YGN ETHICAL HACKER GROUP

Whitepaper

Causes of Security Flaws 101

By

d0ubl3_h3lix

Mon Jan 07 2008

Table of Contents

Abstract

This paper will address various reasons or facts on why security flaws come up and why even old-school flaws still exist. The paper is totally intended for web developers and project managers. It will be updated whenever possible.

Causes 101

- Being lack of Baseline Security Knowledge

Generally developers see security as a thing that is not mostly concerned with them or their life. They may assume security as the sole thing of security professionals. My dear friend 'Ko Min Thu (http://flashband.net)' told me once that he did not take interest in security. He failed to neglect my security advisories. For 80% developers, if their servers get hacked, their first line of defense is to replace hacked pages with good backups rather than fixing security bugs in code. Surely they have no idea on how to fix. Even worse, they do not approach security guys for help or something. There are dozens of places on the web (including from us) that you can ask for free security help. I suggest every IT professional should have foundation security knowledge comparable to Security+ certification not to make old-school vulnerabilities again and again. If not, our IT systems are always vulnerable to dozens of attacks – that will be the universal truth.

- Being Laziness or Too Confident

Developers are sometimes lazy to implement appropriate security measures to protect their applications. To be fully security-compatible, they know a lot of hardwork has to be put. They neglect to do security when applications are only for in-house/intranet realm. They falsely assume Intranet is safe from attacks because of Firewalls+IDS/IPS protections.

Some smart developers use security/vulnerability scanners to scan their applications. Such testing is a blackbox security testing approach. One inherent weakness of such scanners is that they look for exact vulnerable string patterns. If they do not find buggy patterns, they simply tell you that your application is 100% safe from attacks. Or even worse, they can alert you with false signs of vulnerabilities.

As every day advanced hacking techniques are developed and used to defeat security mechanisms, you cannot assume that your applications are safe forever even after thorough penetration testings by great white hackers or security scanners checking.

- Facing Too-Tight Deadlines

As developers have to do multiple projects at the same time, they are forced to finish projects in deadlines. Project managers almost always measure Quality Products with functionality but not with security. They are not to blame. Software Engineering methods say that a product is successful if it meets users' requirements. No Actual Security Testing. Thus, they should study and research developments such as Secure Software Developments, Security Engineering, Security Testings and Methodologies …etc.

- Using Security-Buggy Reusable Components

As developers fail to do security testing and make their code security bug free, their reusable libraries or components have similar flaws. Subsequently all of their applications have similar flaws.

Read the following scenario.

XYZ web development firm is well-known for their amazing web designs, and functionalities. They are proud to include their logo texts (something like Design by XYZ) on every application they have developed. Unfortunately they fail to secure their applications. Bad guys can know their applications have flaws. Simply using Google, blackhats can dig XYZ's application developments history. Using same techniques and tools, they can exploit all of XYZ's applications within a few minutes. What an easy hacking, uh?

- Using Third-Party Plug-ins or Applications

To certain extent, commercial third-party plug-ins or applications are safe. Even if attacks occurred due to those, you can claim compensation for loss from them. Yes, it is like insurance. But you need to contract with them for any emergency cases.

Open-source or freeware third-party plug-ins or applications are not trusted except widely-used-popular ones because such widely-used applications may have relatively free of (some) bugs because of testing by world-wide community. Unknown third-party plug-ins or applications pose possible risks to your applications if you use them without thorough testing of both security and functionality.

Bad guys can host significantly-great-functioning plug-ins in freeware hostings like hotscripts.com, sf.net …etc. Their wares may secretly steal files or run shell codes on your servers. Or even if they are not bad guys, they may be immature programmers who do not even know how to write secure programs.