Rob Wagner, Ryne Bell, Alex Petitjean

Cyber Security

Presentation Outline

- What does software quality mean in the domain track?

    Software quality is essential in cyber security. In order to assure that you keep your users' and any other sensitive information secure, the software must be able to prevent unauthorized access, recover from a system failure, and prevent corruption of data; whilst still granting access to authorized users. The software must be designed with these features in mind, and because data sensitivity is an issue, developers need to pay extra attention to any security flaws there system could have, and rectify them immediately.

- What are the important quality attributes?

    Thorough testing strategies / bug removal:

    It is important when designing with cyber security in mind that all bugs concerning data security must be removed from the system before release, and the system must remain up to date and tested so that nothing can threaten users' private/secure data.

    Monitoring system capabilities:

    The data must be secure, but it also needs to be accessible and productive to valid users. For that reason, developers need to make sure their system can handle the stress it will be put under at any time. The system must be able to safely recover from a crash and not lose any data as well.

    Secure coding:

    Developers also need to be aware of vulnerabilities in their software. Some tricky problems to deal with are buffer overflows, integer overflow, and code injection. Developers need to take whatever extra precautions (such as sanitizing input) so that data is not compromised.

- What processes/mechanisms are used to guarantee quality?

    Breaking the system up into smaller components

    - complexity of individual components is reduced
    - techniques such as automated theorem can be used to prove the correctness of crucial software subsystems

    "Defense in depth" - multiple subsystems must be violated to compromise the entire system

When formal proofs are not possible, rigorous code review and unit testing are used

- What metrics if any are used?

  Core vs. ecosystem

  - Standard metrics can be used to ensure the core works properly, but tests for the ecosystem are a little more specialized
  - Core functionality is tested much the same as any other code, but often with more emphasis on code review and other methods to ensure the program not only works properly, but is written in a security conscious manner
  - Ecosystem tests determine what the software does on a compromised system, or when a user of the software has malicious intent

  Current ratio of detections to actual infections as compared to the baseline figure

  - Or blockings of something malicious (depending on exactly what the software is)

  "Fuzzing" - sending various types of pseudorandom data to available interfaces to discover unknown flaws present in the software

  Error or vulnerability reporting in final product reported by users

- What processes/mechanisms do you suggest?

  Ratio of time spent reviewing old code to developing new

  - This should increase over time as the codebase becomes more complete

  Number of comment lines per method

  - This should be higher for security sensitive sections which are reviewed often

  Average Method Size

  - This should be relatively low to make understanding each method very easy
  - Makes ensuring correctness easier

  Penetration Testing

  - People who attempt to purposefully break the system in any way
  - These attempts should be converted into test cases if possible to keep track of the progress on preventing them, and ensuring the continue to fall to break the system

- What metrics do you suggest?

  Process metrics

  - Use of secure coding standards
  - Time to correct vulnerabilities
  - Trustworthiness of development team

  Vulnerability metrics

  - Vulnerability types
  - % of code re-use from other projects or products
  - % of code that is third party

- Examples of software development successes and failures

  Windows NTLMv1

  - Credentials compatible with many different applications
  - These credentials could be intercepted and used to access other applications

- Are there any standards/regulations that are in use?

  - Carnegie Mellon's Computer Emergency Response Team (CERT) have published secure coding standards for several languages. The standards were developed by members of their team and a community of software developers.