-based approach for detecting phishing pages. In: Proc. of Security and Privacy in Communications Networks Workshops; 2007. p. 454–63.

RSA. RSA online fraud report. 2014 [Online]. Available at: http://www.emc.com/collateral/fraud-report/rsa-online-fraud-report-012014.pdf [Accessed: 10.03.15].

Shahzad A, Hussain M, Khan M. Protecting from zero-day malware attacks. In: Proc. of the Middle-East Journal of Scientific Research, vol. 17, no. 4; 2013. p. 455–64.

Sheng S, Wardman B, Warner G, Cranor L, Hong J, Zhang C. An empirical analysis of phishing blacklists. In: Proc. of the 6th Conference on email and anti-spam; 2009.

Shin S, Xu Z, Gu G. EFFORT: efficient and effective bot malware detection. Comput Netw 2012;57(13):2846–50.

Sobrier J., Google safe browsing v2 API: Implementation notes. [Online]. Available: http://www.zscaler.com/research/Google%20Safe%20Browsing%20v2%20API.pdf [Accessed: 10.01.14].

Symantec, 2014 internet security threat report. [Online]. Available at: http://www.symantec.com/security_response/publications/threatreport.jsp [Accessed 10.10.14].

Symantec, Safe web [online]. Available: https://safeweb.norton.com/ [Accessed: 06.08.14].

Vadrevu P, Rahbarinia B, Perdisci R, Li K, Antonakakis M. Measuring and detecting malware downloads in live network traffic. Comput Secur – ESORICS 2013:556–73 [online]. Available at: http://dx.doi.org/10.1007/978-3-642-40203-6_31 [Accessed 19.07.14].

Vidas T, Owusu E, Wang S, Zeng C, Cranor L, Christin N. QRishing: the susceptibility of smartphone users to QR code phishing attacks. In: Proc. of financial cryptography and data security; 2013. p. 52–69.

VirusTotal, VirusTotal [online]. Available: https://www.virustotal.com/ [Accessed: 06.08.14].

Virvilis N, Tsalis N, Mylonas A, Gritzalis D. Mobile devices: a phisher's paradise. In: Proc. of the 11th international conference on security and Cryptography (SECRYPT-2014), Austria; August 2014.

Xu Z, Zhu S. Abusing notification services on smartphones for phishing and spamming. In: Proc. the 6th USENIX Conference on Offensive Technologies; 2012. p. 1–11.

Zhang J, Seifert C, Stokes J, Lee W. Arrow: generating signatures to detect drive-by downloads. In: Proc. of the 20th International Conference on world wide web, WWW '11; 2011.

Zhang H, Liu G, Chow TW, Liu W. Textual and visual content-based anti-phishing: a Bayesian approach. In: Proc. IEEE Transactions on Neural Networks, vol. 22, issue 10; 2011. p. 1532–4.

Zhou Y, Jiang X. Dissecting Android malware: characterization and evolution. In: Proc. of the IEEE Symposium on Security and Privacy. IEEE; 2012. p. 95–109.

Zhou Y, Wang Z, Zhou W, Jiang X. Hey, you, get off of my market: detecting malicious apps in official and alternative Android markets. In: Proc. of the 19th network and distributed system security symposium. USA; 2012.

Dimitris Gritzalis is a Professor (ICT Security) with the Dept. of Computer Science at Athens University of Economics & Business, Greece, where he leads the INFOSEC Laboratory. He is also the Director of the MSc Programme on Information Systems and the Vice-Chair of the University's Research Center. He is the Academic Editor of the Computers & Security Journal.

Nikos Virvilis is a Ph.D. candidate with the Dept. of Computer Science at Athens University of Economics & Business, Greece. He holds a BSc (Informatics) from this University, and an M.Sc. (Information Security) from Royal Holloway, Univ. of London.

Alexios Mylonas is a Lecturer (Digital Forensics) with the Faculty of Computing, Engineering and Sciences at Staffordshire University, United Kingdom.

Nikolaos Tsalis is a Ph.D. candidate with the Dept. of Computer Science at Athens University of Economics & Business, Greece. He holds a BSc (Informatics) from this University, and an M.Sc. (Information Security) from Royal Holloway, Univ. of London.