APPLICATION SECURITY (HTTPS://BLOG.CONVISOAPPSEC.COM/EN/CATEGORY/APLICATION-SECURITY-EN/)

06/04/2022

Developers: How to deal with some of the biggest security challenges during software development

By Gabriel Galdino (https://blog.convisoappsec.com/en/author/gabogaldino/)

Share

It has become common for people to choose not to leave home and do banking, shopping, food ordering and many other tasks through online applications. For these activities, people create a “login” with a unique password, confirm its authenticity through a captcha, and then feel an almost permanent sense of comfort and “security.”

You can also listen to this article:

09:54 1.0x

12 plays (https://vooozer.com/pt/home/?

However, we realized that these measures are often not enough to protect users from possible software vulnerabilities and, in addition, we identified that developers are on the front line of defense against possible threats, since they are the ones in charge to write the application code.

It is undeniable that security is crucial in all parts and processes of creating and maintaining an application. The evolution of systems requires cybersecurity to increasingly rely on the help of developers to solve the most pressing problems in the industry.

Knowing this, we have prepared a review of some of the main challenges, as well as some tips for developers, in order to make it easier to resolve these issues.

7 Security Challenges During Software Development

Identify vulnerabilities in code

Each programming language and framework has its own specific vulnerabilities, which can affect the security of the application as a whole.

Examples: SQL injection and XSS attacks.

SQL injection attacks allow an attacker to read or modify a database by sending SQL code through an unfiltered user input mechanism. On the other hand, an XSS attack allows attackers to run scripts on users’ browsers, redirecting them to malicious websites or stealing information through cookies.

In programming treadmills, it is very common to integrate different codes through commits. At this stage, a good practice is to perform a Code-Review to verify that the code is well written, following the market standards and, also, that the code is efficient, scalable and secure.

Therefore, it is important to be aware of the risks of the specific vulnerabilities that the programming language presents, as the lack of knowledge about security can result in the inclusion of exploitable features without knowing it.

Keeping the company and its staff aware of the specific problems of the programming language used helps to reduce security risks in its applications.

Recognize vulnerabilities in third-party components and dependencies

Another problem among the teams is the vulnerabilities of third-party components and dependencies. Usually, when building their applications, development teams install different external packages for different purposes, but not all of them are secure or are frequently updated.

Depending on the structure of your company, it may be up to your team or the security team to fix any vulnerabilities found in these packages.

Trying to manage this manually can become a never-ending task, depending on the size of the application. Imagine the hassle of manually comparing the latest security bulletins with dependencies in your code.

There are many services that focus on checking codebase dependencies, finding common security vulnerabilities and exposures (CVEs), making it possible to automate this step, giving the developer more time to fix the remaining security issues, focusing on the quality of the code.

Realize the risks of using open source components

Most applications use open source components, be they libraries, frameworks or individual code snippets. Because they are pre-built, these components can help reduce development time, but they also tend to introduce vulnerabilities to the application.

Part of the open source components are used in the form of “black box”, that is, developers know the functionality of the code, but not necessarily know how its logic works exactly.

Other problems can arise if developers do not know where their components are downloaded from or constantly validate their version numbers. This can increase the potential for application insecurity, due to possible insecure open source code and also to inconsistencies. in the correction and application of version control.

Protect and Manage Sensitive Data – DataOps

Knowing the security practices around data management and DataOps has also been a growing market requirement for the developer.

It is important to note that managing confidential data goes beyond data security. With this, it is necessary to capture the data lineage to show by who, when, where and how the data was changed. Manual processes cannot solve this problem. All this must happen through secure data integration and management platforms.

DataSecOps is an approach to automating data privacy and security by integrating privacy, security, and governance into automated workflows that run alongside data development, deployment, and analytics operations.

Therefore, centralizing identity management, defining role-based permissions, and masking sensitive data in development environments are important data security and privacy practices that a developer needs to keep in mind during development.

Defend the importance of secure software development (AppSec) for the Company

In the daily life of a developer, one of the biggest difficulties faced is to reconcile secure development with short and inflexible deadlines, often imposed by product owners, project managers or other types of company stakeholders.

In this way, it is necessary to defend the importance of AppSec, that is, security in development, highlighting the risks of developing insecure applications for the other instances of the company. Finding and fixing vulnerabilities during development and testing is more efficient and less expensive than doing it later in the process, when an application is already in production.

A concept that has become popular in the developer world is “Shift-Left”, which means making security part of software development from the conception and design phase, through the entire development process to production.

Spending a few extra hours on development, prioritizing security, can save a lot of time and expense in the future, when we weigh the consequences of a vulnerability exploited by a malicious actor.

Encourage AppSec planning

In the course of these topics, it became clear how important AppSec’s planning and organizational culture is for the company’s entire development team.

Security-related activities are usually performed only as part of the test phase, at the end of the SDLC (Software Development Life Cycle), generating rework for the developer and costs for the business. Can you imagine building an entire code to discover only at the end that it has a basic security flaw?

In addition,very often development teams leave application protection solely to the security team. However, the only way to increase security is to allow them to also be self-sufficient in terms of application protection.

This type of planning isn’t just about adding a lot of tools. It’s also important to understand which tools are actually needed and to encourage an AppSec culture throughout the process.

If you’re the leader of your team, start by creating clear guidelines about who is responsible for each security task.

- Do threat modeling (https://blog.convisoappsec.com/o-que-e-modelagem-de-ameacas/)during your application planning, that is, before you even start coding.
- Build a safety training practice and education programs on the topic.
- As part of your plan, include procedures with traceability of the progress of identifying and resolving the vulnerabilities found, using, for example, playbooks or manuals to standardize the steps to be followed depending on the identified scenario.

By following these steps, you ensure that issues are not overlooked and further simplifies future fixes. The key is to synchronize security throughout the development and planning process.

The importance of implementing a security culture

As technology advances, people are becoming more and more reliant on applications to perform daily tasks. This trust results in a growing awareness of security, and thus a growing customer demand for products that provide safe and reliable services.

Review the list, consider which items are most applicable to your organization, and then start making necessary adjustments to improve your safety culture.

Also, talk to the developers on your team and practice empathy, understanding their main difficulties. You’re likely to find really talented developers who know security and can be groomed to act as Security Champions (https://blog.convisoappsec.com/security-champions-voce-precisa-pensar-sobre-isso/).

The purpose of getting internal help from developers is to leverage them as force multipliers for this cultural shift. By adopting some of these practices, you can mitigate future vulnerabilities in your software. With this, you can rest easy knowing that you and your customers are safer.

Share

Related posts