Software Security

Group project:
application security verification
using OWASP ASVS
Brainstorm

What would you do if someone asked you to check if some application they use (and possibly bought) is secure?
Assurance

Big challenge:

how can we provide assurance that an application is secure?

NB: it is much easier to
demonstrate that an application is not secure
than it is to

guarantee that it is secure
Assurance

Why is the question below silly?

how can we provide assurance that an application is secure?

We only want to know if the level of security is acceptable given the associated risks
Group Project

- We do a security code review of a web-application
- following the OWASP standard
- trying out some source code analysis tools

Time for this +/- 50 hours, ie ≥ afternoon/morning per week for next months
Goals

- experiencing a software security review process
- how useful are existing standards and approaches?
  - esp OWASP Application Security Verification Standard
- how good are tools we can use?
  - eg Fortify, RATS, CheckMarx
- how *should* security design and implementation decisions wrt security have been made and documented?
Non-goals

• For some, this is throwing you in at the deep end. I realise your experience varies a lot!

• Don’t be tempted in copying results from other groups
  – whether or not find any security problems is not important, it’s about forming an well-argued opinion about code reviews, the ASVS as guide for this, static analysis tools, etc.

• My hidden agenda: getting some empirical data on doing code reviews

aims to normalise the range in coverage & level of rigour in performing web application security verification

NB not “verification” in the mathematical or even testing sense
OWASP ASVS Levels

ASVS defines detailed verification requirements for Levels 1 and above; whereas Level 0 is meant to be flexible and is customized by each organization.
Categories of Verification Requirements

V3. Session Management    V11. HTTP Security
V6. Output Encoding/Escaping
V8. Error Handling & Logging
V1 removed in 2014 edition
V13 & 17 out of scope
ASVS Security Requirements

ASVS provides checklists of security requirements to check

- clustered in categories

where security requirements are stated in a “positive” way, eg

- negative
  - there are no XSS attacks

- positive
  - all HTML output that includes user-supplied input is properly escaped

Note: checking positive rather than negative statements is very different
Verification Techniques

- **Dynamic**
  - using running application
  - aka (penetration) testing

This can be

- manual application penetration testing
- automated application penetration testing

- **Static**
  - using source code
  - aka code review

This can be

- manual
- automated using code analysis tools

Focus of the group project
Caveat: tools cover at best 45%? [study by MITRE]

• All application security tool vendors’ claims put together cover only 45% of the 695 known vulnerability types

• Very little overlap between tools, so to get 45% you need them all (assuming their claims are true)
static source code analyzers

- **RATS**
  basic old tool – glorified GREP/CTRL-F

- **Fortify SCA & CheckMarx**
  modern state-of-the-art commercial tools

- **RIPS**
  open source tool, but RIP?

Any other tools you know or have experience with?

Yasca, PHPLint, FindSecurityBugs ....
To start

- Fix a weekly morning/afternoon/night to work on this?
- Keep a log what you are doing, and who does what
- Read the AVSV
- Look at the code

- Panic?
- Install the code to get a feel for functionality?
- Try some of the code analysis tools?

- Think about you divide the work & organise this
- April 24 : progress discussion in class with SIG
NB We skip the most important steps

NB by jumping straight to look at the code using the ASVS we skip the most important first steps of any security analysis:

1. identifying security requirements and their importance
   - ie. threats and impacts, for a good risk assessment

2. defining attacker model
   - eg `standard' online attacker, insiders, vandals, hacktivists, mafia, NSA, …
   - should also consider capabilities & motivation
     even if your site has little value, an attacker might still be interested:
     to break into the machine hosting it, to use it as an attack vector to attack your visitors, to steal re-used passwords, …