The importance of secure software development

by Juan Pablo Bello - Wednesday, November 02, 2016

http://blog.belatrixsf.com/the-importance-of-secure-software-development/

A few weeks ago I attended a course in secure software development by SANS, one of the worldwide leaders in information security. I want to share with you some of my thoughts on developing secure software, and the importance of having security principles running through the whole software development lifecycle, not just with regard to testing your final application.

Unfortunately it is still the case today that many software developers lack expertise in security best practices, and don't have the knowledge of how they can best architect and develop software based on core security principles. In the worst case scenarios that we heard during the course, some developers believe that security is either someone else's responsibility, or that they will not be affected because they think other people will not be interested in the information they are handling. But in many cases it is much worse: the developers and QA engineers do not even know what the risks are. So there is no way for them to build secure software.

What this means is that the software product or application that you are developing, and which is critical for your business success, may not adhere to best practices of information security including data integrity, availability, and confidentiality. Such is the importance of software in today's' business processes, that a simple error can end up resulting in millions of dollars of loses.

Not fully understanding how you can integrate security principles throughout the software lifecycle can result in your organization inadvertently releasing private customer data, or having to implement expensive updates to the software once it has already gone to market (often this is due to ignorance of the risks that exist). And this can have major business implications, from a loss of customer trust to a public relations nightmare. There are many examples of attacks against well known firms:

- In 2011 Sony Pictures suffered a simple SQL Injection attack by LulzSec (a hacktivist group), which released around 1 million user accounts, including passwords, email addresses, home addresses, dates of birth, and all Sony opt-in data associated with their accounts. What was worse is that every bit of data they took wasn't encrypted. Sony stored over 1,000,000 passwords of its customers in plaintext, which means it was just a matter of taking it. This was insecure.
- In the same year Citigroup was another victim. Hackers who stole bank account details for 200,000 Citigroup customers infiltrated the company's system by exploiting a garden-variety security hole, known as Insecure Direct Object Reference, in the company's website for credit card users.
- And Apple has also not been left behind. In 2010 a security breach exposed iPad owners including dozens of CEOs, military officials, and top politicians. The attack in this case was known as Failure to Restrict URL access, which allowed the attackers to steal subscribers' email addresses, coupled with an associated ID used to authenticate the subscriber on AT&T's network, known as the ICC-ID (ICC-ID stands for integrated circuit card identifier and is used to identify the SIM cards that associate a mobile device with a particular subscriber), just iterating this ICC-ID in an

unprotected URL.
- Meanwhile, in 2015 Uber accidently revealed the personal information of hundreds of its US drivers; and even security firms themselves are not immune from making errors, as Let's Encrypt found out.

So, as you can see all of these companies are big, well-known firms, and you would imagine they would never let themselves open to attack. Well, they were. Now imagine other companies with less resources which need to enforce security.

So we need to change our mindset and accept that security risks will always be present. We need to understand that it is our responsibility to manage and mitigate them.

However, given the different nature of the various systems we build, there is no manual to tell us exactly what to do to mitigate the risks that exist for each of them. But there are guides to help us find the correct way to implement security best practices for each of them. So the first step is to change the way we work. We need to change our “Software Development Lifecycle” to a “Secure Software Development Lifecycle”, where we have detailed checklists to follow and integrate security best practices into every step.

When working with this different framework (the Secure Software Development Lifecycle) we primary introduce security in different ways in each well-known step of the software development lifecycle after having specific security training based on our client’s needs.

I recommend reviewing and learning from the excellent guides provided by OWASP (Open Web Application Security Project). One of the most important is its “Top 10 Most Critical Web Application Security Risks”. Another one is CAPEC (Common Attack Pattern Enumeration and Classification), where an extensive list of different attack patterns are described, giving you information about the scenarios where an attack is possible and how to prevent or mitigate the impact of the attack.

Some high level best practices for integrating security principles include:

1. Understand the sensitivity of the information you will be handling in the system and classify it.
2. Take trainings on application security, including endpoint security, network security, and content security.
3. Build security controls in each access point based on needs.
4. Test each security control.

As software engineers who are responsible for building systems that handle sensitive information, we need to stop thinking that security is someone else's responsibility, that we are exempt from risks. And most importantly, we need to stop thinking that software security is a different career. It needs to be present in every product we develop. So we need to internalize and understand the risks that exist and how to mitigate them. Software security is an essential part of our job that completes us as professionals.
