q-logger skimmer keeps Magecart attacks going

By Threat Intelligence Team

Published: 2021-10-18 · Archived: 2026-04-05 18:04:03 UTC

This blog post was authored by Jérôme Segura

Although global e-commerce is continuing to grow rapidly, it seems as though Magecart attacks via digital skimmers have not followed the same trend. This is certainly true if we only look at recent newsworthy attacks; indeed when a victim is a large business or popular brand we typically are more likely to remember it.

From a research standpoint, we have observed certain shifts in the scope of attacks. For instance, the different threat actors are continuing to expand and diversify their methods and infrastructure. In a blog post about Magecart Group 8, we documented some of the various web properties used to serve skimmers and exfiltrate stolen data.

But at the end of the day, we only know about attacks that we can see, that is until we discover more. Case in point, one particular skimmer identified as q-logger, has been active for several months. But it wasn't until we started digging further that we realized how much bigger it was.

Q-logger origins

This skimmer was originally flagged by Eric Brandel as q-logger. Depending on how much you enjoy parsing JavaScript you may have a love/hate relationship with it. The code is dense and using an obfuscator that is as generic as can be, making identification using signatures challenging.

This skimmer can be found loaded directly into compromised e-commerce sites. However, in the majority of cases we found it loaded externally.

The loader

The loader is also an encoded piece of JavaScript that is somewhat obscure. It is injected inline within the DOM right before the *text/x-magento-init* tag or separated by copious amounts of white space.

One way to understand what the code does is by using a debugger and setting a breakpoint at a particular spot. It is best to either use an already compromised site or bypass the check for the address bar (onestepcheckout).

We can now see the purpose of this script: it is to load the proper skimmer.

The skimmer

As mentioned previously, the skimmer is quite opaque and makes debugging effort difficult and lengthy.

To cut to the chase, the skimmer exfiltrates data via a POST request to the same domain name where the JavaScript is loaded from.

Threat actor and victims

We were able to collect a few indicators from the threat actor behind this campaign. One was the use of netmail.tk, also observed by Luke Leal, for registering skimmer domains.

Although there are clusters of domains from the same registrant, we see that they are trying to compartmentalize their infrastructure and hide the hosting provider's true IP address. They also register domains en masse, which allows them to defeat traditional blocklists.

We don't have a good estimate of how prevalent this campaign is, but we certainly run into it regularly while monitoring e-commerce sites for malicious code. The victims are various small businesses with an online shop running Magento.

Conclusion

The large number of e-commerce sites that are running outdated versions of their CMS is a low hanging fruit for threat actors interested in stealing credit card data. In a sense, there is always a baseline of potential victims that can be harvested.

And every now and again, some opportunities appear. They could be as simple as a zero-day in a plugin or CMS, or maybe an entry point into more valuable targets via a supply-chain attack.

Threat actors are always ready to pounce on those and may well have established their infrastructure ahead of time, waiting for such opportunities.

Malwarebytes customers are protected against this skimmer.

Indicators of Compromise

Email addresses (registrant)

- wxugvvvu@netmail[.]tk
- isgskpys@netmail[.]tk
- zulhqmnr@netmail[.]tk
- yzzljjkmc@emlhub[.]com
- foyiy11183@macosnine[.]com

Skimmer domains

Skimmer URLs

YARA rules
