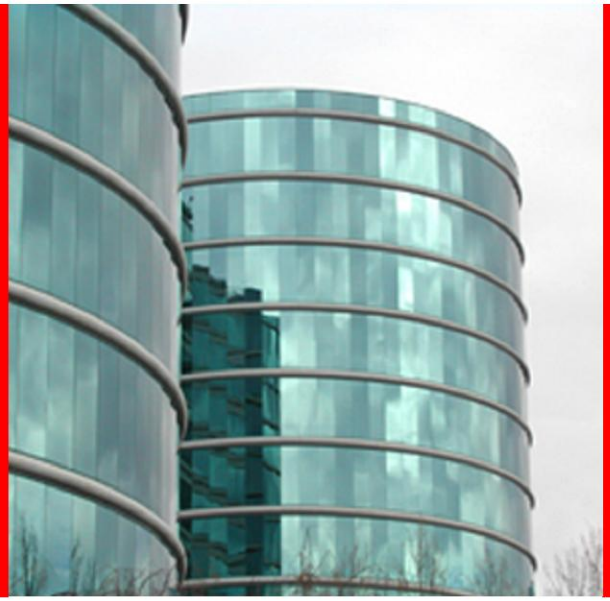

ORACLE®



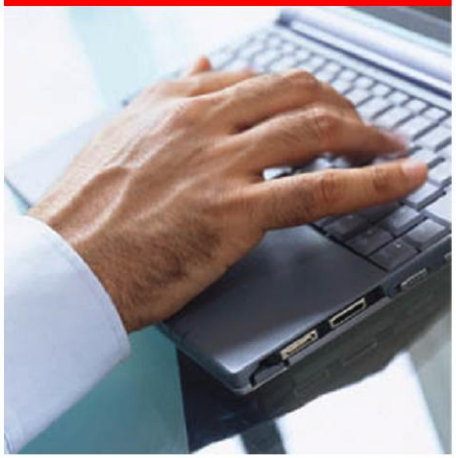
ORACLE

Common Criteria and Source Code Analysis Tools: Competitors or Complements

Adam O'Brien
Security Analyst

Contents

- The vulnerability landscape
- Secure code analysis tools
- Incorporating the tools in CC
- Conclusions

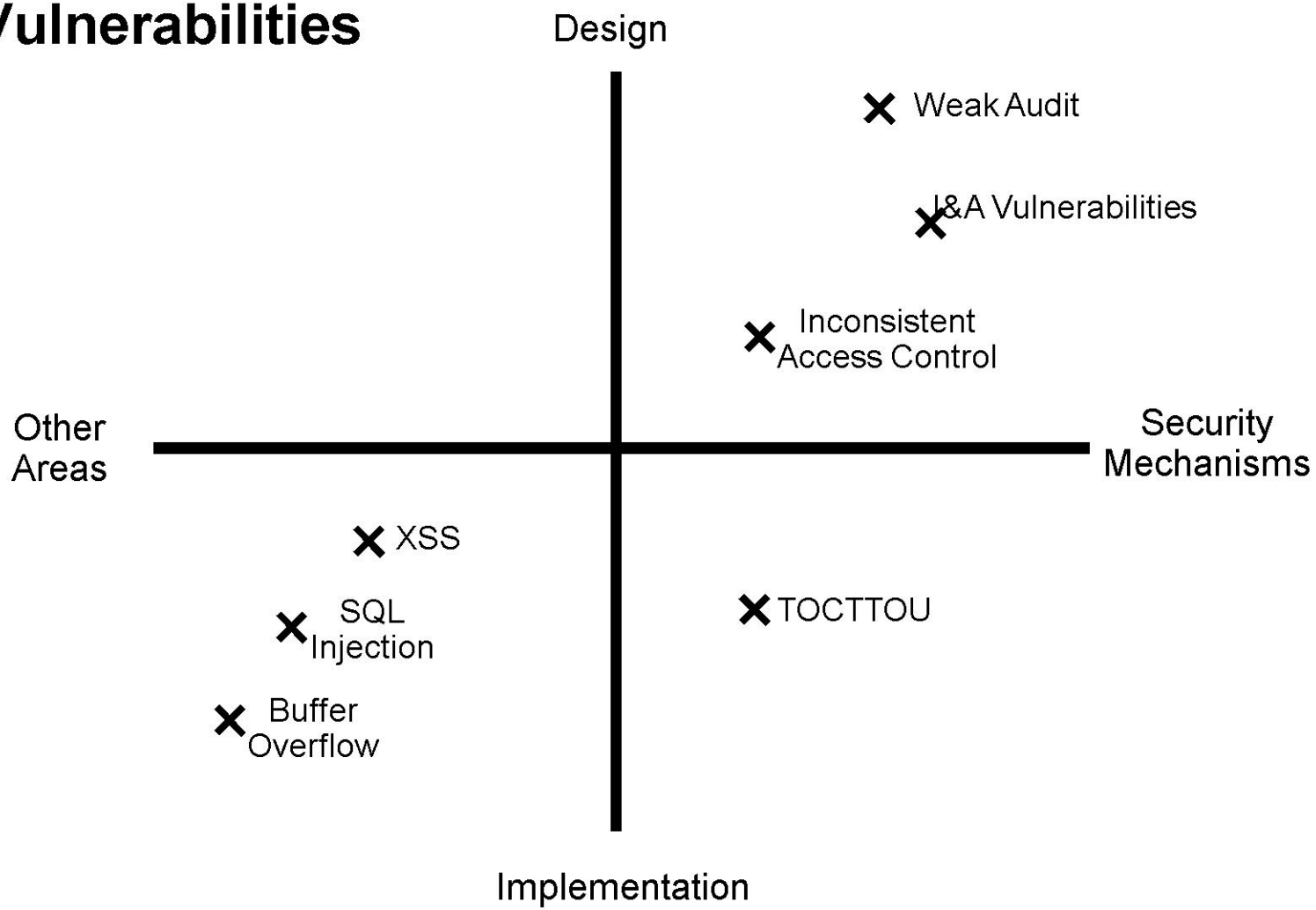


A Taxonomy of Vulnerabilities

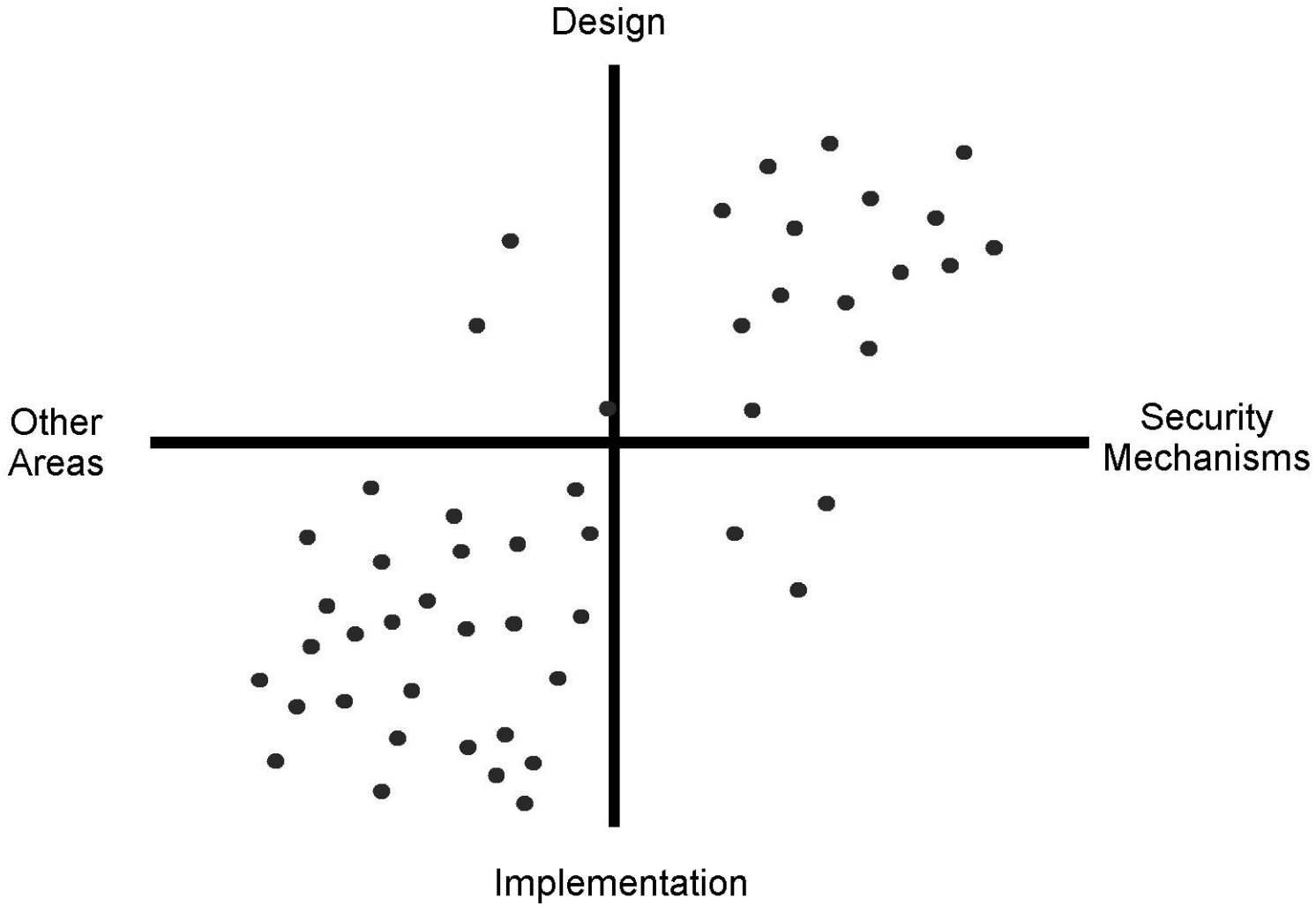
- Two important dimensions: when and where
  - When
    - Design
    - Implementation
    - (Operation)
  - Where
    - Security Mechanisms
    - Other parts of the code base

    A secure system must allow users to do everything they are allowed to do and nothing they are not allowed.

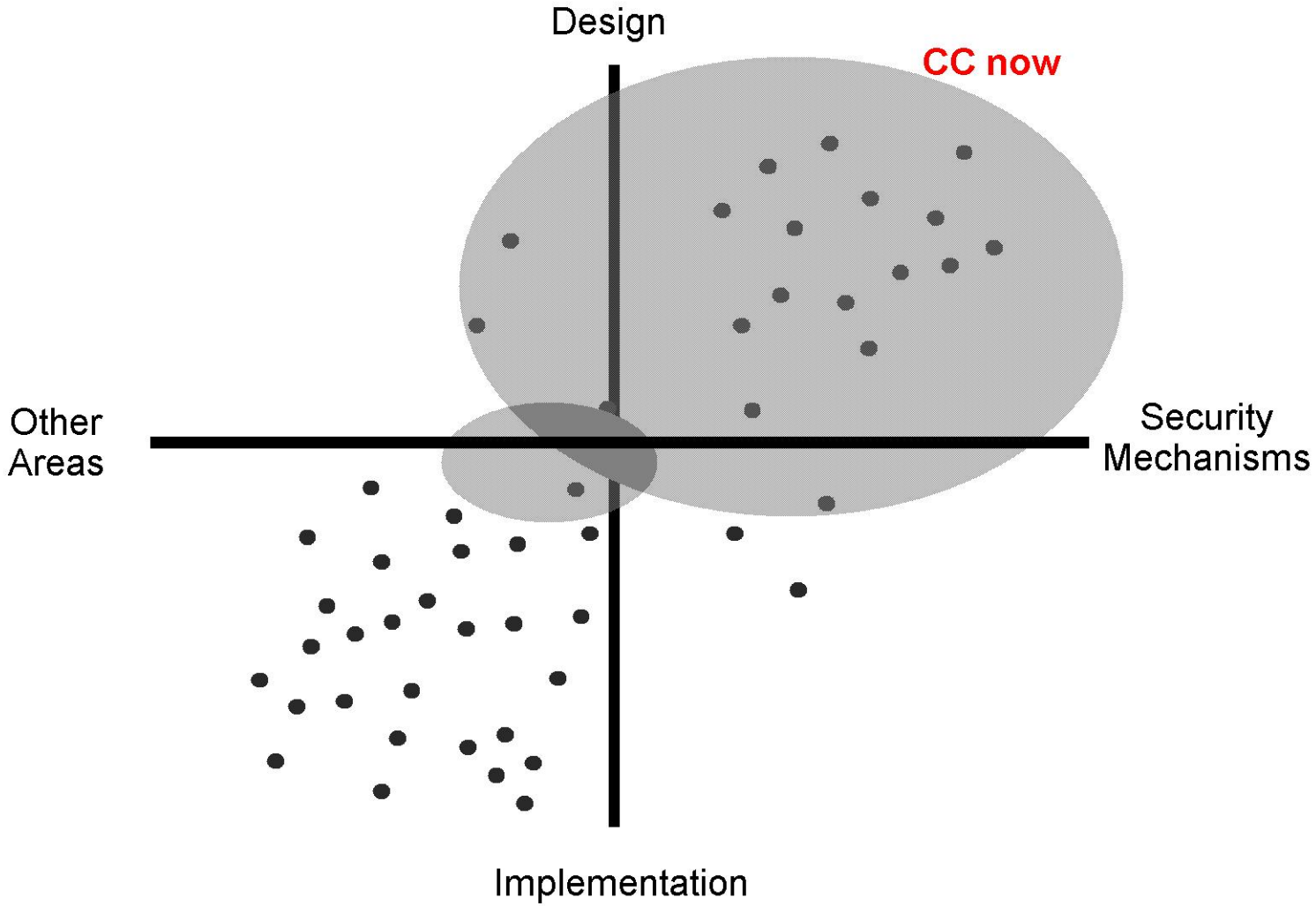
The Vulnerability Landscape: Common Vulnerabilities



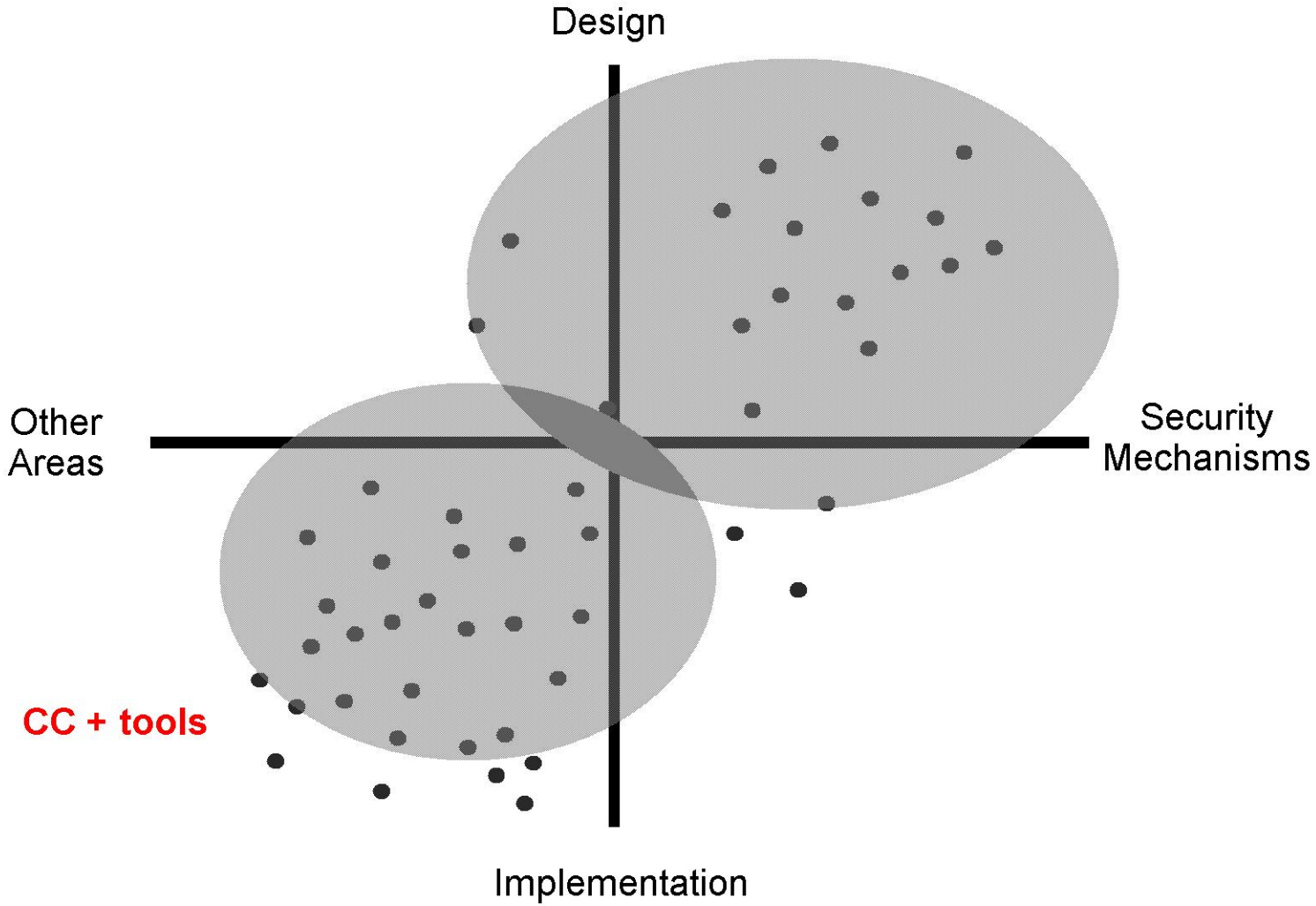
The Vulnerability Landscape: The Pattern



The Vulnerability Landscape: Current Coverage



The Vulnerability Landscape: Coverage with Tools



Secure Code Analysis Tools

Type of Secure Code Analysis Tools

- Purely static

  This can be as simple as grepping for insecure functions. More sophisticated analysis can find a range of secure coding standards violations. E.g. RATS

- Flow Analysis

  These tools compile the source code and examine information flows. This allows detection of memory vulnerabilities, like buffer overflows. E.g. Fortify

- Dynamic

  These tools examine the product at run-time. Specific test cases must be produced.

Using the Tools

- Set-up
  - Static: a few days
  - Flow Analysis: a few weeks
  - Dynamic: a few months
- Interpretation of the results
  - The tools will find hundreds or thousands of vulnerabilities.
  - It is hard to classify these as:
    - False Positives
    - Useful warnings
    - Significant Vulnerabilities

Using CC to Make Sense of the Tools' Results

A CC evaluation postulates:

- A threat model: assets, threats, assumptions
- A deployment scenario: external countermeasures, security policies

This information can be used to make sense of the results. Many tools can formally make use of such information.

Incorporation into CC

How to Introduce Secure Code Analysis Tools into CC

CC should examine process not content.

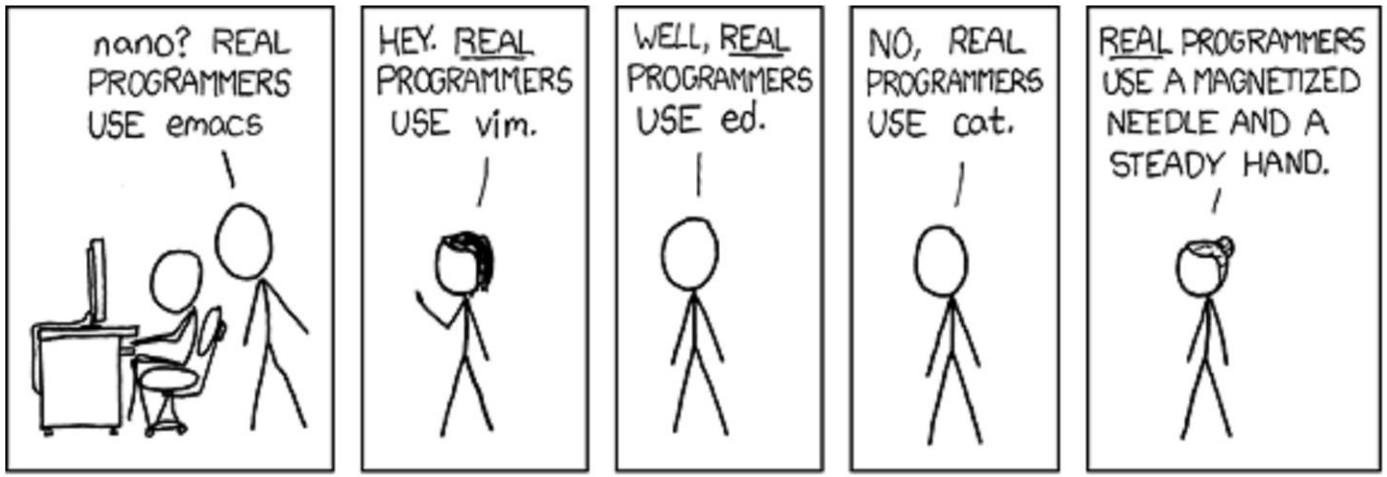
It should ensure:

- That the tools are used as part of a coherent plan
- That appropriate tools are used
- That the results are acted upon:
  - Systematically
  - In a robust manner
  - In a timely fashion

The assurance requirements suggested by Miguel Bañon last year seem fine.

Where to Introduce Secure Code Analysis Tools into CC

CC shouldn't tell developers what tools to use…



© xkcd.com

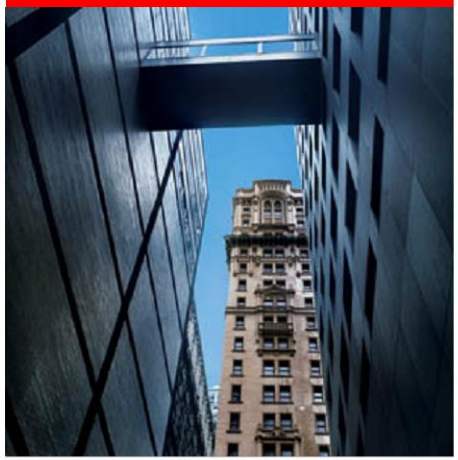
…but customers should be able to.

Tools should be added as an augmentation rather than incorporated into the EAL structure.

When to Introduce Secure Code Analysis Tools into CC

- Soon
  - Source code tools are cheaper?
  - Faster?
  - Find more vulnerabilities?
- They are a real threat to CC: *We must assimilate*

Conclusions



Conclusions

- Source code analysis tools are maturing fast
- They may be more cost-effective than CC
- CC and source code analysis tools are not rivals:
  - They find different types of vulnerabilities
- Indeed they are complements:
  - Together they discover many common vulnerability types
  - CC threat modeling gives a context to the tools results
- The use of these tools should be incorporated into CC:
  - Quickly
  - Outside the EAL structure

Questions

adam.obrien@oracle.com

ORACLE IS THE INFORMATION COMPANY