Betrayed by Your Dashboard: Discovering Malicious Campaigns via Web Analytics

Oleksii Starov
Stony Brook University
ostarov@cs.stonybrook.edu

Yuchen Zhou
Palo Alto Networks, Inc.
yzhou@paloaltonetworks.com

Xiao Zhang
Palo Alto Networks, Inc.
xizhang@paloaltonetworks.com

Najmeh Miramirkhani
Stony Brook University
nmiramirkhani@cs.stonybrook.edu

Nick Nikiforakis
Stony Brook University
nick@cs.stonybrook.edu

ABSTRACT

To better understand the demographics of their visitors and their paths through their websites, the vast majority of modern website owners make use of third-party analytics platforms, such as, Google Analytics and ClickTale. Given that all the clients of a third-party analytics platform report to the same server, the tracking requests need to contain identifiers that allow the analytics server to differentiate between their clients.

In this paper, we analyze the analytics identifiers utilized by eighteen different third-party analytics platforms and show that these identifiers enable the clustering of seemingly unrelated websites as part of a common third-party analytics account (i.e. websites whose analytics are managed by a single person or team). We focus our attention on malicious websites that also utilize third-party web analytics and show that threat analysts can utilize web analytics to both discover previously unknown malicious pages in a threat-agnostic fashion, as well as to cluster malicious websites into campaigns. We build a system for automatically identifying, isolating, and querying analytics identifiers from malicious pages and use it to discover an additional 11K live domains that use analytics associated with malicious pages. We show how our system can be used to improve the coverage of existing blacklists, discover previously unknown phishing campaigns, identify malicious binaries and Android apps, and even aid in attribution of malicious domains with protected WHOIS information.

ACM Reference Format:
Oleksii Starov, Yuchen Zhou, Xiao Zhang, Najmeh Miramirkhani, and Nick Nikiforakis. 2018. Betrayed by Your Dashboard: Discovering Malicious Campaigns via Web Analytics. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3186089

1 INTRODUCTION

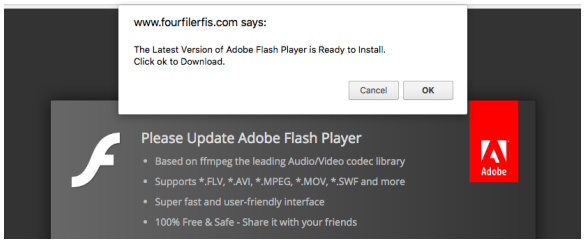
Web analytics is a necessary tool for modern websites to better understand their users and how they interact with their content. Most web developers make use of third-party analytics platforms, such as, Google Analytics, Yandex Metrica, and ClickTale, both because of their ease of adoption as well as the typical presence of no-cost/"freemium" plans. For instance, according to recent statistics from BuiltWith [2], 77.8% of the web's 1 million most popular sites utilize Google Analytics.

Given that all the clients of a third-party analytics platform report to the same centralized backend, the tracking requests emitted from web browsers need to contain identifiers that allow the analytics server to differentiate between their clients. A single identifier (referred to as *ID* throughout this paper) is often shared across different websites that belong to the same account, or to the same project in the analytics dashboard, thus effectively becoming a method of grouping websites together, even among domains that otherwise seem unrelated. As a result, there exist services for reverse lookups of Google Analytics IDs (e.g., SpyOnWeb [15] and SameID [13]) which are used, for example, by journalists to reveal hidden connections between websites [18].

In this paper, we analyze the analytics identifiers utilized by eighteen different third-party analytics platforms and show that these identifiers allow for the clustering of seemingly unrelated websites as part of a common third-party analytics account (i.e. websites whose analytics are managed by a single person or team). We use this observation to perform the first large-scale analysis of analytics utilized by malicious content and quantify the extent to which matching analytics IDs allows for the identification of new malicious content, the clustering of malicious content into campaigns, and even the deanonymization of malicious actors. To that extent, we design and develop a reliable pipeline for parsing sources of malicious content, identifying and extracting IDs associated with the studied analytics services, and searching for new malicious content that shares the extracted IDs in a *threat-agnostic fashion*, i.e., being able to identify malicious content without tailored, abuse-specific detection methods.

We use our system to crawl 145K malicious URLs provided by VirusTotal on a daily basis for a period of two weeks and identify a total of 9,395 unique analytics IDs associated with malicious pages. Our system was able to, on average, discover 1,442 malicious analytics IDs per day, most of which belonging to Google Analytics. Moreover, we extracted 872 analytics identifiers from a two-year corpus of technical support scams and other social-engineering attacks, allowing us to calculate the lifetime of some scam campaigns to more than two years. By searching for domains and URLs reusing the extracted IDs

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
WWW 2018, April 23–27, 2018, Lyon, France
© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.
ACM ISBN 978-1-4503-5639-8.
https://doi.org/10.1145/3178876.3186089




```
http://www.google-analytics.com/__utm.gif?utmwv=5.7.0&utms=3&utmn
=318899286&utmhn=www.fourfilerfis.com&utme=8(Nombre%20landing*
Hash)9(flash%20player%20-%20grey-fp*dnqO3b3R)&utmcs=UTF-8&utmsr
=1440x900&utmvp=1433x372&utmsc=24-bit&...&utmac=UA-41451094...
```

Figure 1: Example of a scam page that calls Google Analytics.

in the wild, we were able to discover 11K additional websites and showed how the sharing of analytic IDs can allow for the deanonymization of owners of domains, even when WHOIS privacy solutions are utilized. Next, we show how our analytics-ID-matching technique applies beyond regular websites (to malicious mobile apps, suspicious extensions, and malware binaries) and how we were able to utilize it to discover 13 phishing campaigns against popular websites. Finally, we explain why evading our detection methods will not be trivial for attackers as long as they find value in analytics, and we describe how analytics companies can utilize their privileged positions to assist in discovering malicious content and aid law-enforcement identify the real culprits behind attacks.

2 BACKGROUND ON WEB ANALYTICS

For virtually all types of web analytics, web developers are asked by the analytics service to embed a piece of JavaScript code throughout their website. This JavaScript code includes logic for tracking user visits and at least one identifier (referred to as *ID* throughout the paper) that is used by the analytics platform to later differentiate between tracking requests of their clients. When visitors load one of the corresponding pages inside their browsers, the analytics script issues requests to the analytics backend which collects tracking data about the current visitor. The analytics services then aggregate the data and make them available through a convenient web dashboard which is made available to website owners.

At this point, it is important to note that while the analytics IDs embedded in websites need to be per-analytics-client unique, they do not need to be per-domain unique. That is, website owners can manage multiple websites as part of a single *project* where, e.g., all the analytics requests for example.com, example.net, and example.org are aggregated together. In this case, the JavaScript code embedded in all three websites would be utilizing the same analytics ID. This allows a third-party observer to infer that these three domains are somehow related (i.e. managed by the same person/team) even when the ID-sharing domains are lexically different, are hosted on different servers, and utilize WHOIS privacy solutions. Through our experiments, we have found that this type of aggregation is very common across both benign and malicious website owners and can therefore be used for clustering seemingly-unrelated websites together into campaigns.

Table 1: Comparison of popular web analytics

Web analytics	Price	Leaked ID	Example ID
Google Analytics	Free	Account	UA-22417551-1
Google Tag Manager	Free	Project	GTM-N7R3KH
New Relic Insight	Paid	Account	9a40653a95
Yandex Metrica	Free	Project	42880164
Quantcast	Free	Account	p-b6_rD1Ba7gEIM
StatCounter	Free	Project	7040321/0/9a83071e
Optimizely	Paid	Project	5328963582
CrazyEgg	Paid	Account	0023/6581
Clicky	Free	Project	101071552
Mixpanel	Free	Account	481d51295e...f5547
Segment	Free	Project	6q5KVhqz...6DONr
Mouseflow	Free	Project	ff5128b8-7...caba8
Chartbeat	Paid	Account	50874
Heap Analytics	Free	Project	429571327
Kissmetrics	Paid	Project	e4756f9bee...c2dc3
ClickTale	Paid	Account	6ea876d3-3...b4f00
Gauges	Paid	Project	58caae4f4b...1c18a
W3Counter	Free	Project	63908

To gauge how well this ID-sharing observation generalizes across different analytics services, we analyze 18 popular services offering general web analytics (listed in Table 1 according to their popularity, as reported by BuiltWith [2]). One can notice that the majority of services provide an option of no-cost subscription, which makes them even more attractive for website owners. Upon signup to any web analytics service, a web developer gets an ability to set up a project, which may or may not require to specify targeted domains. We want to emphasize that even if specific domains are specified in the analytics dashboard, traffic statistics are collected across different origins, and thus the analytics script can be distributed across different websites at the discretion of the web developer. In this case, the analytics request from each website contains the same project ID, which can be used to associate them. Moreover, even if the website owner creates a separate analytics project per each monitored domain, there are services that still require a separate *account identifier*, such as, Google Analytics. Specifically, each Google Analytics account can create up to 100 identifiers of the following format $UA-XX...XX-YY$, where $UA-XX...XX$ is the constant account ID. Similarly, tracking requests to New Relic Insight include a "global license key" which is common across all websites managed by a single account.

The exact format of each service's analytics ID influences the difficulty of correctly identifying other websites that share a given ID. For example, as shown in Table 1, Yandex Metrica uses a highly ambiguous format consisting of a short string of digits. In order to find other websites using the same ID, we need to crawl as many websites as possible and either dynamically locate requests to Yandex backend servers as they are occurring, or statically attempt to locate Yandex-related JavaScript code which may be further complicated through the use of minimization and obfuscation.

Contrastingly, ClickTale utilizes longer strings (e.g., "6ea876 d3-3...f00") while StatCounter uses a combination of the project ID and additional identifier (e.g., "/7040321/0/9a83071e/1/"). In both cases, the resulting IDs are more likely to be globally unique,

and thus can be searched with generic search engines that index the source code of web pages (such as, PublicWWW [11] and NerdyData [5]). Furthermore, the specific format of some analytics providers, such as, Google Analytics and Google Tag Manager, provide us with the ability to not only search for a specific ID, but to retrieve all the identifiers while statically analyzing a page's source code (e.g., all IDs of the format $GTM-XXXXXX$).

Finally, it is worth pointing out that the aforementioned analytics services are not necessarily limited to websites. Browser extensions can straightforwardly utilize web analytics by including the appropriate JavaScript code in their background pages [33] while Android APKs can include analytics SDKs that emit the appropriate HTTP requests that are recognized by the analytics backend servers. In addition to Google Analytics and Google Tag Manager which are available for both websites as well as Android apps, we analyzed the following mobile-specific, analytics services: FireBase, Appflyer, App Metrica, Flurry, Umeng, and Adjust. Google Analytics, FireBase, and Appflyer leak a global analytics account ID with each tracking request, whereas the remaining register unique application IDs for each separate app.

3 DATA COLLECTION AND ANALYSIS

In this section, we describe our pipeline for mining analytics IDs from different sources of malicious URLs, browser extensions, and mobile applications.

3.1 Analytics IDs from malicious websites

For our project we utilize two sources of malicious URLs: i) daily lists of malicious URLs from VirusTotal and, ii) URLs and HTML code of typosquatting domains and the destination URLs of ad-based URL shorteners, kindly provided to us by Miramirkhani et al. [28]. Given these two sources, extracting analytics IDs from malicious websites appears, at first, straightforward. One would need to merely visit each URL, identify the presence of one or more analytics providers, and isolate the utilized analytics IDs. Unfortunately, the following reasons complicate this seemingly straightforward process:

- When visiting a malicious URL some time after it was first reported, the resulting page may now be operated by a domain parking company with its own benign analytics.
- When trusting third-party verdicts about the maliciousness of a given URL, it is unclear which part of the page was malicious, i.e., the main page versus a particular iframe embedded in the page. Malicious pages may include benign content and vice versa, both of which may be utilizing their own web analytics.

To account for these complications during the VirusTotal crawl, we deploy a set of filters, as shown in Figure 2. First, after crawling URLs from VirusTotal, we extract the pairs of valid analytics IDs and actual domain where they were found (i.e. the domain of the main page or that of an iframe). We then check whether that domain was malicious according to VirusTotal and discard those that are reported as benign, allowing us to remove many instances which would otherwise be false positives. Second, we filter out whitelisted domains and known benign analytics IDs from websites in the Alexa top 100K.

During pilot experiments, we discovered that domains resulting from these two filtering steps still contained a large number of false positives. These false positives were mainly due to a few common analytics IDs that were present in large numbers of pages that are part of the lifecycle of a malicious URL but are not malicious in-and-of themselves. For example, the error pages shown by hosting providers when they have suspended a user's account (a common reaction to a malicious URL) may all share the same analytics ID. Thus, if we do not exclude these pages, we would be marking all domains that were suspended/deleted as malicious. To account for such cases, we utilize PublicWWW and SpyOnWeb (two search engines for HTML code) to find other domains with pages that utilize the same analytics ID and ignore a given ID if it is used by more pages than an empirically discovered threshold (500 domains according to our experiments). Our rationale for this threshold is that, if we discover more than 500 unpopular domains all of which share the same analytics ID and some of which are marked as malicious, we consider it more plausible that these are related to a known benign service rather that they are managed by a single dedicated attacker.

Contrastingly, because Miramirkhani et al. [28] provide us with both HTML code as well as URLs, we can develop our own heuristics for identifying a malicious page and if those heuristics match a page that contains an analytics ID, we can immediately isolate and extract that ID. Given the nature of their project and data sources, we search through the HTML and JS corpus for keywords associated with technical support scams, toll-free phone numbers, and messages indicating that we need to download new software (e.g. missing codecs), or update our existing one (e.g. update Flash, Chrome, or Java).

To faithfully mimic a user who lands on a malicious domain, our crawler is based on the headless Chrome Browser. Our crawler is capable of intercepting JavaScript alerts, simulate clicks, and extract analytics identifiers from both network traffic, as well as the page's HTML code and browser DOM. By running our crawler on multiple machines, we are able to crawl and analyze over 10 million domains per day. All network traffic and extracted analytics IDs are stored in database for further analysis.

The statistics described in the remainder of the paper, are based on the following datasets:

- Three daily sets of malicious URLs reported by VirusTotal (VT) in August 2017 and fourteen consecutive VT URL dumps from September 2017. Each set consists on average of 145K unique URLs belonging to 24.3K unique TLD+1 domains. For example, just for the September, we crawled more than two million URLs on 340,873 unique domains.
- The domains from Miramirkhani et al. [28] include almost two years of crawling 10,000 typosquatting domains daily from September, 2015, and a set of 3,000 shortened URLs from top ad-based URL shorteners starting from April, 2016.

Finally, we make use of a commercial URL filtering service belonging to Palo Alto Networks, a network and enterprise security company, which provides its customers with URL categories including: malware, phishing, adult, drugs, and

Table 2: Mining malicious analytics IDs

Web analytics	IDs	Domains	Potential	Verified	Unseen
Google Analytics	7,945	8,182	27,472	10,901	8,132
Yandex	816	912	-	1,364	971
Google Tag Manager	278	289	1,598	683	564
StatCounter	155	144	-	22	20
Clicky	58	68	-	113	83
New Relic Insights	55	107	-	803	779
Quantcast	46	56	336	113	101
CrazyEgg	13	17	-	0	0
Optimizely	11	12	-	4	2
MouseFlow	9	9	-	4	1
Mixpanel	5	5	-	272	272
Segment	2	2	-	0	0
ClickTale	1	1	-	1	0
Heap Analytics	1	1	-	0	0
Overall	9,395	9,226	-	14,267	10,921

new crawls). As before, the recurrent reuse of analytics means that attackers are deploying the same analytics code, across multiple malicious pages.

In contrast with the VirusTotal source, the data provided to us by Miramirkhani et al. [28] is by definition skewed towards social-engineering attacks, particularly of the fake technical support kind. From that data, we were able to extract 872 unique Google Analytics IDs across 3,185 domain names. Most of these IDs (89.2%) were located on technical support scams while the remaining ones were on other types of scam pages, such as, fake surveys and fake plugin updates.

Interestingly, while 51.8% of these analytics IDs were only observed for a single day, there were other IDs that belonged to long-running campaigns (overall, average lifetime was 46 days). For example, using a common Google Analytics ID, we observed a fake-survey campaign that was live for at least 764 days (UA-11040674 seen on 4 captured domains) and a separate fake Flash Player update / fake technical support campaign that was live for at least 730 days (UA-67441257 seen on 44 captured domains). The fact that these campaigns lasted for over two years, suggests not only that these attackers are able to avoid detection for prolonged period of time but also that our proposed method of utilizing analytics IDs to discover campaigns of seemingly unrelated URLs is currently not utilized.

Discovering malicious web campaigns. Having a set of analytics IDs associated with malicious websites allows us to search for the same IDs in the wild and discover previously-unreported malicious websites. Using two code search engines (PublicWWW and SpyOnWeb), we were able to find other known domains for more than 63% of the 9,395 malicious analytics IDs discovered from our VT crawls. Table 2 shows how many potentially-malicious domains we discovered, and how many of them were verified with our own crawlers to still have the matched analytics IDs. We restrict the potential results to analytics providers with sufficiently distinct ID formats (described in Section 2) to ensure that we are really discovering analytics-related identifiers.

Overall, we were able to discover 14,267 live websites containing malicious analytic IDs, 76.5% of which were new, previously unseen domains, i.e., not part of our VT-sourced URLs. As Table 2 shows, for many analytics providers, we are able to at least

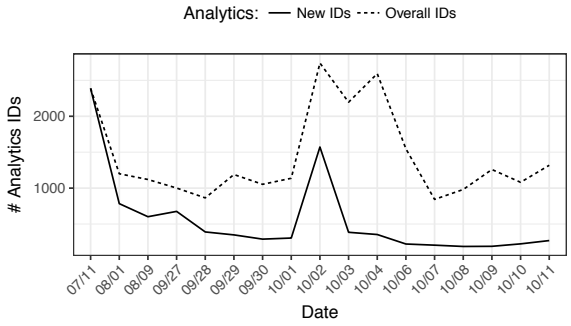


Figure 4: Discovery rate of malicious analytics IDs during the daily crawl of VirusTotal feed.

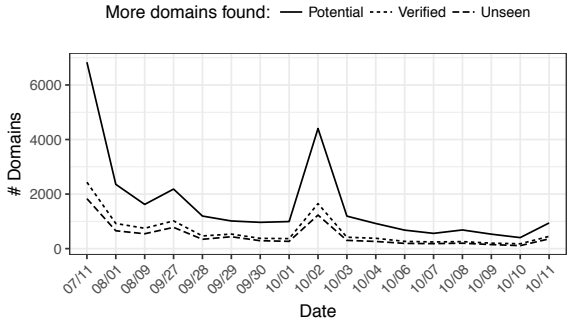


Figure 5: Discovery rate of malicious Google Analytics domains during the daily crawl of VirusTotal feed.

double the number of known live malicious domains from our VT seeds (e.g, we discovered another 8,132 Google analytics domains reusing analytics IDs from the original 8,182 domains), and presumably can at least *triple* the number if we include potentially or formerly malicious websites (e.g. 27K Google analytics domains for the original 8.1K domains). Moreover, by querying VT about our newly-discovered domains, we find that the vast majority of new websites have successfully avoided detection, i.e., only 18.9% of the newly discovered websites are marked as malicious. We argue that this shows the power of this technique since it can associate seemingly benign websites to the same adversaries operating the more explicitly malicious ones.

Figure 5 shows the daily rate on the number of newly discovered domains reusing malicious Google Analytics IDs. We observe the same peak on 10/02 as we did in the daily number of VT-sourced malicious IDs (Figure 4). Using this approach one can expect to, on average, be able to associate 364 new, previously unseen, malicious domains per day, which share analytics IDs with existing malicious domains (considering results starting from 09/27). Similarly, we were able to identify 2,926 other domains (with 2,821 being newly discovered) for 33.6% of the 872 scam-related Google Analytics IDs. Out of those, 836 were still active at the time of this writing with 95% of them being flagged as malicious by VirusTotal scanners. Many examples like error01234567890microsoft.xyz (combosquatting domain [26] utilized for technical support scams) or

Figure 6: Example of "unknown" website detected via UA-81239183. Each click navigates to a different scam page.

Table 3: Intersection with "unknown" websites

Web analytics	# IDs	# Domains
Google Analytics	122	774
Yandex	16	671
Google Tag Manager	7	108
New Reli cInsights	5	26
Clicky	4	3
Optimizely	3	79
CrazyEgg	2	28
MouseFlow	1	72
Mixpanel	1	25
Quantcast	1	1
Overall	162	1785

search-privacy.online (potentially unwanted programs) were discovered to be threats by matching Google Analytics.

Domain and Campaign Forensics. The ability to, given a list of malicious analytics IDs, identify other websites sharing these IDs, allows for a number of powerful forensic applications including: i) estimating the size of malicious campaigns, ii) identifying common attackers behind diverse attacks, and iii) deanonymizing malicious actors. In terms of campaign-size, using the VT-sourced URLs, we were able to find an average campaign size of 7.6 domains, with the largest campaign including 480 domains (note that this is a lower bound on the largest size as we decided to cut-off analytics IDs with more than 500 discovered domains as those may include false-positives). Similarly, for the scam-related dataset, the average campaign size was 3.6 with the largest campaign including 293 domains. By analyzing some of the scam-related data we were able to identify one New Relic Analytics ID ("bfd2c38d50") which was associated with both technical support scams as well as phishing domains. This finding suggests either that one attacker/team is responsible for a wide-range of attacks, or that they have outsourced their analytics to the same entity. In both cases, this identified "link" can be powerful for both expedient take-downs as well as for identifying the culprits by law-enforcement.

Finally, we were able to deanonymize 59 malicious actors behind VirusTotal URLs by finding public WHOIS records for other domains sharing the same analytics IDs. This means that as long as an, otherwise careful attacker who uses WHOIS privacy and fake registration details, links at least one onymous domain to the same analytics account, this can be used to link all of the anonymously-registered domains back to him. For

Table 4: Querying benign and malicious samples

Google Analytics	# Benign	# Malware
Top IDs from VirusTotal URLs		
UA-43126514	137	23,333
UA-69254683	3	605
UA-74694740	22	541
UA-260627	52	520
UA-63404602	0	204
Top IDs from scam pages		
UA-56634126	19	8,609
UA-41451094	16	892
UA-72722497	272	37
UA-82355406	166	26
UA-61108112	0	21

example, Google Analytics ID UA-58907283 was originally found on verification-login.com from VirusTotal, and later was discovered on many phishing pages for Instagram. The same analytics ID was present on ccg.gal which had public WHOIS information, including the registrant's email address. Other examples include deanonymization of suspicious online shops and web pages that distribute suspicious software.

Preventing unknown attacks. To further evaluate our threat-agnostic, analytics-ID-matching, method of finding and correlating malicious websites, we extracted the analytics IDs from the UNKNOWN_URLs dataset (the list of URLs whose categorization was unknown, as described in Section 3.1) and calculated the overlap with our 9,395 malicious analytics IDs. Table 3 shows the number of overlapping IDs across services and the number of previously unlabeled domains that we could associate with known malicious URLs. Figure 6 shows an example of "unknown" website matched via UA-81239183, a Google Analytics from another website known to be malicious by 5 reputable scanners from VirusTotal. Each click on that website navigates the user to a different scam page. Other examples include UA-89467400 found on over 63 "unknown" websites, each leading to a different scam (like harbiturk.xyz or digiz.xyz).

Intersection between web and malware. Knowing that more than 173K malware samples send requests to Google Analytics (Section 3.2), we decided to quantify the possible overlap between the malware IDs and those of malicious websites. Specifically, we calculated the intersection of 7,945 Google Analytics IDs collected from our VirusTotal crawl and the 872 IDs found on scam pages, with the ones present in the HTTP traffic dumps of malicious APKs and malware binaries for different OS. Overall, 29 matched in former case, and 35 in the later. Table 4 presents the most popular of the common IDs.

For example, UA-43126514 is found in the traffic of 23,333 malware samples. The most recent public sample on VirusTotal is detected as "SoftPulse" by 53 out of 64 AV engines. This type of malware exhibits dropper behavior (downloading additional malicious software) and modifying registry entries. The same analytics ID is found on many domains that deliver PUPs in the form of video players (e.g. vidplayer.net and magnoplayer.com). Similarly, UA-56634126 with 8,609 matches in traffic from malware binaries has been spotted on domains that distribute suspicious versions of MacKeeper and other tools to repair Mac computers.

Table 5: Extensions found over intrusive install pages

# Users	# Extensions	Example
1-4MM	29	Search Manager (searchmgr.com)
100K-1MM	88	Movie Search (softorama.com)
10K-100K	91	betterMovies Search (bettersearchtools.com)
0-10K	29	LastLogin Now (lastlog.in)
Unknown	78	Private Search Plus

To better understand the overlap between benign and malicious IDs, we randomly sampled a few of the 137 benign binaries with the UA-43126514 ID and resubmitted them to VirusTotal. There, we saw that most of them were now detected as malware by at least 15 AV engines. This result further strengthens the idea that matching analytics IDs can reveal the true malicious nature of a seemingly benign binary, before that binary is eventually detected as malware by traditional AV engines.

4.2 Analytics from browser extensions

During our crawl of potential scam pages, we were able to collect 333 unique extension IDs. During the manual investigation of some of these extensions, we noticed that some of them were benign, highly popular extensions and therefore, as described in Section 3.2, we filtered out all extensions that were installed by more than 4 million users.

Our filtered list contained 315 extensions served over 11,096 unique URLs hosted on 86 unique TLD+1 domains. Table 5 shows a number of these extensions to allow the reader to develop an intuition of the types of malicious extensions that are offered to users. Across different rankings, we observe extensions which are detected as Potentially Unwanted Programs (PUPs) according to different AV sources (e.g., "safe4search" extension with 5,742 users [4] and "BlpSearch" extensions with 332,610 users [1]). At the time of our analysis we were able to download only 255 extensions, while the remaining ones were no longer hosted on the Chrome Store. Almost half of the collected suspicious extensions (43.5%) belong to the "Search" category which allows them a reasonable cover for requesting full permissions across all tabs and websites of a user's browser.

Out of 255 extensions that we could successfully download and unpack, we found Google Analytics IDs on 120 extensions. Overall, we detected 70 unique Google Analytics accounts which is already evidence of ID sharing across extensions. For example, UA-98374100 is utilized in 14 different Chrome extensions, with installation base ranging from 10K to 162K active users, all developed by a developer called "Better Search Tools." In other cases, we can associate two different extension developers, such as in the case of UA-48154225 used by SearchAssist Tools from searchassist.net (4,221 users) and similarly named extension from privacyassistant.net (4,636 users). By attempting to revisit these extensions two weeks later after our initial crawl, we noticed that both had been deleted from the Chrome Store.

Using the two aforementioned code search engines (PublicWWW and SpyOnWeb) we searched for these 70 extension-originating analytics IDs and found 264 websites which utilized one of these IDs. Among others, we found that UA-101669006 is used on medianetnow.com (associated with the developer of a specific suspicious extension) and on a set of domains that follow the nextlnkN.com format, where N can be substituted with different integers and redirect users to a page requesting the installation of an extension. Our VirusTotal feed exhibits similar results with an average of 54 unique extensions discovered per day, half of which also belong to the "Search" category.

Table 6: Analytics in malware Android APKs

Analytics	# IDs	# Hits	Non Mal.	Benign
App Metrica	12,622	13,445	32.9%	2.4%
Umeng	5,196	92,442	9.6%	0.3%
Google Analytics	551	379	22.7%	10.6%
FireBase	350	136	55.9%	32.4%
Localytics	9	1	100.0%	0.0%
Google Tag Manager	3	0	0.0%	0.0%
Flurry	2	1	0.0%	100.0%
AppsFlyer	1	1	0.0%	0.0%
Overall	18,734	100,379	9.7%	0.2%

4.3 Analytics from malicious Android apps

Overall, from 477,829 malicious APKs, we retrieved 18,734 unique analytics identifiers over 273,232 samples. For example, the Google Analytics ID UA-77544562 is present on 8 malware APKs, labeled as `Android.Trojan.Dropper`. Table 6 shows the distribution of malware-related IDs across popular mobile analytics. Compared to web analytics, Google Analytics is not the most popular choice among malicious actors, taking the third place after by App Metrica and Umeng.

Separately, we tested the detection possibilities enabled by our threat-agnostic, analytics-matching scheme. For that, we collected a testing sample of 330,117 newer APKs from late September 2017. By matching analytics IDs found on previous malicious samples, 100,379 unique APKs were flagged as malware. Of them, 9,775 were classified by Palo Alto Networks' systems as not malicious, however, with 9,332 marked as gray, 289 as unknown, and only 154 received the stronger "benign" verdict. This means that our system can complement existing static/dynamic-analysis malware classifiers and assist in reducing potential false negatives (gray and unknown samples) with low rate of newly introduced false positives (benign samples). Table 6 shows the classification results for all mobile analytics. Among others, we find that Umeng analytics IDs helped to detect the largest fraction of malicious Android apps.

An interesting case was the Google Analytics ID UA-2126908, which was found among many malware APKs, and also on 12 websites related to distribution of cracked mobile apps (like iphonecake.com or directapk.net).

5 DETECTING PHISHING WEBSITES

In this section, we present separate results for phishing and how analytics-ID matching can assist in the quick identification of phishing websites. While crawling URLs from our VirusTotal feeds, we noticed many cases of phishing websites, such as the ones targeting PayPal and LinkedIn users. We were surprised to discover that these types of phishing websites often include the

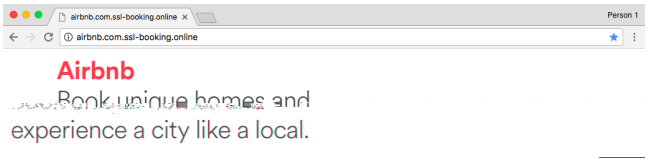


Figure 7: Example of a phishing website that includes original benign analytics ID from Airbnb.

Algorithm 1 Pseudocode for detecting phishing websites

```
target_URLs ← getPotentialTargets(...)
target_IDs ← crawlAnalyticsIDs(target_URLs)
for website in unknown do
    found_IDs ← crawlAnalyticsIDs(website)
    for found_ID in found_IDs do
        if found_ID in target_IDs then
            found_Target ← website(found_ID)
            if not hasDowngradedTLD(website, found_Target) then
                continue
            if not hasLowerRank(website, found_Target) then
                continue
            reportSuspectedPhishing(website)
```

benign analytics IDs of their victim websites and were therefore initially whitelisted by our approach of filtering out websites that utilize analytics IDs present in popular Alexa websites. By investigating the rest of their source code we came to the conclusion that the reason why these phishing websites reuse the benign analytics IDs of their victims is because the software that is used to clone the benign websites, does not remove/substitute the analytics code. Popular phishing frameworks like Social Engineering Toolkit (SET) [14] and Gophish [6] make cloning websites a streamlined process yet they do not account for analytics code.

We can therefore take advantage of this behavior to identify phishing websites by matching the analytics IDs present in unlabeled websites with those of popular websites that are often targeted for phishing. Algorithm 1 shows the high-level steps involved in our approach. We start by monitoring the daily lists of phishing websites from the OpenPhish project [8] which allows us to identify the websites most commonly targeted. We amplify that list by manually adding labels wherever they were missing and adding social networking websites and booking websites to the lists of potential victims. Given our final list of 270 potential targets, we can automatically crawl the benign websites, extract the benign analytics ID associated with each website, and then search for the presence of these IDs in our sources of malicious URLs.

Next, we use additional filters to automatically remove potential false positives. For example, we select only cases with downgraded TLDs (e.g., if the original target is a ".com" website, we match other analytics-ID-sharing websites hosted on ".xyz", ".online", or on a raw IP address). We also filter out popular domains, i.e., ones that appear in Alexa's top 100K, as those that are sharing IDs are most likely managed by the same entities. While false positives do remain, these can be further filtered-out by investigating the IP address space and Autonomous System on which the suspicious website is hosted. We argue, however, that each and every one of these matches is suspicious enough to warrant the attention of a human analyst.

Table 7: Phishing campaigns detected during study

Target domain	Web analytics	# Domains
us.battle.net	GA	12
www.airbnb.com	GA, GTM	10
dailymail.co.uk	GA	4
www.flixster.com	GA, Quantcast	4
serasaexperian.com.br	GA, GTM	2
www.hotwire.com	GTM, Optimizely	2
lonelyplanet.com	GA, GTM	1
made-in-china.com	GA	1
metrobankonline.co.uk	GA, GTM	1
microsoft.com	Optimizely	1
www.bnz.co.nz	GA	1
www.irs.gov	GA, New Relic	1
www.singtel.com	GA	1

After applying Algorithm 1 to the two-week collection of "unknown" websites (UNKNOWN_URLs dataset, described in Section 3.1), we could identify 13 phishing campaigns (e.g., Figure 7). Table 7 lists the discovered phishing targets, associated web analytics, and the number of unique domains used to distribute the phishing attacks. We recorded attacks on the Battle.net gaming portal, Airbnb booking platform, email services, and bank accounts. Usually the attackers used phishing replicas hosted on .xyz, .club, .online domains or links including IP addresses. We also found examples of .com and .ru TLDs to be gateways leading to the final phishing pages, and, interestingly, a replica of the IRS website was found on a .ru domain.

6 DISCUSSION

Summary of findings. Our results from the previous sections clearly indicate that not only do malicious actors utilize analytics in their attacks, but also that they reuse analytics IDs across websites and even across platforms. We showed that using analytics IDs extracted from known malicious websites allows analysts to double the number of malicious websites, group seemingly unrelated websites into campaigns, and deanonymize malicious actors hiding behind WHOIS privacy proxies (Section 4.1). Using the same ID-matching technique, we were able to identify hundreds of intrusive browser extensions in the Chrome Store and cluster extensions together, even when those claimed to be developed by different developers (Section 4.2). Finally, we showed that the same techniques apply to mobile malware (Section 4.3) and we uncovered a design error of modern website-cloning tools that enables the detection of phishing websites by the mere fact that they reuse the analytics IDs of their victims (Section 5).

Analytics providers. Given our findings, we see many avenues where existing web analytics providers can assist in the identification of malicious websites and in the attribution of these websites back to unscrupulous individuals. The analytics providers that we investigated in this paper are clearly in the position, given an analytics ID, to identify all the websites associated with this ID and potentially all other websites managed

by the same account. We argue that this information, together with details about the owner of a given analytics account, would be invaluable for law-enforcement purposes. Moreover, analytics platforms can help website owners in identifying active phishing websites, by warning them about the existence of a new domain that is sharing an analytics ID with their current domain and exhibits suspicious behavior.

Attacker adaptation. Even though attackers can, as a result of this work, change their modus operandi for launching attacks and utilizing web analytics, a change that is effective for evasion purposes is harder than it might first appear. For example, even though some attackers may start utilizing their own web analytics platforms (backed by software such as OWA [7] and Piwik [10]), these analytics backends will clearly only be utilized by malicious websites and can therefore become signals of website maliciousness, similar to Indicators of Compromise (IOCs) present in benign but compromised websites [22]. Alternatively, if they identify lesser known analytics platforms that offer stronger privacy guarantees, they would still be standing out, assuming that the vast majority of benign websites keeps utilizing the popular analytics platforms investigated in this paper. Even then, these analytics platforms could still assist law-enforcement in identifying the operators of malicious websites. Finally, even though website-cloning software can be modified so that it does not clone the victim analytics ID, the ID present on the benign website can be bound to the visual representation of that website and form a stronger website identity. Phishing-discovery tools can use the *absence* of such an ID from websites that are visually similar to popular phishing targets, as an extra signal for identifying new phishing attacks.

Method generalization. In this work we report on the effectiveness of associating analytics IDs for detection and discovery of malicious websites and other malware. At the same time, we argue that the developed method can be generalized to support other artifacts of the modern web and mobile applications, which tend to be shared and are likely to be used by malicious actors. Examples include payment addresses, affiliation identifiers, generated code snippets with license tokens, accounts for different widgets and services. As with analytics IDs, we expect similar challenges in extracting the identifiers, reducing false-positives, and evaluating the effectiveness in order to assign proper threat scores to the discovered matches.

7 RELATED WORK

The motivation to this work came from an article by Lawrence Alexander about discovering hidden connections of websites via Google Analytics IDs [17]. Specifically, Alexander used shared Google Analytics IDs to reveal pro-Kremlin web campaigns [18]. At the same time, there already exist services for reverse Google Analytics lookups, such as, SpyOnWeb [15], SameID [13], domainIQ [3], and RiskIQ [12]. However, to the best of our knowledge, we are the first to generalize the ID-sharing problem to many analytics services and perform a large-scale analysis of the applicability of this technique for identifying malicious websites, clustering malicious content, and performing cross-platform attribution.

In general, the detection of malicious websites by inspecting HTTP requests and responses is a known approach, e.g., Kosba et al. [27] created ADAM, a system that evaluates network metadata by rendering web pages in a sandbox. A cross-layer detection model was developed by Xu et al. [35], considering both network and application level features. Drew et al. [24] investigated the HTML similarities of replicated criminal websites and Cova et al. [23] analyzed the phishing websites created by "free" phishing kits. Invernizzi et al. [25] proposed the idea of discovering more malicious pages by leveraging the crawling infrastructure of third-party search engines, which is conceptually similar to our method of discovering other domains using the same analytics IDs. Catakoglu et al. [22] showed that it is possible to use high-interaction honeypots to automatically extract Indicators of Compromise that can be then used to identify compromised websites in the wild. Even though our method is focused on identifying malicious infrastructure, it could also, in principle, be used to identify compromised websites where attackers injected their own analytics IDs. Similarly, there has been substantial work in the behavioral analysis and classification of HTTP-based malware [29, 31, 32], mainly focusing on the network traces between malware installations and attacker-controlled servers. In addition, Aresu et al. [19] research the clustering of Android malware based on HTTP traffic, and Zheng et al. [36] propose a signature scheme for associating Android malware.

Most of the recent antiphishing research is based on crowd-sourced solutions like PhishTank [9] and OpenPhish [8], detecting visual similarities [34], detecting suspicious URLs [21], and proposing machine learning models [16, 37]. Morever, some studies investigate source code features [20] and anomalies in HTTP transactions [30], but do not consider the presence or absence of analytics ID as a feature.

8 CONCLUSION

In this paper, we investigated the design of 18 third-party analytics services and the reuse of analytics IDs across websites. Focusing on malicious sites, we showed that attackers share analytics IDs across URLs and even across platforms. We developed a pipeline for efficiently and accurately isolating and extracting analytics IDs from malicious websites, extensions, binaries, and mobile apps and showed that using our system we can discover tens of thousands of new malicious URLs and perform attribution of malicious domains even when they utilize WHOIS privacy protection services. Finally, we described how we can take advantage of an oversight of website-cloning tools for identifying phishing campaigns in the wild and discussed how analytics services can take advantage of their already-collected data to aid in the identification of malicious websites and the individuals behind them.

Acknowledgments: We thank the reviewers for their valuable feedback. This research was supported by the Office of Naval Research (ONR) under grants N00014-16-1-2264 and N00014-17-1-2541 as well as the National Science Foundation under grants CNS-1617902, CNS-1617593, and CNS-1527086. Some of our experiments were conducted with equipment purchased through NSF CISE Research Infrastructure Grant No. 1405641.

REFERENCES

[1] 2017. Anti-Malware Zone: BlpSearch, Logiciel Potentiellement Indesirable. (2017). https://nicolascoolman.eu/2017/09/09/pup-optional-blpsearch/.

[2] 2017. BuiltWith Technology Lookup. (2017). https://builtwith.com.

[3] 2017. DomainIQ: Reverse Analytics. (2017). https://www.domainiq.com/reverse_analytics.

[4] 2017. How to remove Safe4Search redirect (Virus Removal Guide). (2017). https://malwaretips.com/blogs/remove-safe4search/.

[5] 2017. NerdyData: Search Engine For Source Code. (2017). https://nerdydata.com.

[6] 2017. Open-Source Phishing Framework. (2017). https://getgophish.com/.

[7] 2017. Open Web Analytics (OWA). (2017). http://www.openwebanalytics.com.

[8] 2017. OpenPhish: Phishing Intelligence. (2017). https://openphish.com/.

[9] 2017. PhishTank: Join the fight against phishing. (2017). https://www.phishtank.com/.

[10] 2017. PIWIK: Open Analytics Platform. (2017). https://piwik.org.

[11] 2017. PublicWWW: Search Engine for Source Code. (2017). https://publicwww.com.

[12] 2017. RiskIQ: PassiveTotal Threat Investigation Platform. (2017). https://www.riskiq.com.

[13] 2017. SameID.net: Cut through hours of keyword research in seconds. (2017). http://sameid.net.

[14] 2017. The Social-Engineer Toolkit (SET). (2017). https://github.com/trustedsec/social-engineer-toolkit.

[15] 2017. SpyOnWeb Research Tool: Internet Competitive Intelligence. (2017). http://spyonweb.com.

[16] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. 2007. A Comparison of Machine Learning Techniques for Phishing Detection. In *Proceedings of the Anti-phishing Working Groups 2Nd Annual eCrime Researchers Summit (eCrime '07)*. ACM, New York, NY, USA, 60–69. https://doi.org/10.1145/1299015.1299021

[17] Lawrence Alexander. 2015. Bellingcat: Unveiling Hidden Connections with Google Analytics IDs. (2015). https://www.bellingcat.com/resources/how-tos/2015/07/23/unveiling-hidden-connections-with-google-analytics-ids/.

[18] Lawrence Alexander. 2015. Open-Source Information Reveals Pro-Kremlin Web Campaign. (2015). https://globalvoices.org/2015/07/13/open-source-information-reveals-pro-kremlin-web-campaign/.

[19] Marco Aresu, Davide Ariu, Mansour Ahmadi, Davide Maiorca, and Giorgio Giacinto. 2015. Clustering Android Malware Families by Http Traffic. In *Proceedings of the 2015 10th International Conference on Malicious and Unwanted Software (MALWARE) (MALWARE '15)*. IEEE Computer Society, Washington, DC, USA, 128–135. https://doi.org/10.1109/MALWARE.2015.7413693

[20] Suresh Babu.K. 2013. Phishing Websites Detection Based on Web Source Code and Url in the Webpage.

[21] Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. 2010. Lexical Feature Based Phishing URL Detection Using Online Learning. In *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security (AISec '10)*. ACM, New York, NY, USA, 54–60. https://doi.org/10.1145/1866423.1866434

[22] Onur Catakoglu, Marco Balduzzi, and Davide Balzarotti. 2016. Automatic Extraction of Indicators of Compromise for Web Applications. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*. 333–343.

[23] Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2008. There Is No Free Phish: An Analysis of "Free" and Live Phishing Kits. *WOOT* 8 (2008), 1–8.

[24] Jake Drew and Tyler Moore. 2014. Automatic Identification of Replicated Criminal Websites Using Combined Clustering. In *International Workshop on Cyber Crime (IWCC), IEEE Security and Privacy Workshops*. IEEE. http://lyle.smu.edu/~tylerm/iwcc14.pdf

[25] Luca Invernizzi and Paolo Milani Comparetti. 2012. EvilSeed: A Guided Approach to Finding Malicious Web Pages. In *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*. 428–442. https://doi.org/10.1109/SP.2012.33

[26] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gomez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*.

[27] Ahmed E. Kosba, Aziz Mohaisen, Andrew West, Trevor Tonn, and Huy Kang Kim. 2015. *ADAM: Automated Detection and Attribution of Malicious Webpages*. Springer International Publishing, Cham, 3–16. https://doi.org/10.1007/978-3-319-15087-1_1

[28] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. 2017. Dial One for Scam: A Large-Scale Analysis of Technical Support Scams. In *Proceedings of the 24th Network and Distributed System Security Symposium (NDSS)*.

[29] Saeed Nari and Ali A. Ghorbani. 2013. Automated Malware Classification Based on Network Behavior. In *Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC) (ICNC '13)*. IEEE Computer Society, Washington, DC, USA, 642–647. https://doi.org/10.1109/ICCNC.2013.6504162

[30] Ying Pan and Xuhua Ding. 2006. Anomaly Based Web Phishing Page Detection. In *Proceedings of the 22Nd Annual Computer Security Applications Conference (ACSAC '06)*. IEEE Computer Society, Washington, DC, USA, 381–392. https://doi.org/10.1109/ACSAC.2006.13

[31] Roberto Perdisci, Davide Ariu, and Giorgio Giacinto. 2013. Scalable Fine-grained Behavioral Clustering of HTTP-based Malware. *Comput. Netw.* 57, 2 (Feb. 2013), 487–500. https://doi.org/10.1016/j.comnet.2012.06.022

[32] Roberto Perdisci, Wenke Lee, and Nick Feamster. 2010. Behavioral Clustering of HTTP-based Malware and Signature Generation SignaMalicious Network Traces. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10)*. USENIX Association, Berkeley, CA, USA, 26–26. http://dl.acm.org/citation.cfm?id=1855711.1855737

[33] Oleksii Starov and Nick Nikiforakis. 2017. Extended Tracking Powers: Measuring the Privacy Diffusion Enabled by Browser Extensions. In *Proceedings of the 26th International World Wide Web Conference (WWW)*.

[34] Liu Wenyin, Guanglin Huang, Liu Xiaoyue, Zhang Min, and Xiaotie Deng. 2005. Detection of Phishing Webpages Based on Visual Similarity. In *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web (WWW '05)*. ACM, New York, NY, USA, 1060–1061. https://doi.org/10.1145/1062745.1062868

[35] Li Xu, Zhenxin Zhan, Shouhuai Xu, and Keying Ye. 2013. Cross-layer Detection of Malicious Websites. In *Proceedings of the Third ACM Conference on Data and Application Security and Privacy (CODASPY '13)*. ACM, New York, NY, USA, 141–152. https://doi.org/10.1145/2435349.2435366

[36] Min Zheng, Mingshen Sun, and John C. S. Lui. 2013. Droid Analytics: A Signature Based Analytic System to Collect, Extract, Analyze and Associate Android Malware. In *Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TRUSTCOM '13)*. IEEE Computer Society, Washington, DC, USA, 163–171. https://doi.org/10.1109/TrustCom.2013.25

[37] Weiwei Zhuang, Qingshan Jiang, and Tengke Xiong. 2012. An Intelligent Anti-phishing Strategy Model for Phishing Website Detection. In *32nd International Conference on Distributed Computing Systems Workshops (ICDCS 2012 Workshops), Macau, China, June 18-21, 2012*. 51–56. https://doi.org/10.1109/ICDCSW.2012.66