GOTO Chicago 2017

How I learned to quit piling up bugs and

fix the damned software

John Steven

 @m1splacedsoul


ONE DOES NOT SIMPLY
TEST THEIR APPLICATION SECURE

We Can‘t Test Applications Secure

Vulnerability Assessment & Penetration Testing?

67% Discovery on re-test

98% Re-exploit rate

We Can't Band-Aid® Apps Either

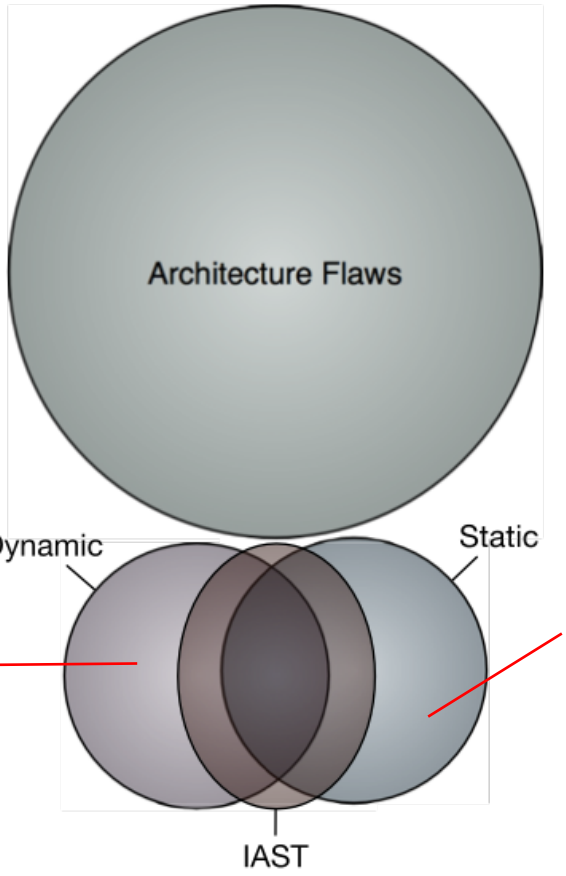
100% Re-exploit rate for underlying app

77% of rules to remediate 1st test evaded

(When RASP deployed to protect app)

Better Mousetraps Don't Change the Game

Coverage – Vulnerability Space



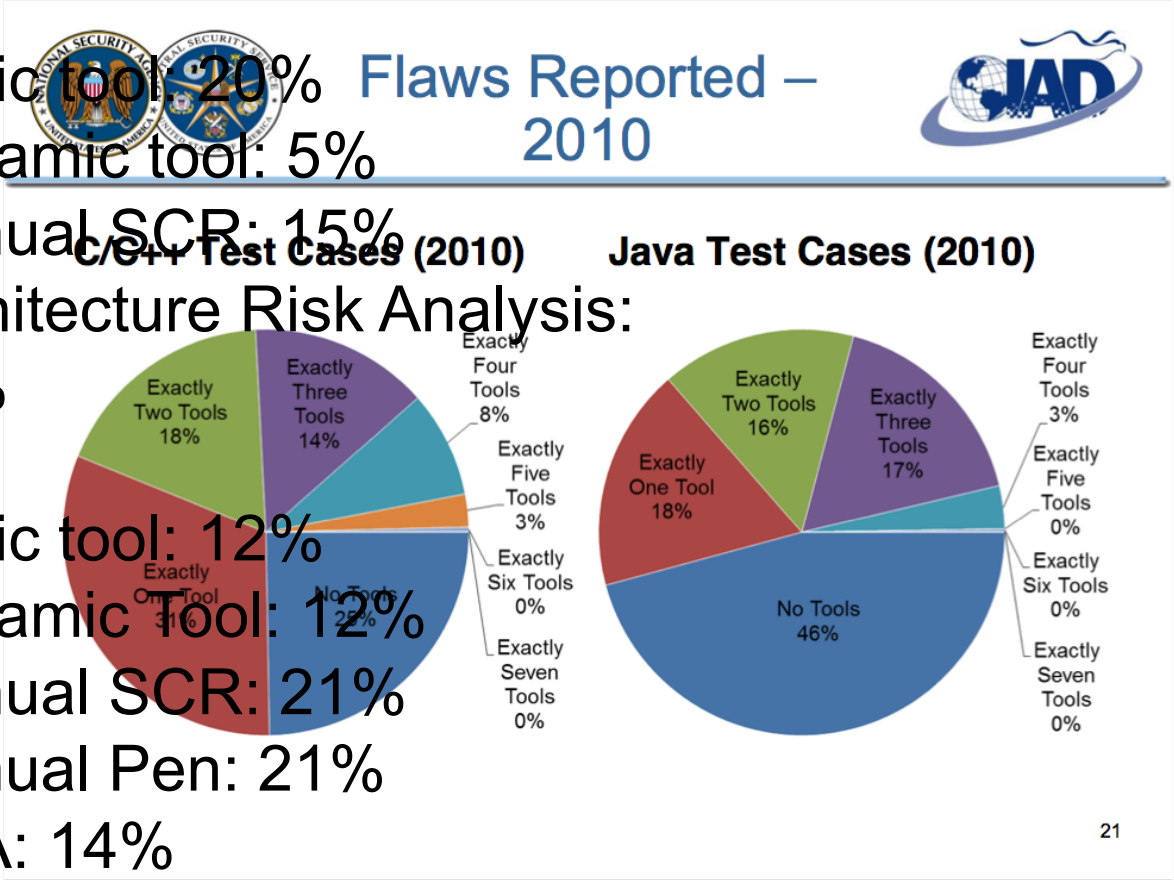
Business logic
State transition

Full program
analysis & SAT

Data Born Out in Cigital's Practice

- Static tool: 20%
- Dynamic tool: 5%
- Manual SCR: 15%
- Architecture Risk Analysis: 60%

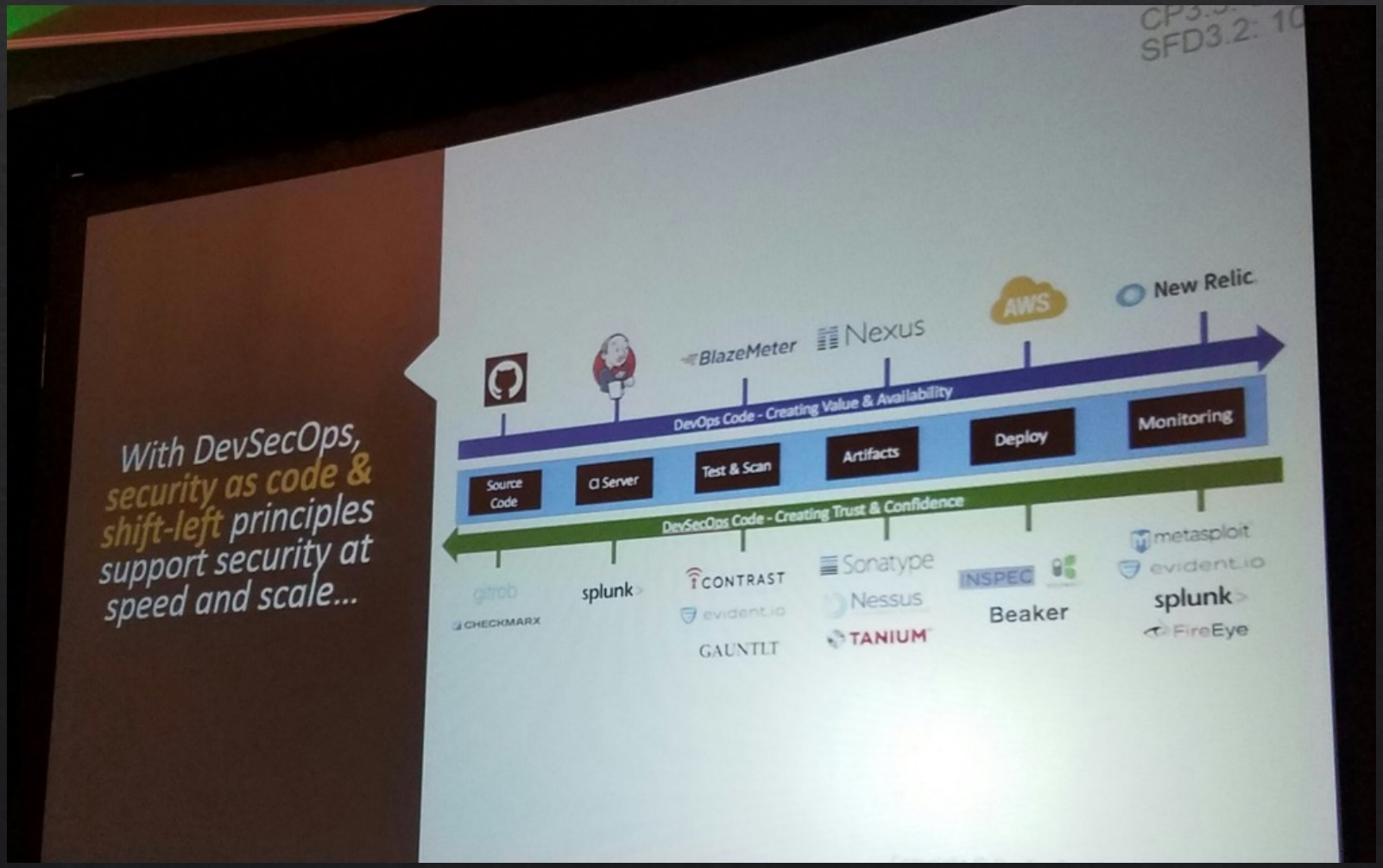
- Static tool: 12%
- Dynamic Tool: 12%
- Manual SCR: 21%
- Manual Pen: 21%
- ARA: 14%
- Sec Testing: 20%



* "Flaws Reported" - http://hcss-cps.org/events/willis.pdf

‘State of the art’ DevSecOps

“…Code First Ask Questions Later"



So, what am I sellin'?

Some Things Orgs Find Useful

Three Capabilities that Might Help You

1. Production Gate (aka “the big red button”)
2. Secure guidance, code, and design
3. Make sure you’re automating (scaling)
   - Facilities to deploy secure code faster and
   - NOT the sources of pain

Keynotes don't change your org's culture

Deeply Cultural

1. Get a big red button
2. Secure guidance, code, and design
3. Make sure you're automating (scaling) [the correct thing]

Plenty of Resources: Find a Culture Match

- Etsy –
  - Effective Approaches to Web Application Security – zL
  - Data-driven Security - nG
- F Secure – Topconf Tallinn '12 – aV
- Twitter - Put Your Robots to Work – aS
- Living Social - AppSec Ritalin, and Failing Fast – kJ
- Riot – Leveling Up Your AppSec Program - dR

Hiring Savvy Devs Proves Successful but Hard

Hands-on
Artistic
Hard to scale

1. Get a big red button
2. Secure guidance, code, and design
3. Make sure you're automating (scaling) [the correct thing]

A Challenge, and a Means to Meet it

1. Get a big red button
2. secure guidance, code, and design
3. Make sure you're automating (scaling) [the correct thing]

Why Others Have Failed

Piling up bugs is hopeless, under any name and by any means

Taking the “reducing friction” stance is DoA

And…

Ineffective enforcement (2%)

A Note about Brakeman

- …and Dr. Justin Collins

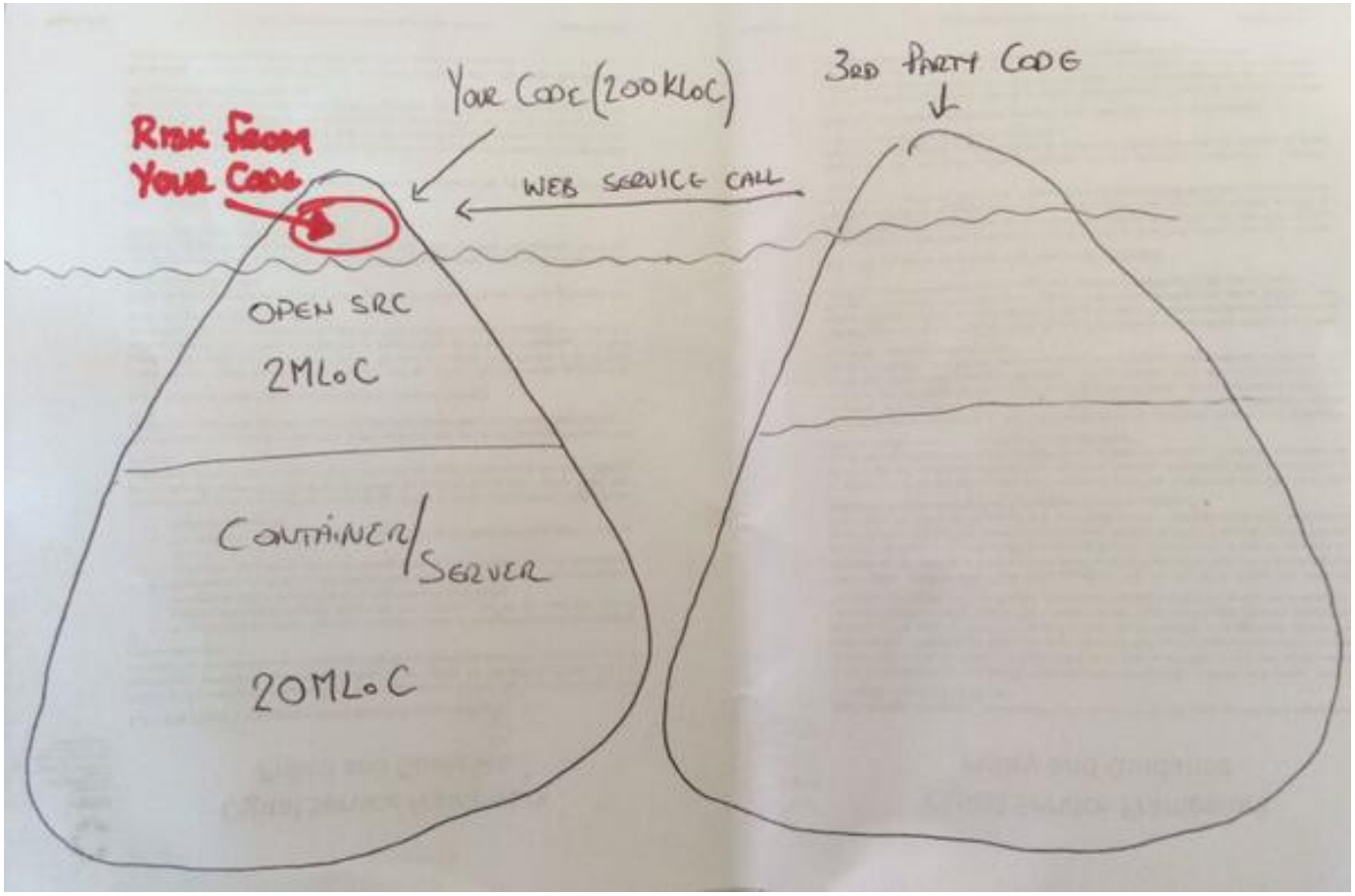
Great SAST

Actively facilitates deploying secure code faster

And

*Provides devs **visibility** towards understanding*

“Your Application” is …



How Tools Map to Your Tasks

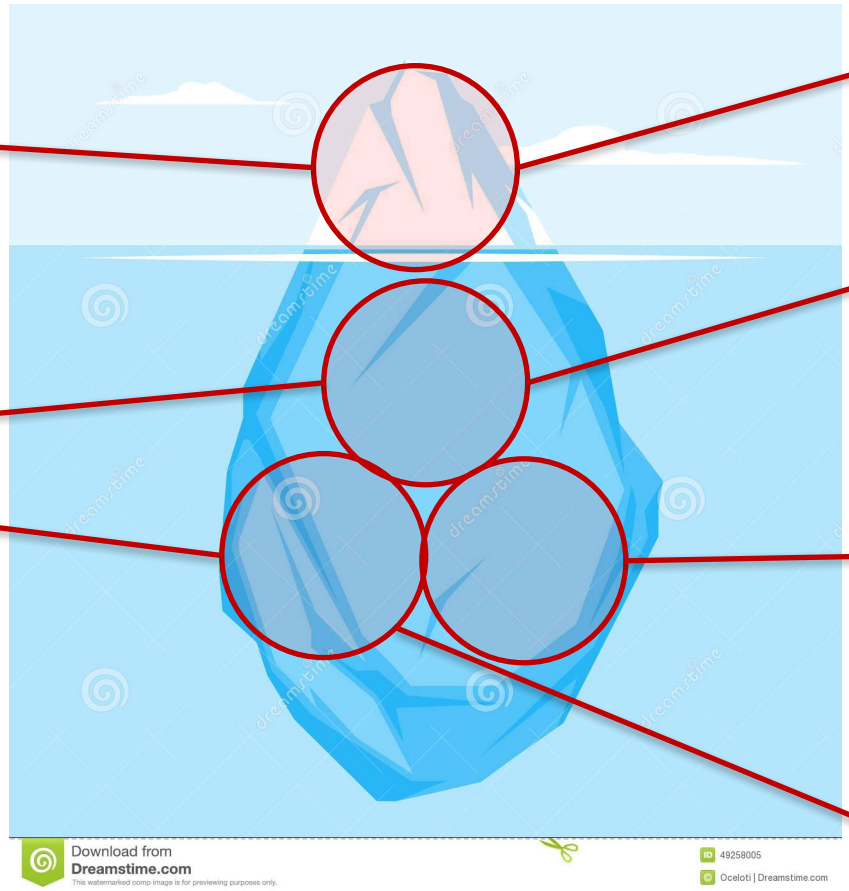
OLD-SCHOOL TOOLS

Identify vulnerabilities

SOME NEW-SCHOOL TOOLS

Patch Open Source

Sometimes: Avoid insecure functionality



Secure Application code

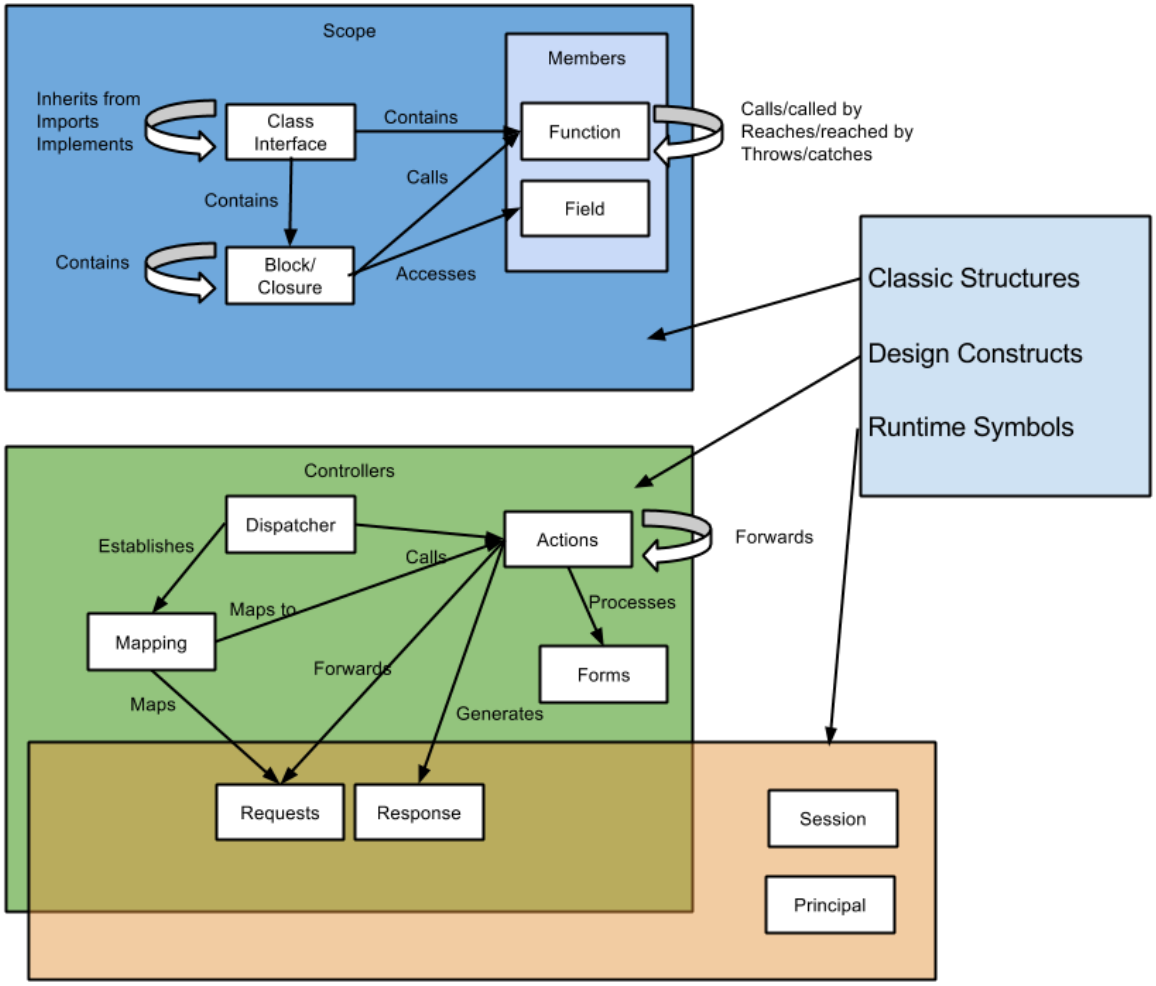
Patch Open Source

Use security controls

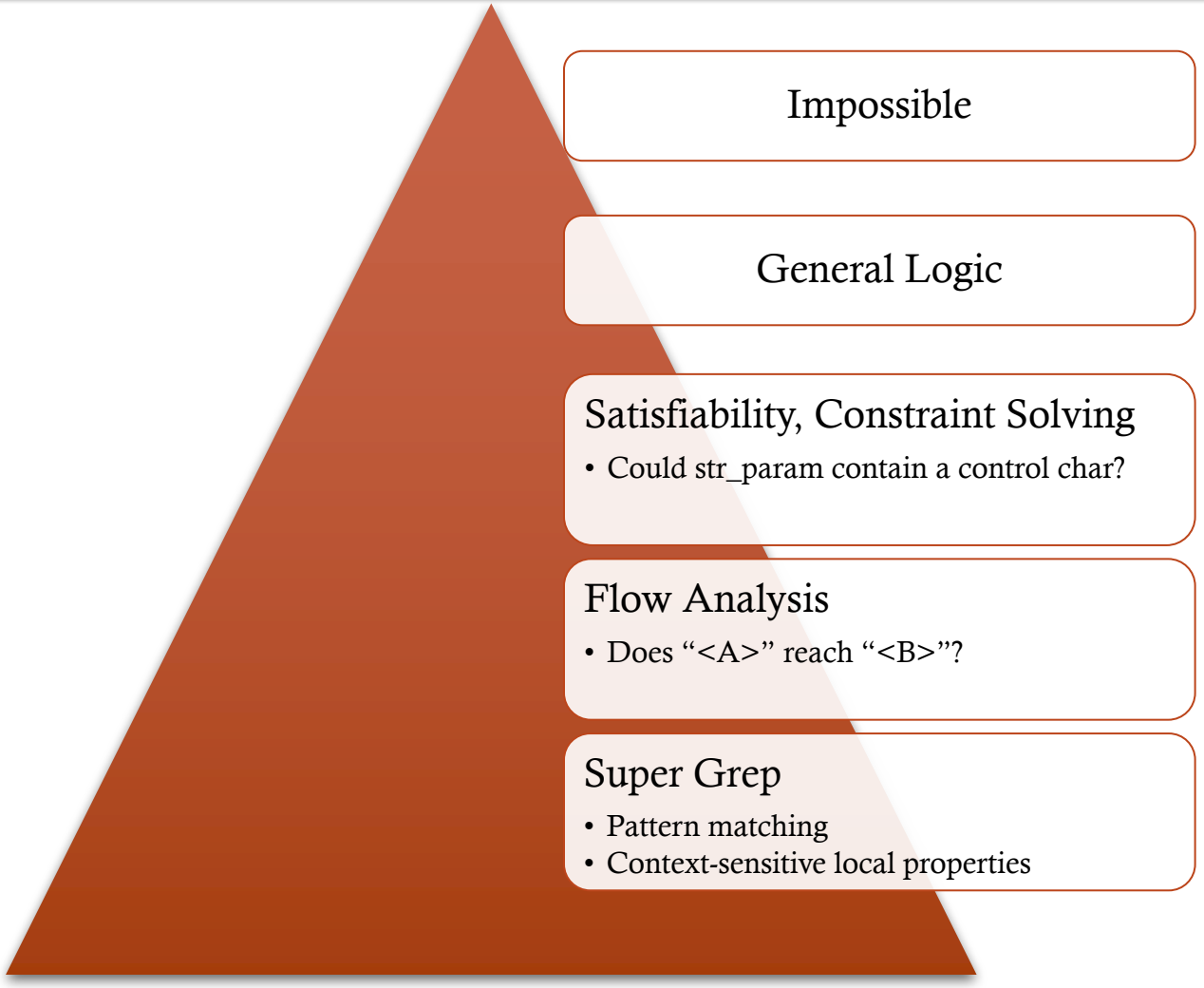
- Configured Correctly
- Everywhere necessary
- For the right reasons

Avoid insecure functionality

What static tools 'see'



What Tools Can Do…



Summarizing

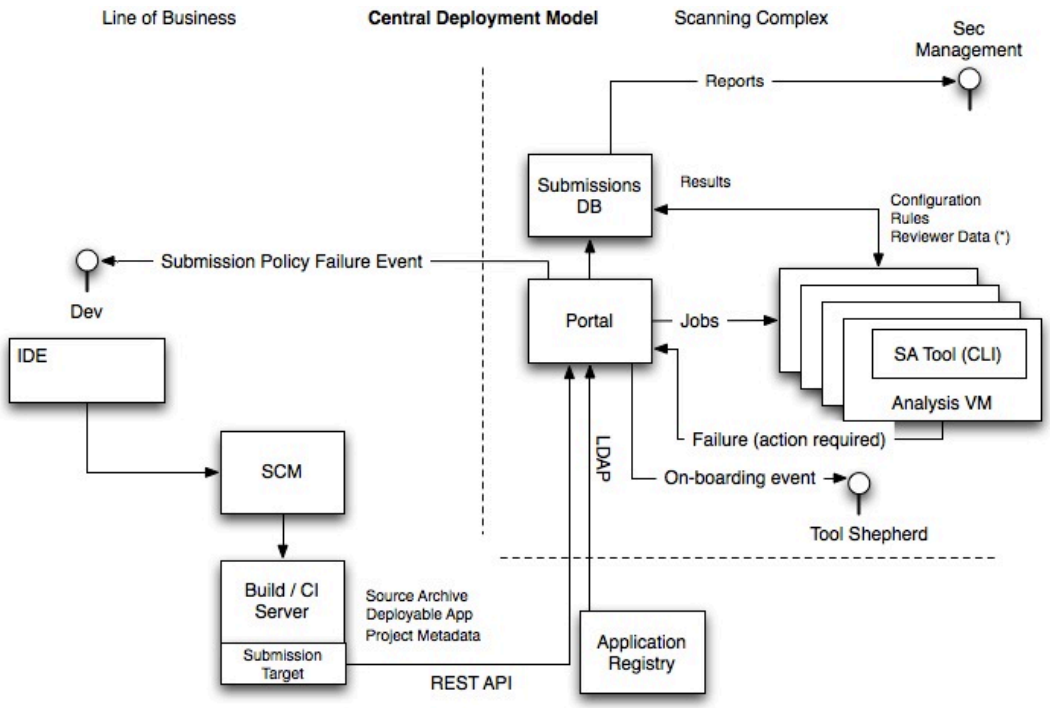
- Does OSS need upgraded or patched?
- Is OSS secure-by-default or in need of configuration
- Are dangerous functions being avoided?
- Are provided controls used …
  - Everywhere necessary?
  - For the correct purpose?
- Is the code written vulnerable to attack?

Solution Topology

What organizations are addressing

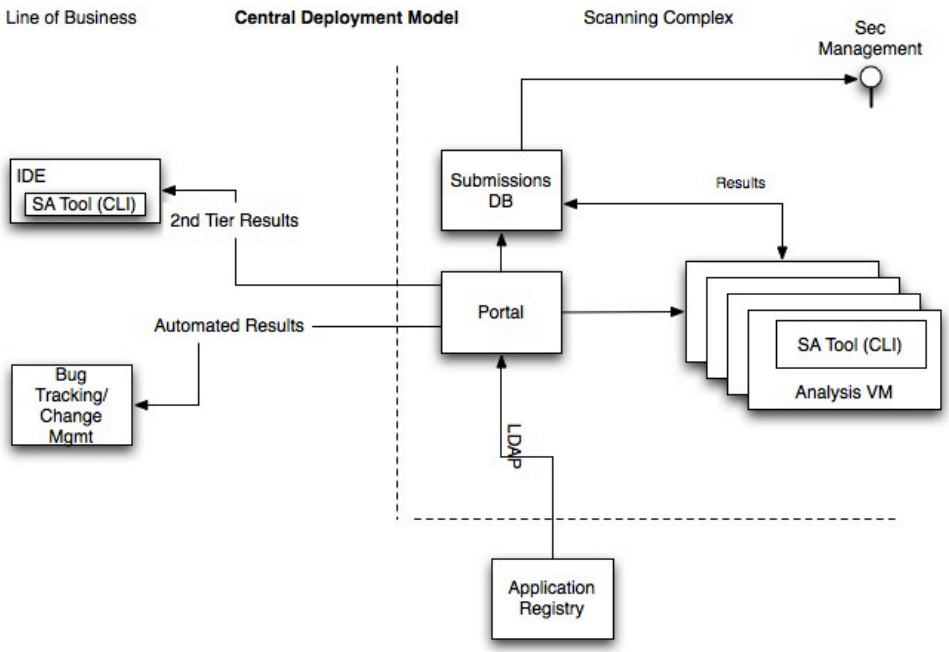
Submission Architecture

- Orchestration points
  - Build/artifact mgmt.,
  - SCM, &
  - IDE
- Responsibilities
  - Goodness (interactive, on-boarding)
  - Formatting, organization, & dependencies
  - Meta-data
    - Blame
    - Change
    - SDL state



Results Architecture

- [Reviewer]
  - Conducts Source Code Review
- Developer
  - Receives automated results from bug tracking
  - Receives 2nd tier of results in plug-in
- QA
  - What is their role?



Rules Management

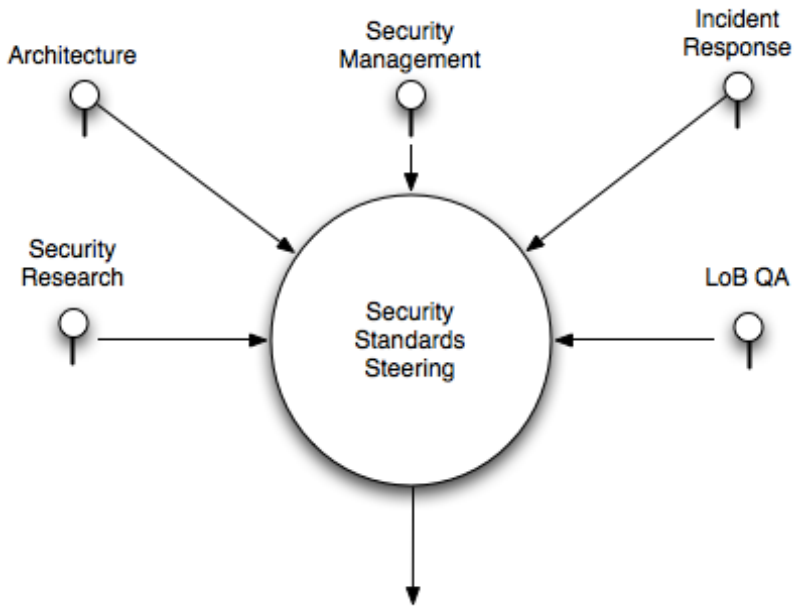
What organizations are addressing

What form should Output Take?

- Findings…
- CBT – Vignettes (6 min)
- Video?
- Discussion Board?
- Guidance
- Contextually-aware Guidance
- JIRA Ticket
- Auto-correct
- Pull-Request

An Opportunity to Provide Guidance

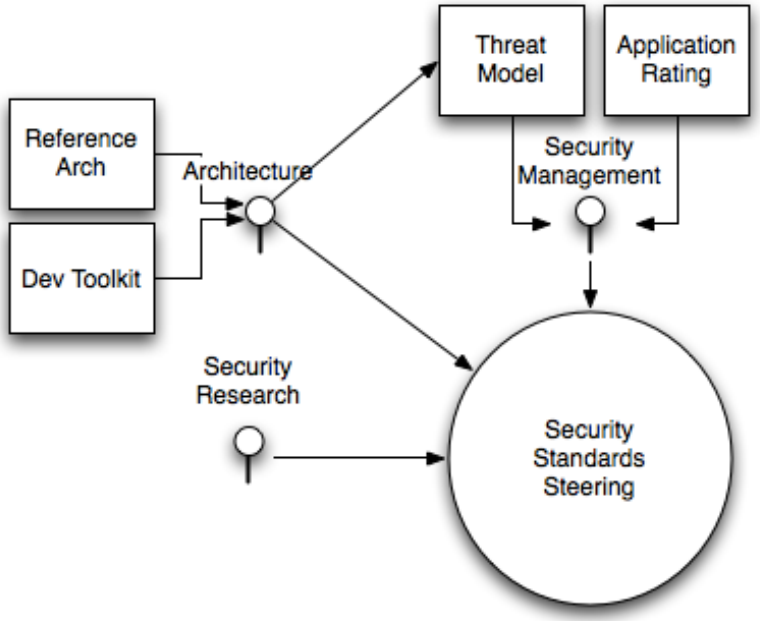
Source of Rules



- Manage rules conceptually
  - Treat rules, tool config. as software release (testing, versioning)
  - Select optimal assurance tool for rule
  - Combine proactive & reactive rule sources
  - Acknowledge multiple stakeholders
- Deploy rules automatically

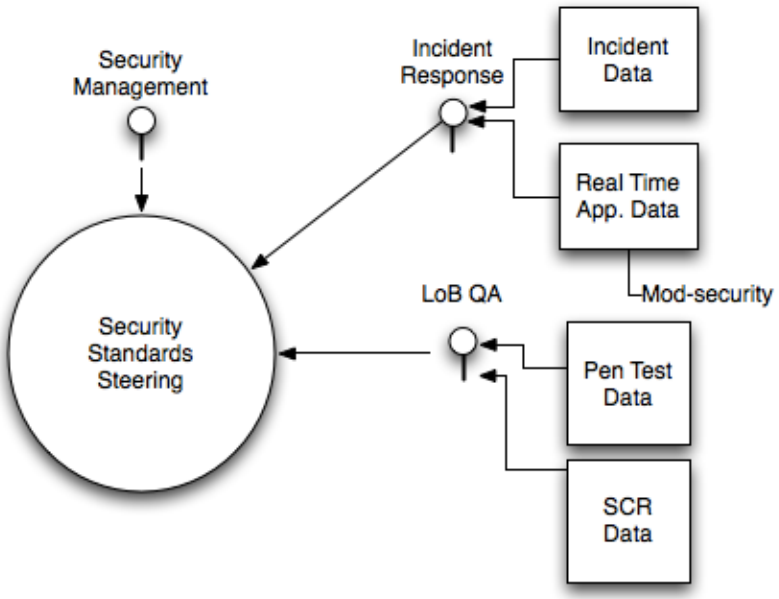
Proactive Stakeholders

- Threat Model/App Rating
  - Drive assessment type, frequency
  - Generate configuration
  - Drive # of rules
  - Drive rules for attack surface
- Maturity of app possible



Reactive Stakeholders

- Actual Incidents
  - Drive high priority
  - Generate new rules
- Assessment Data
  - Drives rules priorities
  - Drives reduction of false positives
  - Creates application-specific rules
  - Creates framework-specific rules



Thank you for your time

-jOHN

State of Demand: SCR Volume

- Central
  - 13.5 MLoC
  - 200 Apps / yr.
  - 50 MLoC
  - 100 MLoC
- Self Service (per year)
  - 550 Apps (23MLoC)
  - 300 Apps (35 MLoC)
  - 350 Apps (14 MLoC)
- Aspirations
  - 100+ MLoC / day
  - 1000s Apps / yr

Selecting Applications

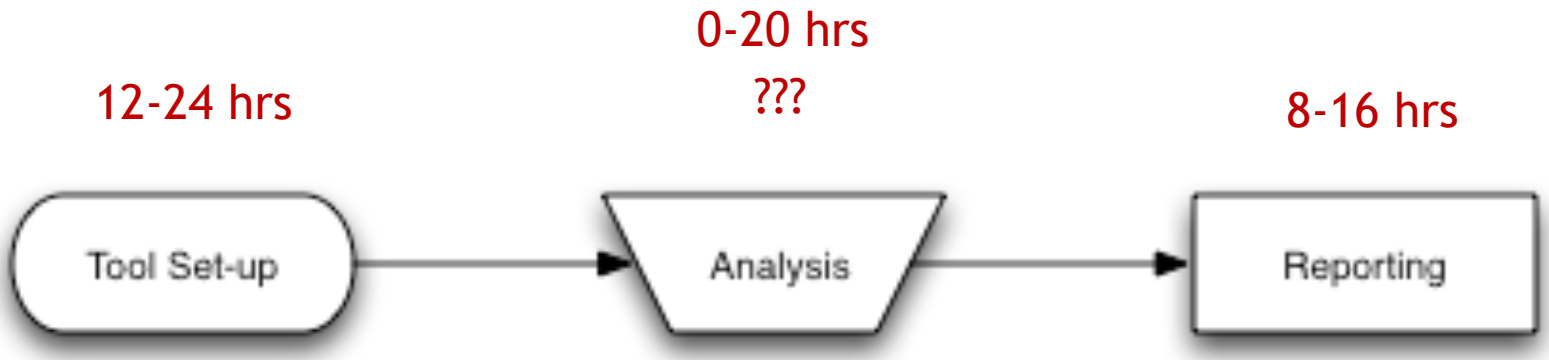
- Risk models in place pick:
  - Automated vs. Manual approach
  - Tools (SaaS, Commercial Big-box, OSS)
  - LoE for manual efforts, results triage

- Orgs picking from *internal* + external apps

Outstanding Issues

- Several arguments persist:
  - Where do SCR tools fit?
  - Who pays for this? (Audit, Security, Business)?
  - Can SCR be combined with other assurance methods?
  - Where can this work be done?
  - What skill-set is necessary to complete this work?

State of the Practice – Code Assessments



- It takes a day and a half to get results
- It takes a day or two to report
- That leaves very little time for thinking

Staffing Trends

- Triage
  - 2-5 persons
  - Tool vendor management
- Review
  - 0-24 reviewers
  - Some organizations remain entirely domestic
- Use vendors
  - Spike management
  - On-boarding

Emerging Roles

- On-boarding specialist
  - *Highly technical & experienced*
  - Writes custom rules for org. in self-service
- Security Researcher
  - Interfaces with tool vendor
  - Extends scanning capabilities
- QA
  - Conducts results triage

Costs

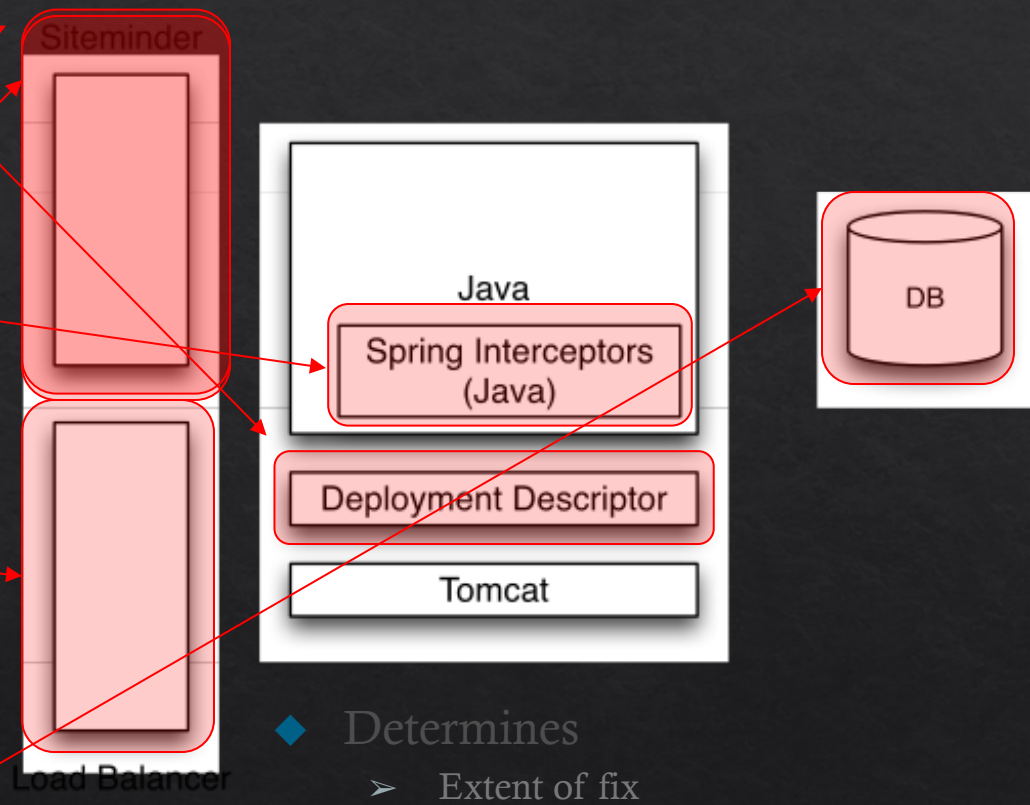
- Licensing
- Staff
  - $820K
  - $5MM
- Total Cost: (Licensing, Staff, Services)
  - $4.8MM
  - $9.2MM

Doing the work

- Perform scan & generate a results file
  - 2 calendar days, 16 mhrs
  - 7 calendar days, 24-32 mhrs
  - 14 calendar days, 40 mhrs
- Conduct Review:
  - 0 mhrs
  - 1-2 calendar weeks, 20-50 mhrs
  - 2-4 calendar weeks, 80-160 mhrs

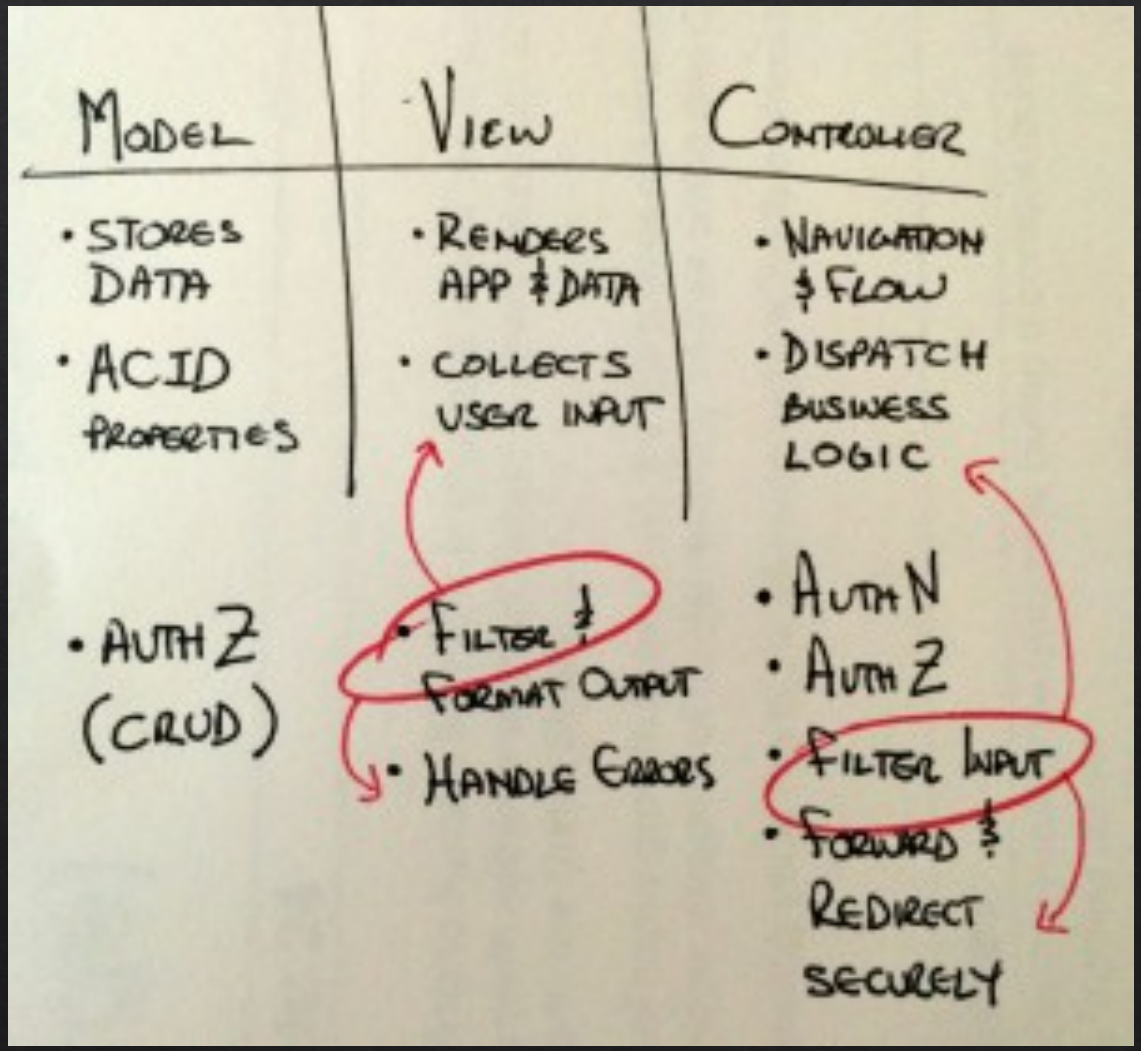
Deciding *Where* to Remediate is Important

- Do not restrict:
  - Max password length
- Do not restrict sets of characters
- Lock account after n attempts (*)
- Send *all* credentials over SSL/TLS
- Implement changing, unguessable, memorable, and definitive security questions



- Determines
  - Extent of fix
  - Level of Effort
  - Interaction w/ other systems
  - Regression/re-exploit potential

…On a Napkin


MODEL
VIEW
CONTROLLER
• STORES DATA
• ACID PROPERTIES
• RENDERS APP & DATA
• COLLECTS USER INPUT
• NAVIGATION & FLOW
• DISPATCH BUSINESS LOGIC
• AUTHZ (CRUD)
• FILTER & FORMAT OUTPUT
• HANDLE ERRORS
• AUTHN
• AUTHZ
• FILTER INPUT
• FORWARD & REDIRECT SECURELY

MVC Element	View		Controller		Model
Component	**Client-side Script**	**Decorator Servlet**	**Controller Servlet**	**Action Servlet**	**Persistent Store**
Responsibility	• Aspects of User experience	• Consuming and hiding error conditions • Filtering output in a target-specific fashion	• Authenticating requests • Filtering / validating input • Limiting user access rights to appropriate workflows • Dispatching actions	• Processing requests • Generating content • Redirecting sessions to different views • Coarse-grain transaction boundary	• ACID transaction properties • Hold data

To Security/Audit – Looks like Control Areas

- Input Validation
- Output Encoding
- Logging
- Masking
- Cryptography
- Authentication
- Authorization
- Session Management
- Debugging
- Handling of Resource Credentials

Place Controls W/in Design & Frameworks