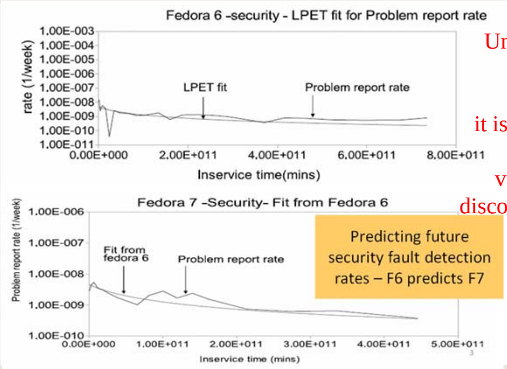
An Investigation of Scientific Principles Involved in Software Security Engineering

Mladen A. Vouk and Laurie Williams

Reliable engineering of secure software is still in many ways an art. System development and maintenance teams often do not have enough information about (or understanding of): a) Software operational profiles when software is under attack – malicious testing changes profile; b) Resultant direct and indirect interactions among system components; and c) The impact of human knowledge and skills (or lack of the same), development design choices, processes, practices and testing on the security of software-intensive systems. High-assurance systems should be reliable, safe AND secure. Software Security Engineering (SSE) is an applied science of predicting, measuring, and managing the security of software-based systems to maximize customer satisfaction. This project is investigating scientific principles behind SSE by extending science behind software reliability engineering (SRE).

Attacks change operational profile of a software product and this has to be taken into account when engineering secure software by engeenering for both aleatoric discovery of vulnerabilities, and for epistemic attacks.

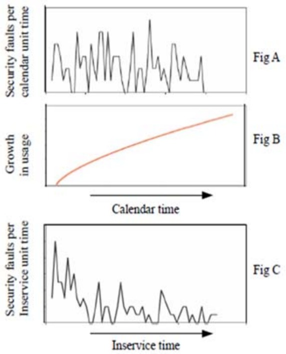


Predicting future security fault detection rates – F6 predicts F7

Under normal operating conditions it is possible to predict vulnerability discovery rate in software.

Software Security Problems are Rare Events*

- FAULTS: Security problems are only 0.05% to 5% of the total number of problems
- Discovery rate is relatively constant
- FAILURES: About 0.05% to 2% of the total number of s security problems in software result in field failures (exploits)
  - **Voluntary** exploits are 30% and 80% of the problems. (“Wash your hands!”)
- DISCLOSURES: Number of security fault disclosures per inservice week (number of systems time in service) over the life-time of a software product release can be as low as 5 e-09
- POISSON: In open source products disclosure of security faults appears to follow Poisson process.



(*) Anbalagan 2011

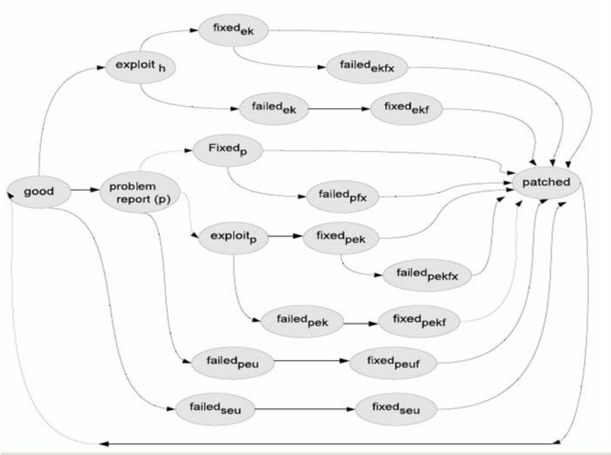
Open Source Software

Note: We are currently confirming above findings on more recent data.

Approach

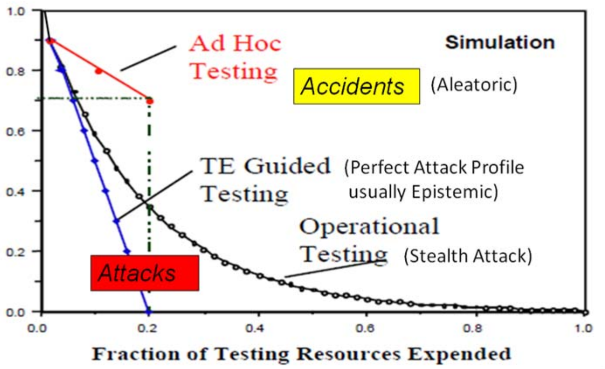
We are exploring: i) Principles behind reliable “attack” profiles; ii) Principles behind two possibly very different categories of security related errors humans make – epistemic (knowledge, cognitions, intent, bias) and aleatoric (stochastic); and iii) Principles behind tools, methods and approaches that successfully target security vulnerabilities. In theory, epistemic faults might be preventable using prior knowledge about developers and attacker categories. Aleatoric failures may not be preventable, but can reduced and handled using “classical” SRE methods.

We propose to develop a corollary to the operational profile, which we call the *attack profile* by a) examining which parts of the code and data are most attractive to hackers, and b) shadowing and “recording” security designers and testers to see how they approach their tasks. We will a) examine already available open source repositories of vulnerability and security failure data, and b) partner with “white hat” testers in industry Analyses will deliver patterns, workflows, effectiveness, etc. This will enable us to design robust SSE practices.



Vulnerability Response Model

Efficiently Guiding Attacks and Security Testing



Hypergeometric Attack Profile Model

2012 Science of Security Community Meeting
Nov. 29-30, 2012
National Harbor, MD
http://cps-vo.org/group/sosmtg

Vote Here

NC STATE UNIVERSITY
Department of Computer Science