RSA CONFERENCE 2014
FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.
Capitalizing on
Collective Intelligence

Mobile Application Assessment By The Numbers – A Whole-istic View

SESSION ID: MBS-F02

Dan Cornell
CTO
Denim Group
@danielcornell

Agenda

- Background
  - Mobile Application Threat Model
  - Assessment Methodology
  - Data Collected
- Findings
  - Types of Vulnerabilities Identified
  - Where Vulnerabilities Were Identified
  - How Vulnerabilities Were Identified

Background

Introduction

- Data comes from:
  - 61 Assessments
  - 20 Applications
- What we found:
  - 957 Vulnerabilities
- Assessment with the most vulnerabilities: 3 assessments had 10 Critical vulnerabilities
- Assessments with the least vulnerabilities: **only** three assessments had one vulnerability (all others had more)

Research Background

- Mobile application threat model
- Assessment methodology
  - Static versus dynamic testing
  - Automated versus manual testing
- Why CWE?
- Assessment data

Mobile Application Threat Model

- More complicated than a “typical” web application threat model
- Not just about code running on the device
- Main components:
  - Mobile application
  - Enterprise web services
  - 3rd party web services

Assessment Methodology

- Testing activities
  - Combination of both static and dynamic activities
  - Combination of automated tools, manual review of automated test results and manual testing
  - Tools include Fortify SCA, IBM Rational AppScan, Portswigger BurpSuite
- Scope can include:
  - Code running on the device itself
  - Enterprise services
  - 3rd party supporting services

Determining Severity

Based on customized DREAD model

- Damage potential
- Reproducibility
- Exploitability
- Affected users
- Discoverability

- Each factor ranked 1-3

Collapsed to single dimension

- Critical: > 2.6
- High: 2.3 – 2.6
- Medium: 2.0 – 2.3
- Low: < 2

Why CWE?

- Vulnerability taxonomy used was MITRE's Common Weakness Enumeration (CWE)
  - http://cwe.mitre.org/
- Every tool has its own "spin" on naming vulnerabilities
- OWASP Top 10 / WASC 24 are helpful but not comprehensive
- CWE is exhaustive (though a bit sprawling at times)
- Reasonably well-adopted standard
- Many tools have mappings to CWE for their results

Assessment Data

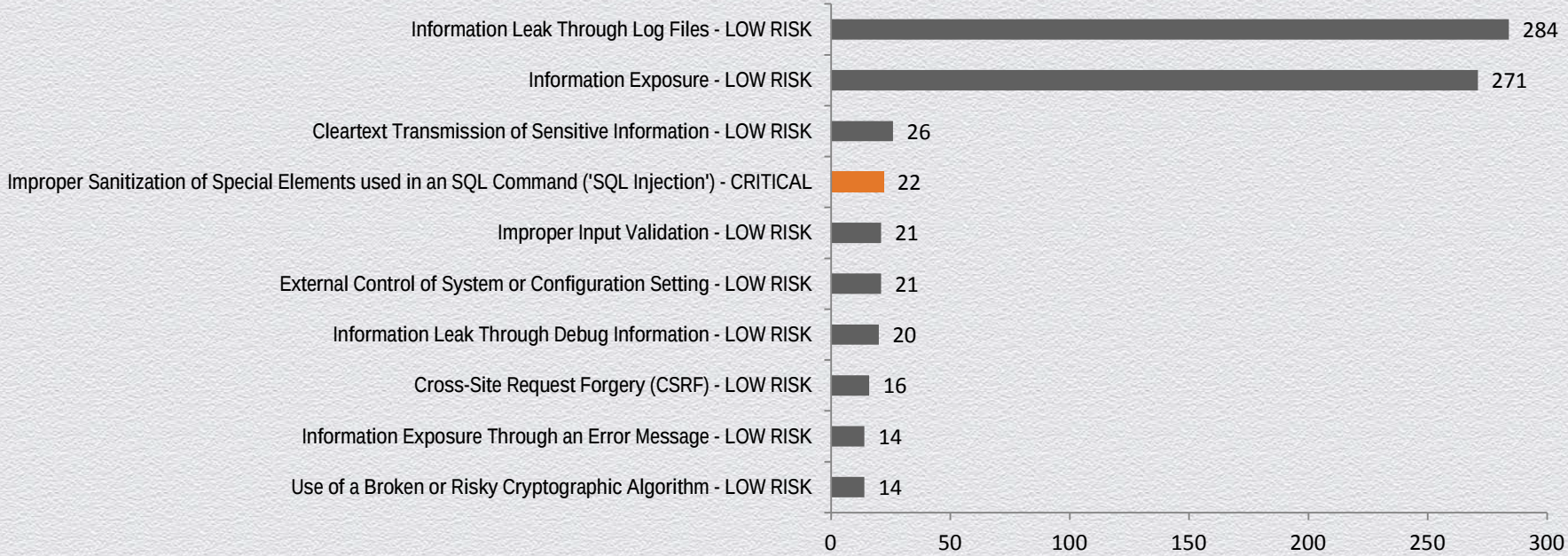
- Subset of mobile assessments
- Mostly customer-facing applications from financial services organizations
- Primarily iOS and Android applications
  - Some WAP, Windows Phone 7

What Did We Find?

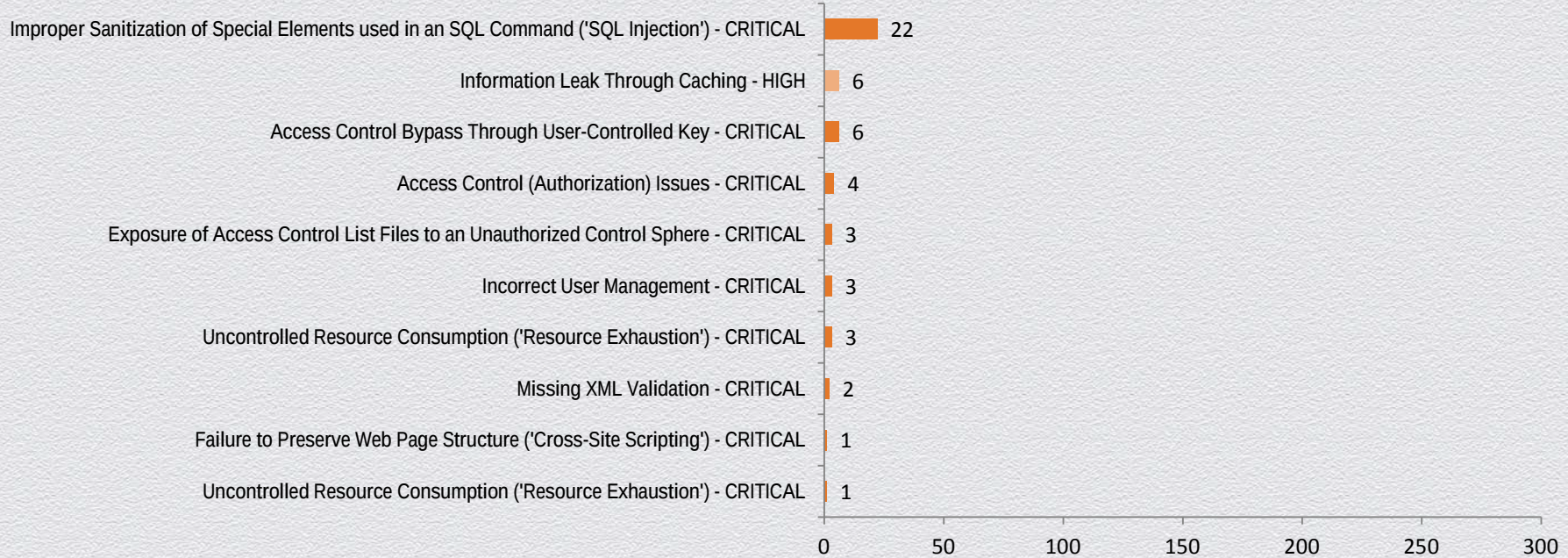
Types of Vulnerabilities Found

- Top 10 Most Prevalent CWEs – Overall
- Top 10 Most Prevalent CWEs – Critical/High Risk

Top 10 Most Prevalent CWEs – Overall



Top 10 Most Prevalent CWEs – Critical/High Risk



OWASP Top 10 Mobile Risks

- Similar to the OWASP Top 10 Web Application Risks, but targeted at mobile applications (obviously)
- Top risks to mobile applications:
  - https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_Ten_Mobile_Risks
- Work in progress to update this based on industry-contributed data

OWASP Top 10 Mobile Risks

M1: Insecure Data Storage

M2: Weak Server Side Controls

M3: Insufficient Transport Layer Protection

M4: Client Side Injection

M5: Poor Authorization and Authentication

M6: Improper Session Handling

M7: Security Decisions Via Untrusted Inputs

M8: Side Channel Data Leakage

M9: Broken Cryptography

M10: Sensitive Information Disclosure

Compare to OWASP Top 10 Mobile Risks

Strong Overlap

- Weak server-side controls
- Poor authentication and authorization
- Security decisions via untrusted inputs
- Sensitive information disclosure

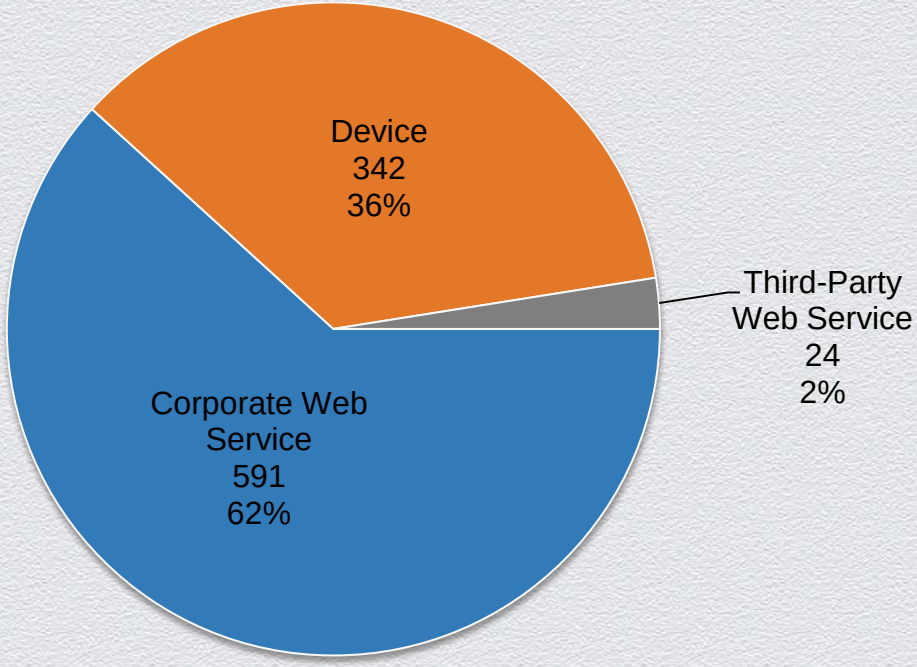
Overlap

- Insecure data storage
- Insufficient transport layer data protection
- Improper session handling
- Side channel data leakage
- Broken cryptography

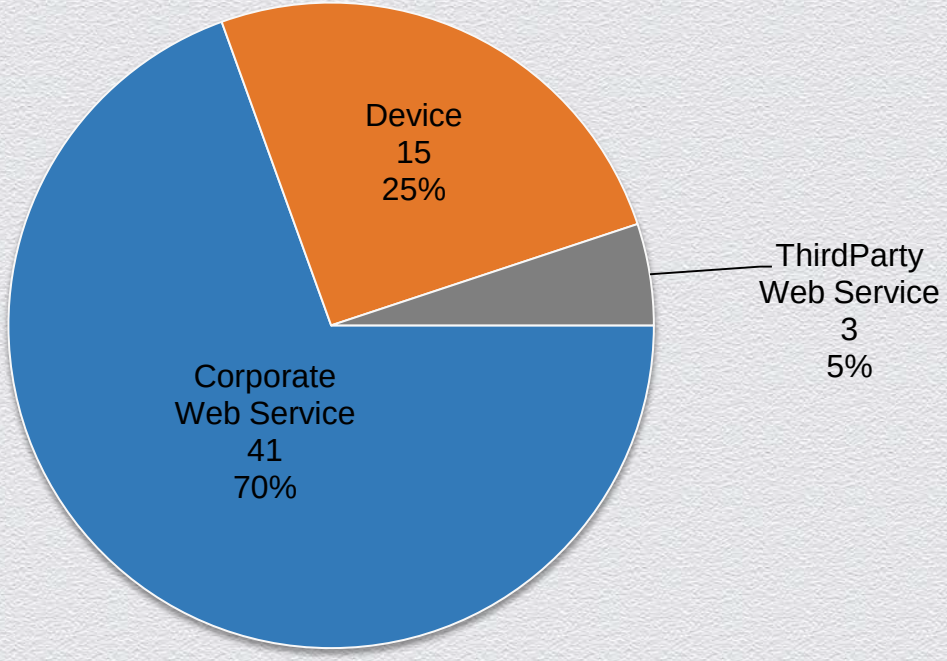
Weak Overlap

- Client-side injection

Where Did We Find Overall Vulnerabilities?

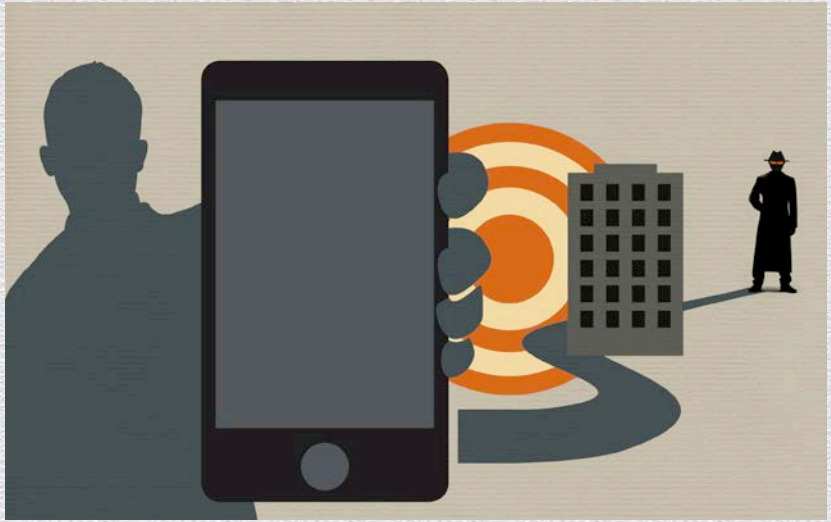


Where Did We Find Critical/High Risk Vulnerabilities?



Analysis of “Where” Data

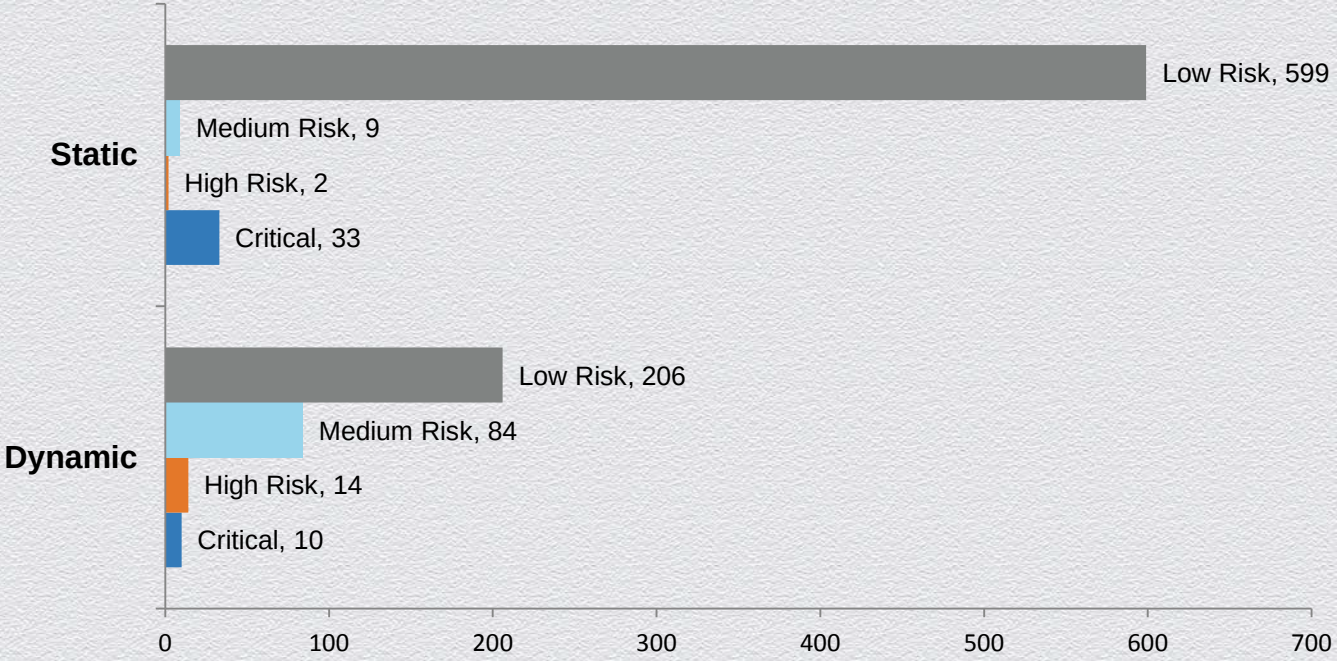
- Mobile security is about more than the code running on the device
- The things we really care about (Critical, High) are most frequently found on corporate web services
  - Then on the device
  - Then on 3rd party web services
- Reflects the “scale” benefits of finding web services vulnerabilities



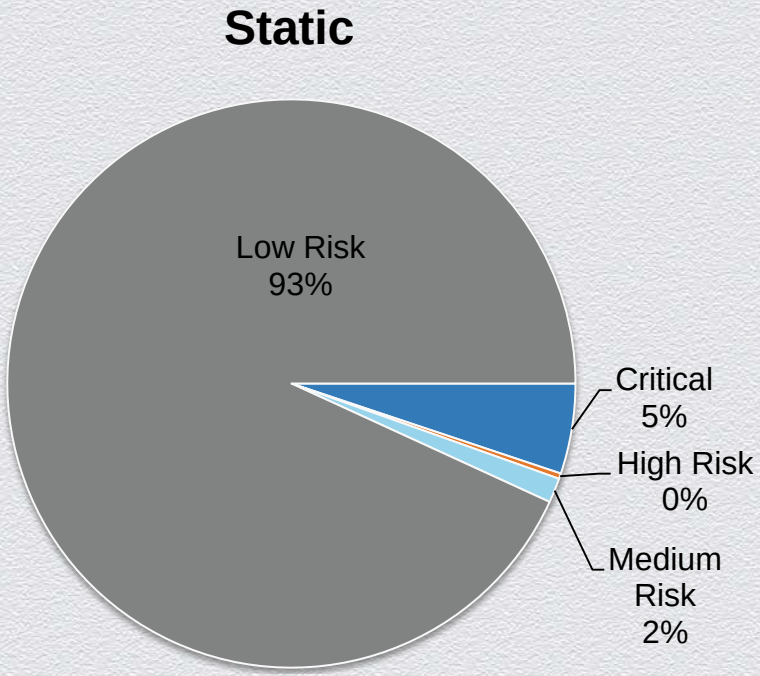
How Did We Find Vulnerabilities?

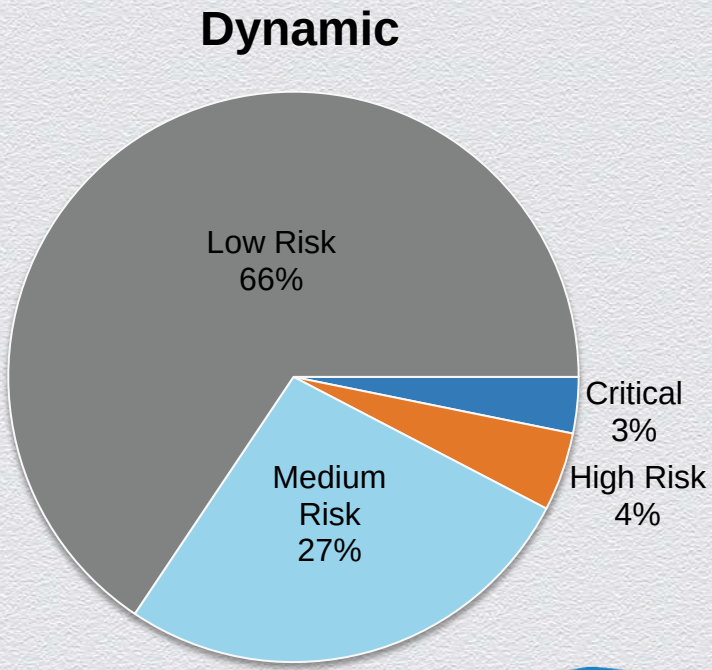
- Static vs. dynamic testing
- Automated vs. manual testing
- What techniques identified the most vulnerabilities?
- What techniques identified the most serious vulnerabilities?

Static vs. Dynamic Method of Finding Vulnerabilities



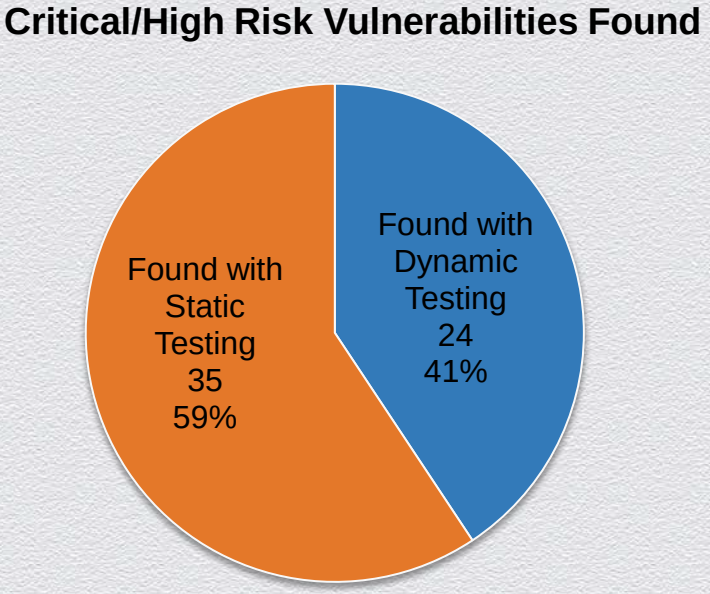
Static vs. Dynamic Method of Finding Vulnerabilities



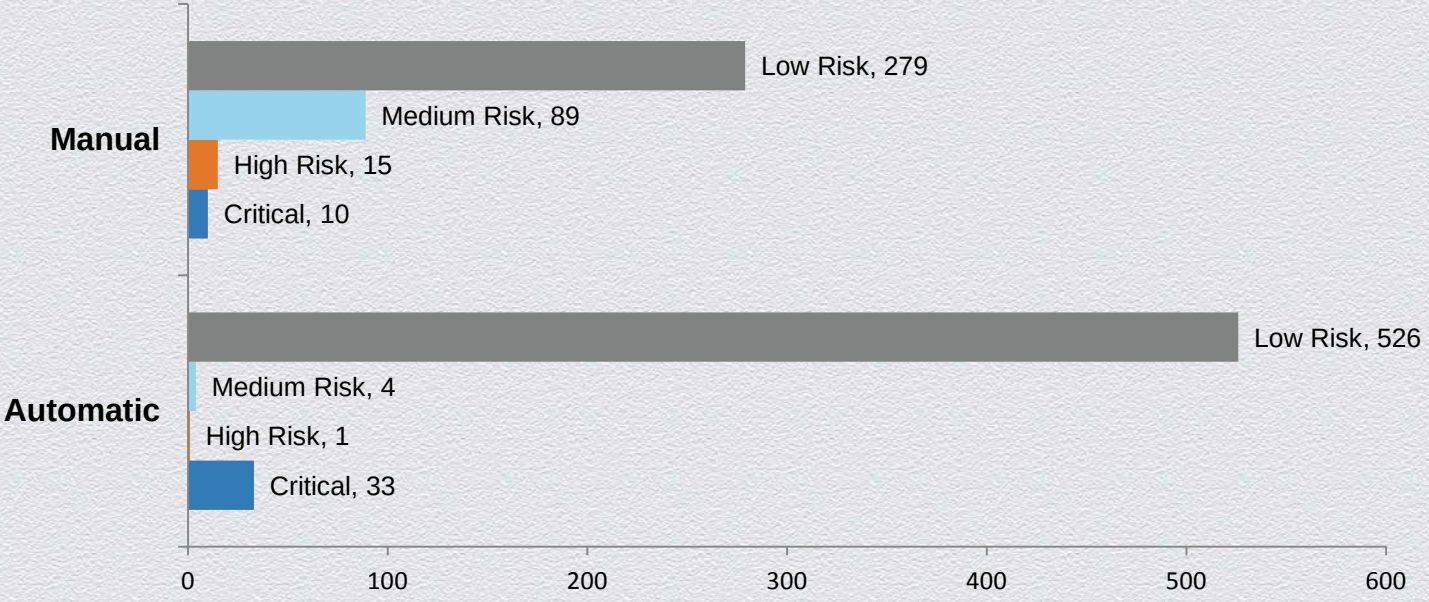


Critical and High Risk Vulnerabilities

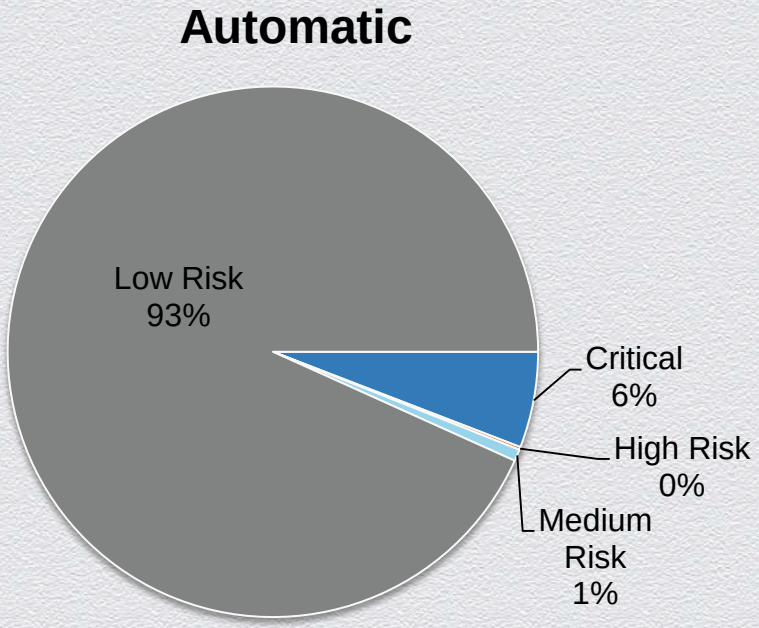
- Static testing was more effective when finding serious (Critical and High) vulnerabilities
- But it also found a **lot** of lower-risk vulnerabilities (as well as results that had to be filtered out)

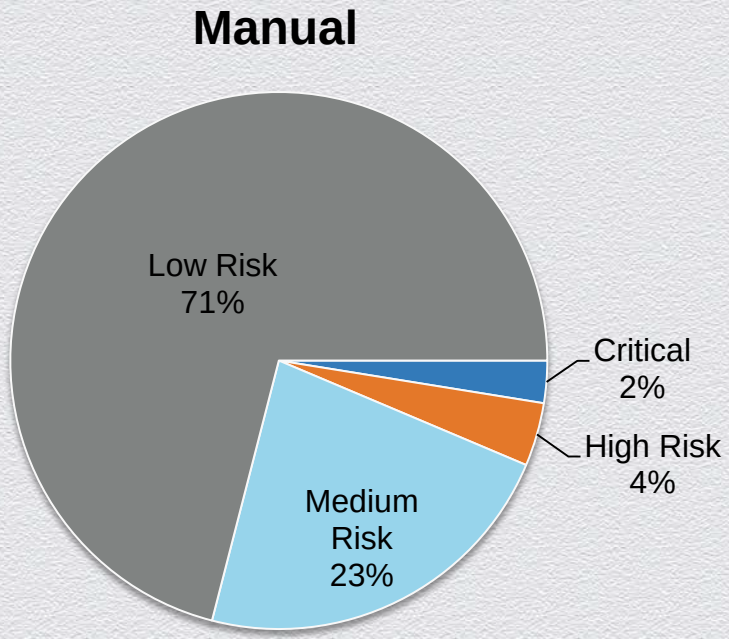


Automated vs. Manual Method of Finding Vulnerabilities



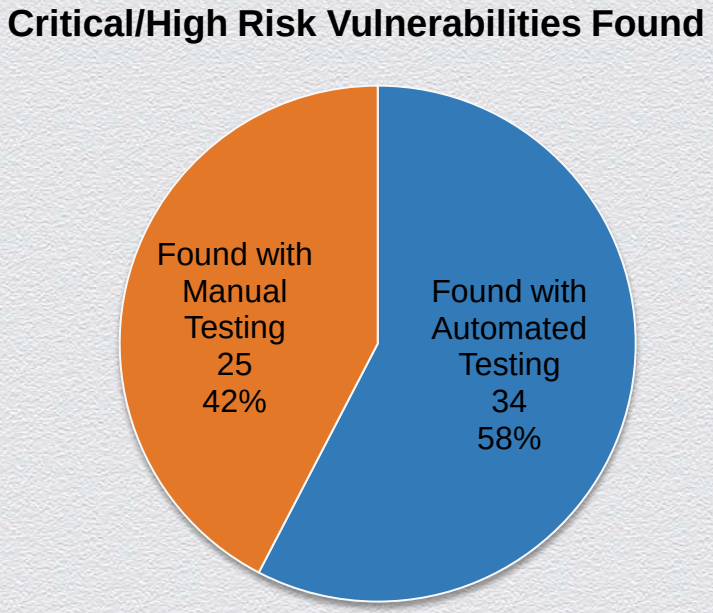
Automated vs. Manual Method of Finding Vulnerabilities



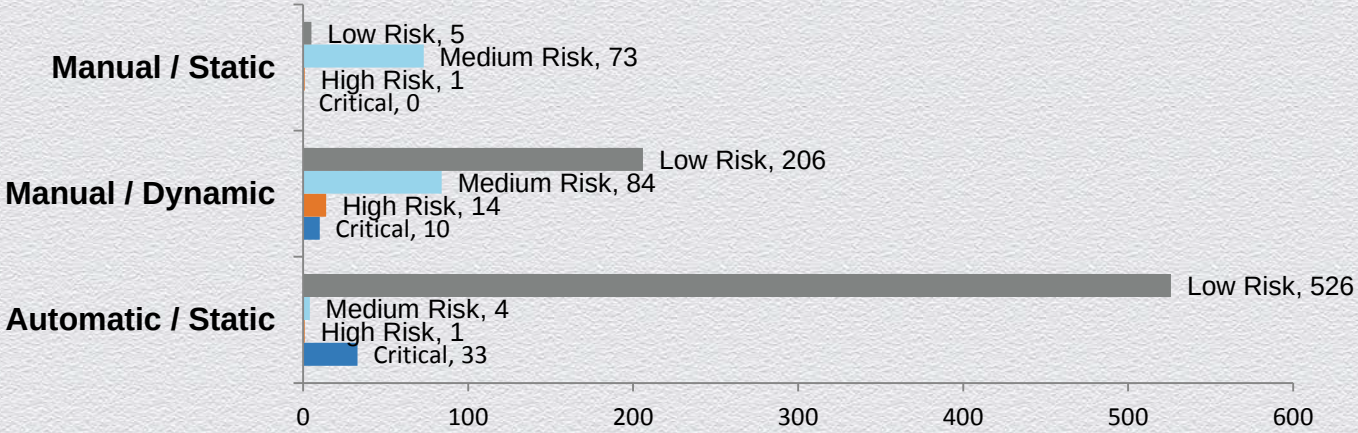


Automated vs. Manual Method of Finding Vulnerabilities (Critical and High)

- Automated testing was more effective when finding serious (Critical and High) vulnerabilities

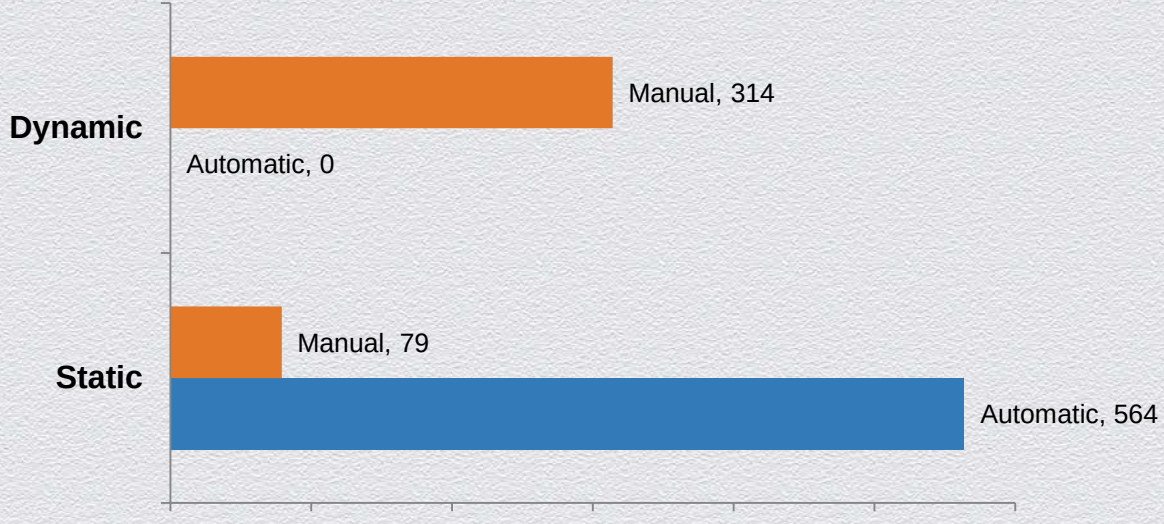


Automated vs. Manual, Static vs. Dynamic Methods



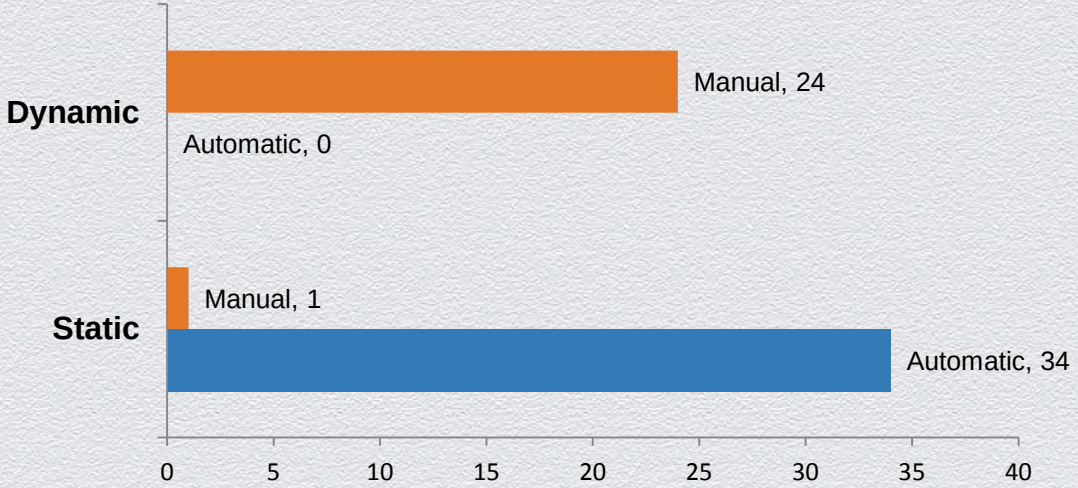
	Automatic / Static	Manual / Dynamic	Manual / Static
Low Risk	526	206	5
Medium Risk	4	84	73
High Risk	1	14	1
Critical	33	10	0

Automated vs. Manual, Static vs. Dynamic Methods



	Static	Dynamic
Manual	79	314
Automatic	564	0

Automated vs. Manual, Static vs. Dynamic for Critical and High Vulnerabilities



	Static	Dynamic
Manual	1	24
Automatic	34	0

Analysis of “How” Data

- A comprehensive mobile application security assessment program must incorporate a significant manual testing component
- Automated tools for testing mobile applications are not as mature as those for testing web applications
- Web services can be challenging to test in an automated manner

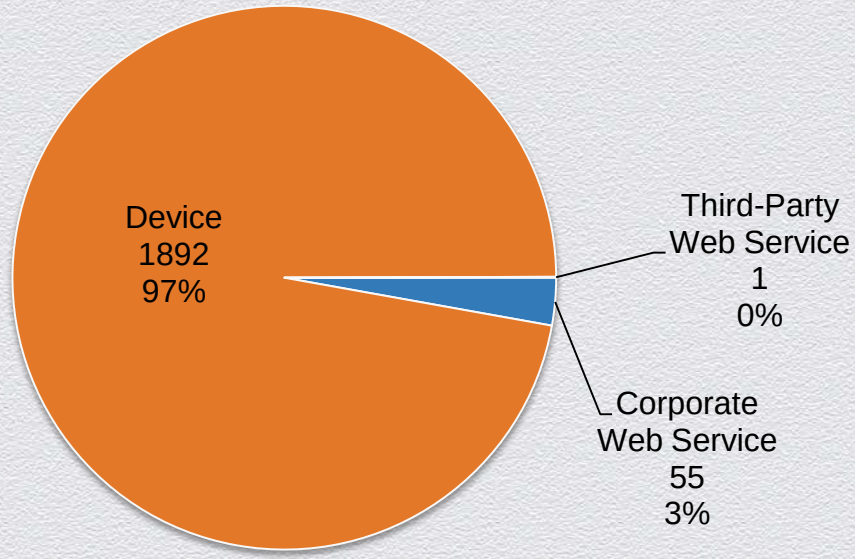
On-Device Vulnerabilities By Platform

Platforms	Number of Assessments on Device	Number of Total Vulnerabilities on Device	Average Number of Vulnerabilities Found per Assessment
iOS	39	252	6.5
Android	19	84	4.4
Windows Phone 7	1	3	3
WAP	1	3	3

Other Observations

- We also include “other observations” as part of our assessments
- These reflect:
  - Application weaknesses
  - Coding flaws or behavior that are not “best practice” but do not reflect an immediate, exploitable vulnerability
- We had 1,948 “other observations”
  - Roughly twice as many as actual vulnerabilities

Other Observations – Where Were They Found?



What Does This Mean?

- Most of these “other observations” are about code on the device
  - Mobile application developers need help building better code
  - AND automated code scanning tools need to be better about filtering less valuable results
- Something that is not a problem today could be later on
  - Identification of new platform vulnerabilities
  - Changes coming along with a new application release

Conclusions

- What To Test?
  - Mobile “apps” are not standalone applications
  - They are systems of applications
  - Serious vulnerabilities can exist in any system component
- How To Test?
  - Mobile application testing does benefit from automation
  - Manual review and testing is required to find the most serious issues
  - A combination of static and dynamic testing is required for coverage

Recommendations

- Plan your mobile application assessment strategy with coverage in mind
- Evaluate the value of automation for your testing
  - More “cost” than simply licensing – deployment time and results culling
- Look for opportunities to streamline
  - Fast application release cycles can require frequent assessments
  - Control scope:
    - Assess application changes (versus entire applications)
    - Manage cost of reporting

Next Steps (For Us)

- Incorporate more assessment data
- Possible collaboration with OWASP Top 10 Mobile Risks
  - Currently being reworked based on data sets such as ours
- Better analysis of applications over time