Software Security in Software Development

Software Security: Lecture 3

Muona, Leo

April 9, 2014

1 Summary

In lecture 3 we had a guest lecture Henri Lindberg from Nsense to discuss about reality of software security in software development. We also discussed about software security cerficates that are trained and sold by multiple companies, and how a vast number of them can be utterly stupid. Finally we learned about software security lifecycle and different security standards that are worth looking at. All in all, software security is done by having attitude to do the right things, and demonstrate corporation leadership why security is good and what is needed for it.

2 Reality today

Software industry has been going downhill in quality, because systems that are delivered are developed with "just get it shipped" mentality and they in most cases lack development’s security budget. Focus is on delivering features, not making secure systems, or even testing if the system that is being delivered is secure. Outsourcing of development is done to decrease costs and that causes dropping of quality for short-term profits, which is not good in the long run. In reality, when things have gone bad, the security portion may go like this:

- Requirements are a line that says "Software has to be secure".

- Outsourcing partner does something, and now project is 6-18 months late and budget has been exceeded several times.

- In verification stage it is verified that the application does not work.

- After the release a vulnerability assessment is done from six months to three years after development, maybe.

As the trend of today’s software development, agile development has been praised. However, there are problems with some people that develop as agile: they might use agile as an excuse to go wild during development, make all solutions ad-hoc and just try to get features to be as they were specified, and later altered. Because of this, security processes are often bypassed and this causes security risks. Thus, it would be important for the developers to understand that agile development does not mean the same as "develop without following correct processes".
3 Security certificates

Security certificates can be a selling point in most software deals or when hiring new developers. But there are few issues what comes to these certificates. Because in reality almost any company could state that they are a software security company, and can create enterprise styled security certificates, like for example ASS (Application Security Specialist Certified), there is no quality over these.

Most of security certificates does not mean that a person has a high skill level in security: they are usually obtained automatically via couple days enterprise courses or online ”click the correct answer” exams. In most cases, a company or an employee is going to spend minimum amount of time, money and dedication in these certificates, thus making them only have a label ”we know software security” when the reality can be quite different.

4 How do you do secure software development?

During the lecture, we listed the following tasks for software development:

- Requirements finding
- Design
- Coding
- Quality assurance (testing)
- Project management
- Documentation
- Shipping (deployment)
- User training
- Operations
- Maintenance
- Customer management
- Decommissioning (end of life)
- Sales and marketing
- Risk management

From these tasks we made notes on what security related tasks are included in them.

When requirements are defined, software security requirements must be done as well. However, this means that we have to require bad things not to happen. In order to do this, abuse cases and attacker stories must be written and then translate them into test cases, security features, maybe hiring a security specialist, and requiring to comply with a security guideline.

For design, we talked about threat modlelling which leads to architectural risk analysis. This means software’s architecture is designed to be secure. For coding, tools like static analysis tools and linters are used. Static analysis tools analyze the source code statically and linters are such tools that try to find bugs in the source code. There are few issues that these tools have: they produce large amounts of false positives, and cannot find all errors, thus they don’t replace code reviews or other processes. Quality assurance (testing) includes blackbox testing with fuzzers and other black box testing tools, and whitebox testing with unit tests and integration tests.

There were also discussion of other tools that weren’t part of software development itself, but are still used to track certain vulnerabilities. These tools include antivirus softwares, firewalls and
wrapping (i.e. EMET) or sandbox tools that run the program and try to notice if anyone is trying to hack it.

It has to be noted as well that software security in software development lifecycle depends on the development lifecycle itself: Agile development has different security lifecycle compared to i.e. waterfall development. For agile development, security must be done as procedures that are done during sprints or certain time laps during continuous integration. Security specifications can also change in the middle of the project, thus project management should consult security specialists to see if everything is in order. For waterfall development, security are done as it was listed above. That is, things are done in chronological order that is specified and when product has been shipped, it can be just maintained if new vulnerabilities are later found.

5 Software security standards

During the lecture there was mentions about software security standards that can be used inside a company to standardize secure software development. First one that was mentioned was Microsoft’s security SDL lifecycle (http://www.microsoft.com/security/sdl/default.aspx). Though it was mentioned during class, that because this is a large company standard, it is consulted by Microsoft and thus is quite expensive.

In comparison there were few cheaper options, like The Software Security Framework (SFF) by BSIMMS (http://bsimms.com/online/). There is also OWASP Software Assurance Maturity Model (http://www.opensamm.org/). Both of these are free online documents and thus may not require expensive consultant support.

There is an ISO standard for software security (ISO-27034-1). It was mentioned that it’s processes can be quite similar to what Microsoft is consulting with SDL.

Also, there is one documentation for Finnish public sector: ”VAHTI - Valtionhallinnon tietoturvalisuuden johtoryhmän tuottama ohjeistus.” It might be worth checking out if you are fluent in Finnish. These are security instruction documents for buying ICT equipment, developing software, and notes for office, personel and workstation security.