Media Contacts

Mark Meissner	Melinda Reinicker
PCI Security Standards Council	RH-ISAC
+1-202-744-8557	202-507-5710
press@pcisecuritystandards.org	pr@rhisac.org
Twitter @PCISSC	Twitter @RH_ISAC

August 1, 2019

THE THREAT OF ONLINE SKIMMING TO PAYMENT SECURITY

The PCI Security Standards Council and the Retail & Hospitality ISAC **(https://rhisac.org/)** want to highlight an emerging threat that requires urgent awareness and attention.

What is the threat?

A growing threat that all merchants and service providers should be aware of is *Web-based or Online Skimming*. These attacks infect e-commerce websites with malicious code, known as *sniffers or JavaScript (JS) sniffers and are* very difficult to detect. Once a website is infected, payment card information is “skimmed” during a transaction without the merchant or consumer being aware that the information has been compromised.

A term sometimes used in the press for this threat is Magecart. *Magecart* is an umbrella term used by some security researchers to describe several criminal hacking groups who are responsible for various online skimming attacks. The term has also been used to generally identify the type of attack being utilized by the groups. These attacks have been active since 2015 and represent the continuously evolving cyber threat behind several high-profile attacks against international organizations.

How do these attacks work?

These threat actors use various methods, which include exploiting vulnerable plugins, brute force login attempts (credential stuffing), phishing and other social engineering techniques. All in an attempt to gain access and inject malicious code. These attacks are either directly into e-commerce websites or often into a third-party’s software libraries that merchants rely upon. These service providers may not be aware of the risk they create for their customers if they are not focused on security and the potential threats targeting them.

Examples of these attacks to third-party applications and services include advertising scripts, live chat functions, and customer rating features. Once compromised, these third-party services are used by attackers to inject malicious JavaScript into the target websites. Because these third-party functions are typically used by multiple e-commerce sites, the compromise of one of these functions can allow an attacker to compromise many websites at the same time through mass distribution of the malicious JavaScript.

The code is often triggered when a victim submits their payment information during checkout. Different threat actors gather different details including, billing address, name, email, phone number, credit card details, username, and password. The malicious code logs the payment data either locally on the compromised website or remotely to a computer controlled by the threat actors.

Who is most at risk?

Any e-commerce implementation that does not have effective security controls in place is potentially vulnerable. Attacks target e-commerce websites, third-party service providers, and companies providing applications used on websites. Magecart hackers and similar threat actors are continuing to evolve and modify their attacks, including customizing malicious code for different targets, and exploiting vulnerabilities in unpatched website software.

Additionally, the threat is persistent. One in five Magecart-infected stores are re-infected within days, according to a report by security researcher Willem de Groot.[i] For that reason, it is crucial that affected systems be cleaned and that underlying vulnerabilities be patched or mitigated. If an underlying vulnerability is not addressed, or if some of the attacker's code remains on the system, it could lead to reinfection.

What are some DETECTION best practices?

The ability to detect these threats before they can cause damage is significantly important. Examples of PCI DSS Requirements providing "detection" controls include:

- Reviewing code in order to identify potential coding vulnerabilities (Req. 6)
- Use of vulnerability security assessment tools to test web applications for vulnerabilities (Req. 6)
- Audit logging and reviewing logs and security events for all system components to identify anomalies or suspicious activity (Req. 10)
- Use of file-integrity monitoring or change-detection software (Req. 11)
- Performing internal and external network vulnerability scans (Req. 11)
- Performing period penetration testing to identify security weaknesses (Req. 11)

Alerting on posts to newly observed domains in proxy logs can further provide additional avenues of detection for future phishing attacks as well as the initial reconnaissance phases of an attack on a third-party JavaScript library.

What are some PREVENTION best practices?

The best protection to mitigate against these attacks is to adopt a layered defense that includes patching operating systems and software with the latest security updates. Some examples from PCI DSS include:

- Disable unnecessary ports/services/functions and configure components securely in accordance with industry accepted system hardening standards (Req. 2)
- Implement malware protection and keep up to date (Req. 5)
- Apply security patches for all software (Req. 6)
- Follow secure coding practices and perform code reviews (Req. 6)
- Restrict access to only what is absolutely needed and deny all other access by default (Req. 7)
- Use strong authentication for all access to system components (Req. 8)
- Implement intrusion-detection and/or intrusion-prevention to detect and prevent intrusions (Req. 11)
- Conduct proper due diligence prior to engagement of third-party service providers and monitor service providers' PCI DSS compliance status (Req. 12)
- Additional controls for hosting service providers to protect their customers' hosted environments and data (Appendix A1)

Third party services and products should be reviewed to identify the impact on the organization's PCI DSS scope. It is recommended that organizations prohibit external assets on pages that accept cardholder data, as it extends the cardholder data environment scope to any environment hosting those assets. Customer contact portal vendors are an example of third-party service providers that should be reviewed as part of the organization's scope. Removing or disabling unnecessary plug-ins and services is also recommended (PCI DSS Req. 2). It is important to ensure that any third-party scripts that are present in other areas of the website cannot gain access to payment pages or other sensitive areas (PCI DSS Req. 2 and Req. 6).

Securing of third-party infrastructure and restricting access and permissions of third-party scripts to only trusted sources is also essential. Merchants should clearly identify the specific PCI DSS requirements covered by the service provider, and any requirements that are the responsibility of the service provider's customers to implement. Organizations should actively monitor for and block attempts to introduce malicious code on to their e-commerce space. Permitting externally hosted JavaScript or Cascading Style Sheets (CSS) on payment acceptance pages should not be allowed. Where feasible, transitioning from using third-party hosted scripts to using internally hosted copies of third-party scripts could significantly decrease the risk of malicious modification. Third party scripts should be monitored to detect changes and the changes be reviewed to identify any potentially malicious code before implementation. Using content security policies (CSP) to restrict compliant browsers from executing JavaScript from sources which have not been explicitly whitelisted is also an added protection that should be incorporated.

Organizations should perform due diligence on third-party service providers and use only trusted software vendors. Choose software vendors that build security into their software products and provide ongoing support for security updates throughout the software lifecycle. Service providers should be committed to providing secure services that do not introduce risk to their customers' e-commerce environments.

Organizations should implement Multi-Factor Authentication (MFA) for all access to their e-commerce systems as well as to systems providing support or administrative functions (PCI DSS Requirement 8.3). In doing so, organizations can prevent the likelihood of these repositories from being accessed by malicious threat actors and better secure their infrastructure. Some simple mitigations include protecting all hosts in a web path with effective antivirus, controlling outbound traffic from all systems in a web path and ensuring that developers are not using third party code repositories.

###

[i] Catalin Cimpanu, one in five Magecart-infected stores get reinfected within days, November 15, 2018 -- 06:30 GMT, https://www.zdnet.com/article/one-in-five-magecart-infected-stores-get-reinfected-within-days/