Security in Software Project

Notes of Lecture 3

Author:
Kati Kyllönen

Version 2.0
1. Introduction

Lecture 3 [1] discussed how software projects are taking software security into account today and what could be the more ideal way to include security aspect to software applications. Visitor lecturer Henri Lindberg gave a short overview what practical challenges he has faced in security line of business. Henri works in a company called nSense [2] that offers security assessments and consultancy services in the field of software security. In the latter part of the lecture, course lecturer Antti Vähä-Sipilä introduced several ideas and models how the software security could be embedded into a software process.

2. Software security business today

Henri started by telling that people who enter the software security business are often very enthusiastic and want to use their skills and knowledge to change the world. Too often these people are greatly disappointed when facing the reality. Reality might contain software systems which have totally wrong architecture or implementation. It is also common that projects have a zero budget to correct security issues and what Henri considers even worse is those people that do not care at all. One speciality that Software business has, are cases that the specification is made correctly but for some reason the software vendor has not followed the specification. To soften newbies collision to reality Henri want’s to give us an overview what is the Software Security Business today.

Next Henri introduced us The Microsoft Security Development Lifecycle model. The model was not examined carefully, instead Henri indicated that this is one model that helps companies in creating secure software. The following image will give an overview of the Microsoft Security Development Lifecycle [3].

![Image 1: Microsoft Security Development Lifecycle](image)

The model pushes the security mindset into early phases of software process. This is the same signal that Antti has been introducing us during the whole ongoing course. Security is not an add-on, but needs to be taken into account from the very beginning of software project.
This model consists 7 phases that together introduce a set of 16 valid security practices. These practices are visible in the following image [3].


This model not only introduces these practices, but also suggest a correct timing for the activities. Microsoft provides a freely available training site that provides further details [4].

Henri’s view of the current reality was quite sceptic. In his opinion the Software security requirements, if made at all, are usually vague. He also indicated that controlling the security aspect in outsourced software is difficult. SW Projects are usually running late on schedule and exceeding their budgets. Henri told that during his career he has never seen a project that has finalized prior deadline. In his opinion the verification outcome is too often that the application does not work, not only from the security aspect but also the business use cases are not functioning. And according Henri’s experiences the vulnerability assessment is made only years after the Software has been deployed, if at all.

When the vulnerability assessment is made at the end of a project does not usually lead to fully secure software as every correction might introduce new vulnerabilities and the problem might originate from the architecture or from design of the software.

![Vulnerability assessment cycle](Image 3: Vulnerability assessment cycle)
Compared to situation of not doing the assessment at all, doing assessment at the end of course improves the situation. But to truly make a secure software requires that the security aspect is taken into account from the very beginning. Henri mentioned that process people do not usually have big fans in the software industry. This makes introducing new ways of working a challenging task. As an example Henri provided the agile-boom, where big software enterprises tried to take the agile practices into use, but too often ended up of having only ad-hoc processes. Henri demands that this must change, processes are there to help you.

In order to improve the current ways of working a change in mindset is needed! Vulnerabilities are relatively easy to find. Problem is that people (in charge) usually believe the severity of the problem only after the vulnerability is exploited. It’s a pity, because using work hours to demonstrate an exploit does not improve a software security. These work hours or money could have been used instead to make a correction based on the information that is available right after the vulnerability is found. Henri reminds that security work is much about tiny details: “Every decision has a consequence and technical decisions may have non-technical outcomes.”

As a last issue Henri told that the offering of security certificates is very broad. What the certificate tells about it’s holders know-how in security or product’s security level is fuzzy. The tricky point is that once you have been certified, the certificate rules prohibits you from criticizing that certificate. This way the information of really nonsense certificates is not spread. Henri asked us to take a careful approach towards certificates. One can get the flavour what he means by comparing these certificates:

Certified ASS (http://www.asscert.com/)
Common criteria (http://www.commoncriteriaportal.org/)
CISSP (https://www.isc2.org/CISSP/Default.aspx)
Hackerproof-Vulnerability Scanning (http://www.comodo.com/hackerproof/what_hackerproof_includes.html)

Henri ended his presentation by emphasizing that attitude (yours and theirs) and leadership within software company have the most impact on how security is taken into account in their products. But no matter how difficult it gets, please, never stop caring!
3. How to bring security aspect to software process

Antti started his presentation by asking a question: “Software project, what phases does it include?” Query to audience generated following phases that are illustrated in the following image.

![Image 4: Phases in software development](image)

But where is the security work? Where should the security be added? Today we will be concentrating how to add security to design and coding phases. But if a thorough security project needs to be run, each and every of these phases needs to be touched with a security aspect.

Most software companies do not invest into security. These companies might not even have a single security person. Having a separate security personnel is not always a meaningful goal, it’s perfectly fine if the development personnel takes care of security aspect if they take the job seriously. Microsoft’s security budget is on correct level - so at least one company in the world has! Microsoft delivers their security knowledge by writing books and creating material and tools that are available in their extranet. Microsoft’s Security Development Lifecycle model gives one example how to embed security work into a process [4].

There are other models though. The BSIMM-V Software Security Framework (SSF) [5] was done by interviewing approximately 70 companies on how they do security work in their sw
projects. These operations were unified to 12 practices and 112 activities that are presented in BSIMM-V web pages under Creative Commons License. Image 5 gives present’s the 12 practices.

![Image 5: The Software Security Framework of BSIMM-V](image5.png)

It’s good to understand that the SSF is influenced by financial sector and large companies. So it might not fit as such to a small startup company doing agile. The SSF is really valuable source of information because it is only public source of information how companies actually do their security work. If somebody happens to know of another similar source, Antti requested to be informed about it. By looking at these practices and the activities, it is easy to see how do they link to the phases present in sw development.

There is another a lot similar model that is OWASP Software Assurance Maturity Model called OpenSAMM [6]. The similarity comes from the fact that these used to be a same model, but the work was separated at some point of time.

![Image 6: The OpenSAMM Software Assurance Maturity Model of OWASP](image6.png)

All these three models, Microsoft SDL, BSIMM SSF and OWASP OPENSAMM are great sources of information if one needs to figure out how to bring the security aspect into the software process.
There is also a ISO standardization work ongoing to create a 27000 series standard of software security [7]. The first part ISO 27034-1 is already published and it defines what software security is and how should it be done. First part contains also an example that looks very much like Microsoft’s SDL. First part of this standard does not yet cover much, but as the standardization work proceeds the next parts will cover more details. Second part is possibly published within next 2 years. This standard is chargeable, so you need to pay a fee to get access to it. Antti has already experienced that some companies require compliance with the ISO 27034, so it might become a default requirement within next years.

It’s also good to know that there is application security guideline for Finnish public sector organizations, called VAHTI (1/2013) [8]. It is probable that this will become requirement within next years in public sector software purchases.

There is also a PA-DSS which is a Payment Application Data Security Standard that must be followed when dealing with payment cards [9]. All applications that deal with credit card numbers needs to be compliant with this standard [1].

Next Antti asked us the following questions: “What would you do in the requirement finding phase to define security?” and “How could you define sw security requirements?” The audience was quite uncertain how this could be done, but after Antti’s steering we came to conclusion that following types of requirements could be defined.

Image 7: Security requirements as feature requirements or as negative requirements

When there is a clear requirement like “a user needs to be authenticated” it becomes quite obvious to feed a requirement of this to sw project’s requirement management system. After having a requirement, most processes take care that each requirement will trigger an action in the test planning. But it is good to remember that one should also define negative requirements, that is, describe what should not happen. After these requirements are logged in to the requirement management these will also generate tests. This way security seeds are planted into a sw-process. In addition to this you could also make requirement that there are test cases that comply with the security coding principles or guideline. It is also a good idea to define requirement of the need of a security specialists engagement in a project. Antti
told that if he would ever see all this happening in a real software project he would be very pleased!

How do these requirements map to phases of sw project? In order to get things done you need to make a mapping. This mapping is always a company specific and depends greatly of the ways of working as well as what process is used. In the following image there is one mapping.

Test cases that were required map naturally to testing phase. And security feature to design and coding phases. The involvement of a security specialist is most valuable if that person is available in the design phase, because mistakes made at this phase become very expensive. How to then to define if the security guideline has been met? If you can turn the security guideline into a checklist, there is also high probability that you can turn the checklist into a test case or do a static analysis that tells how do we comply with security guideline. So far we have actually not discussed of static analysis at all in detail. We will come back to that after a short while. Just be patient!

How is security then translated into actions in the design phase? Those security features that were defined as requirements are of course designed in the design phase. There can be features like cryptography, enciphering and authentication that are security work that is done during a design phase. If there is a security specialist in the project then it’s important that already design is made secure. One security action in design phase is Threat modelling or Architectural analysis. Threat modelling is according to Antti the single most important activity to bring security into a software. We are not going to talk more of threat modelling because it will be covered in next two lessons.
How about security activities in coding phase? We have in earlier lessons gone through already some things that can be done in coding phase. I assume that Antti was here referring to prevention mechanisms of XSS, CSRF and SQL injections as well as tainting. But what we have not discussed is a concept of static analysis. A static analysis is actually a machine based code review. Antti reminded that he told about the tainting method where the code is gone through and all suspicious parts are marked as tainted. The static analysis tools are working in a bit similar manner, but instead of running the code they look or more like do an intelligence investigation the code. An example that Antti presented was a function that compares a variable value to greater than zero and smaller than zero, but the code is missing if the variable is equal to zero. For example this kind of incidents would be flagged by a static analysis tool.

There is a lot of variety in the static analysis tools. Some tools investigate an intermediate code. There are tools like the Findbugs that analyses a Java code. The Coverity checks errors in C and it has also support for Java, but what it actually does it uses Findbugs to this. Because the analysis needs to be language specific, tools that offer support to several languages usually first compile the code into an intermediate language and then they run the analysis on it. There are also static analysis tools that look at the binary itself. The Veracode in one of those tools and it is actually quite prominent. Antti uses a linter himself when he is coding, he claims that it increases his productivity because it colours the mistakes he has made. This way he notices the mistakes and can correct those without making a compilation which would in most cases terminate with an error. Static analysis is great if you can do it parallel when you are programming, because it instantly reminds you of mistakes and things that you forgot. When frequently reminded, one does gradually start remembering the better ways of doing. Unfortunately this type of interactive static analysis tools are rare. Most static analysis tools are such that you write the code first and only then submit the code to the analysis for example to the cloud. And then after the analysis is done you only get the results. At that point you might already have forgotten what you did.

At this point there was a comment from the audience that the human factor should be considered as well. Antti told that yes the developer training and building the security understanding is very important. And the threat modelling, when properly done is one very important way to create this understanding. Here came another comment that a tool is not really solving all, if any problems. Eventually “a fool with a tool is only a faster fool".

Static analysis tools generate typically a lot of false positives - so it conditions people to click issues to solved state, without deeper analysis of the incident. And this is very bad! Antti told an example of a large company that took a static analysis tool into use. After a while they realized that the amount of false positives was so high that they needed to hire a team to clarify those. So what was the value of this effort? Some languages have better static analysis tools than others. There are some languages that are so freely typed that creating a static analysis becomes very difficult. There are also static security analysis tools as well as tools that are actually only “glorified grep". So selecting a proper tool for the project might be a difficult and time consuming task.
There are also Web application firewalls that are ok if used to stop exploits when the fix to vulnerability is not yet published, Unfortunately some companies refuse to use resources to corrections as the firewall can be used to stop those exploits. There is also a heuristics tool called Emet to sniff exploits. But these tools are rather for operations than to coding phase.

Antti ended this lecture by telling about the next assignment that is writing an essay of analysing a security tool.

References:

[1] SW security lecture 3  
https://wiki.helsinki.fi/download/attachments/127960693/Session3.pdf?version=1&modificationDate=1395841684183&api=v2
[2] nSense  
http://www.nsense.fi/
[3] Microsoft Software Development Lifecycle:  
[4] Microsoft SDL process training:  
bsimm.com/online/
[6] OWASP Software Assurance Maturity Model (OpenSAMM)  
https://www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model
[8] Sovelluskehityksen tietoturvaohje, VAHTI 1/2013  
http://www.vm.fi/vm fi/04_julkaisut ja asiakirjat/01_julkaisut/05_valtionhallinnon_tietoturvallisuus/20130207Sovell/name.jsp
[9] Payment Application Data Security Standard  
https://www.pcisecuritystandards.org/security_standards/