What Developers MUST know about PCI DSS 4.0

In Plain English!

wizer-training.com

Original

PCI DSS 4.0 Requirement 6.2.1

Bespoke and custom software are developed securely, as follows:

- **Based on industry standards and/or best practices for secure development.**
- **In accordance with PCI DSS (for example, secure authentication and logging).**
- **Incorporating consideration of information security issues during each stage of the software development lifecycle.**

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.2.1

It basically means that your dev team needs to focus on how they're going to keep PCI Data, like credit card info, safe within the app. This means nailing down the basics, like getting authentication and authorization spot on, encrypting the PCI data properly, and making sure none of that sensitive info slips into debug logs. Now, this isn't just a last-minute thing. They've got to bake this security mindset in right from the start, from the design phase, and keep it up all the way through to when the app goes live.

Original

PCI DSS 4.0 Requirement 6.2.2

Software development personnel working on bespoke and custom software are trained at least once every 12 months as follows:

- **On software security relevant to their job function and development languages.**
- **Including secure software design and secure coding techniques.**
- **Including, if security testing tools are used, how to use the tools for detecting vulnerabilities in software.**

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.2.2

This one is simple – Developers need to be trained at least once a year. Wizer Training has an amazing solution for that :) Even if your devs are top-tier, it doesn't mean they're wired to think like a hacker. They need to learn how to secure their code, understand the hacker mindset, and get familiar with the tools out there that can detect vulnerabilities in their code and put their fixes to the test.

Original

PCI DSS 4.0 Requirement 6.2.3

Bespoke and custom software is reviewed prior to being released into production or to customers, to identify and correct potential coding vulnerabilities, as follows:

- ***Code reviews ensure code is developed according to secure coding guidelines.***
- ***Code reviews look for both existing and emerging software vulnerabilities.***
- ***Appropriate corrections are implemented prior to release.***

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.2.3

This means that before you roll out your app or a new update, someone needs to review the code, making sure it's secure. Who's gonna do the review? Definitely not the person who wrote it – nope! You need either another pro who's skilled at spotting security flaws or use automated tools, like scanners that are designed to scan for vulnerabilities in the code. This is an iterative process, if issues were found they need to be fixed and reviewed all over again.

Original

PCI DSS 4.0 Requirement 6.2.4

Software engineering techniques or other methods are defined and in use by software development personnel to prevent or mitigate common software attacks and related vulnerabilities in bespoke and custom software, including but not limited to the following:

• Injection attacks, including SQL, LDAP, XPath, or other command, parameter, object, fault, or injection-type flaws.

• Attacks on data and data structures, including attempts to manipulate buffers, pointers, input data, or shared data.

• Attacks on cryptography usage, including attempts to exploit weak, insecure, or inappropriate cryptographic implementations, algorithms, cipher suites, or modes of operation.

• Attacks on business logic, including attempts to abuse or bypass application features and functionalities through the manipulation of APIs, communication protocols and channels, client side functionality, or other system/application functions and resources. This includes cross-site scripting (XSS) and cross-site request forgery (CSRF).

• Attacks on access control mechanisms, including attempts to bypass or abuse identification, authentication, or authorization mechanisms, or attempts to exploit weaknesses in the implementation of such mechanisms.

• Attacks via any "high-risk" vulnerabilities identified in the vulnerability identification process, as defined in Requirement 6.3.1.

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.2.4

The requirement specifies the types of attacks to provide clear guidance, moving beyond a mere "to the best of my knowledge" approach. All these types of attacks and how to defend against them should be explained during the developers' secure coding training. Additionally, as outlined in Requirement 6.3.1, your code should be reviewed for these types of vulnerabilities before launching your app or releasing a new update.

Original

PCI DSS 4.0 Requirement 6.3.1

Security vulnerabilities are identified and managed as follows:

- *New security vulnerabilities are identified using industry-recognized sources for security vulnerability information, including alerts from international and national computer emergency response teams (CERTs).*
- *Vulnerabilities are assigned a risk ranking based on industry best practices and consideration of potential impact.*
- *Risk rankings identify, at a minimum, all vulnerabilities considered to be a high-risk or critical to the environment.*
- *Vulnerabilities for bespoke and custom, and third-party software (for example operating systems and databases) are covered.*

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.3.1

Not all vulnerabilities are created equal. Some are high-risk, others low-risk, and it's important to label them accordingly, or at least identify the high-risk ones. You need a process to handle them effectively. This includes documentation, assigning responsibility, prioritizing tasks and making sure that are fixed. If you discover a high-severity vulnerability affecting customer data, you may even need to activate your incident response team to check for exploitation. Additionally, your dev team should stay informed about vulnerabilities in any third-party libraries or products they use to ensure timely patching.

Original

PCI DSS 4.0 Requirement 6.3.2

An inventory of bespoke and custom software, and third-party software components incorporated into bespoke and custom software is maintained to facilitate vulnerability and patch management.

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.3.2

Your dev team probably uses third-party libraries or software, whether free or paid. If these libraries have security issues, it could impact your app. First, it's crucial to know which libraries your team is using and to list them. Second, you need to continuously receive notifications from third-party vendors or websites that track vulnerabilities, to stay informed about any security issues in these libraries.

Original

PCI DSS 4.0 Requirement 6.3.3

All system components are protected from known vulnerabilities by installing applicable security patches/updates as follows:

- *Critical or high-security patches/updates (identified according to the risk ranking process at Requirement 6.3.1) are installed within one month of release.*
- *All other applicable security patches/updates are installed within an appropriate time frame as determined by the entity (for example, within three months of release).*

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.3.3

This requirement focuses on patching third-party libraries/software used by your dev team. In Requirement 6.3.1, we discussed labeling vulnerabilities by risk. Having done that, you should prioritize patching any high-severity vulnerabilities within a month and address lower-severity ones later, for example within three months.

Original

PCI DSS 4.0 Requirement 6.4.1

For public-facing web applications, new threats and vulnerabilities are addressed on an ongoing basis and these applications are protected against known attacks as follows:

Reviewing public-facing web applications via manual or automated application vulnerability security assessment tools or methods as follows:
– At least once every 12 months and after significant changes.
– By an entity that specializes in application security.
– Including, at a minimum, all common software attacks in Requirement 6.2.4.
– All vulnerabilities are ranked in accordance with requirement 6.3.1.
– All vulnerabilities are corrected.
– The application is re-evaluated after the corrections

OR

Installing an automated technical solution(s) that continually detects and prevents web-based attacks as follows:
– Installed in front of public-facing web applications to detect and prevent web based attacks.
– Actively running and up to date as applicable.
– Generating audit logs.
– Configured to either block web-based attacks or generate an alert that is immediately investigated.

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.4.1

If your application is a web-based solution, like a website or Software as a Service, you need to conduct a thorough check at least once a year or every time you have a major release. You should either find a professional in application security to do this or use automated tools that are designed for the task. Regardless of the method you choose, you need to test for the requirements listed in Requirement 6.2.4, and if you find any vulnerabilities, they should be fixed based on the priorities listed in Requirement 6.3.1.

Another option you might consider is a solution like a Web Application Firewall that was designed to constantly monitor the traffic and detect attacks attempts on your web app. You can set it up to either block potential threats outright or send alerts so someone can jump in and investigate immediately. Just keep in mind that whatever solution you are using it needs to be able to detect that attacks listen in requirement 6.2.4

Original

PCI DSS 4.0 Requirement 6.4.2

For public-facing web applications, an automated technical solution is deployed that continually detects and prevents web-based attacks, with at least the following:

- *Is installed in front of public-facing web applications and is configured to detect and prevent web-based attacks.*
- *Actively running and up to date as applicable.*
- *Generating audit logs.*
- *Configured to either block web-based attacks or generate an alert that is immediately investigated.*

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.4.2

This is very similar to the previous requirement 6.4.1 and will replace it eventually. It focuses on the automatic tools like Web Application Firewall. You need to ensure that it is configured correctly, always up-to-date, and that there is someone who constantly reviews the logs. Additionally, whenever there is an alert, it should be investigated immediately.

Original

PCI DSS 4.0 Requirement 6.4.3

All payment page scripts that are loaded and executed in the consumer's browser are managed as follows:

- ***A method is implemented to confirm that each script is authorized.***
- ***A method is implemented to assure the integrity of each script.***
- ***An inventory of all scripts is maintained with written justification as to why each is necessary.***

#StayWizer

In Plain English

PCI DSS 4.0 Requirement 6.4.3

If your website has payment pages, such as a checkout page or cart, etc, you need to ensure that an attacker cannot manipulate these pages by including malicious scripts. If they succeed, they could capture the credit card information entered by the user and steal it. This concept is similar to the need for protecting physical devices used for swiping cards. There are various ways to safeguard these pages, such as using a Content Security Policy (CSP) to ensure that scripts can only be loaded from authorized locations. Part of the secure code training involves teaching developers how to implement these protections.