Dan Geer

geer@stake.com
+1.617.768.2723

Art v. Science

Characterization and Specialization

Time Line and Drivers

Put up or shut up...

Applications are where the action is

- Security trends say so
- Business realities say so
- Risk management needs quantitative decision support
- Application pen-tests can yield that support

Security trend 1
Applications are federating

- **Distributed applications have multiple security domains**
  - **The firm**: client service & administrative functions
  - **External providers**: front-end Web farms and application hosting
  - **Partner interfaces**: data streams (inventory, payment, real-time feeds)
- **Applications get ever more moving parts**
  - Mainframe → client-server → *n*-tier → Model 2 (J2EE and .Net)
- **Network service stratification**
  - Bandwidth, hosting, provisioning, delivery

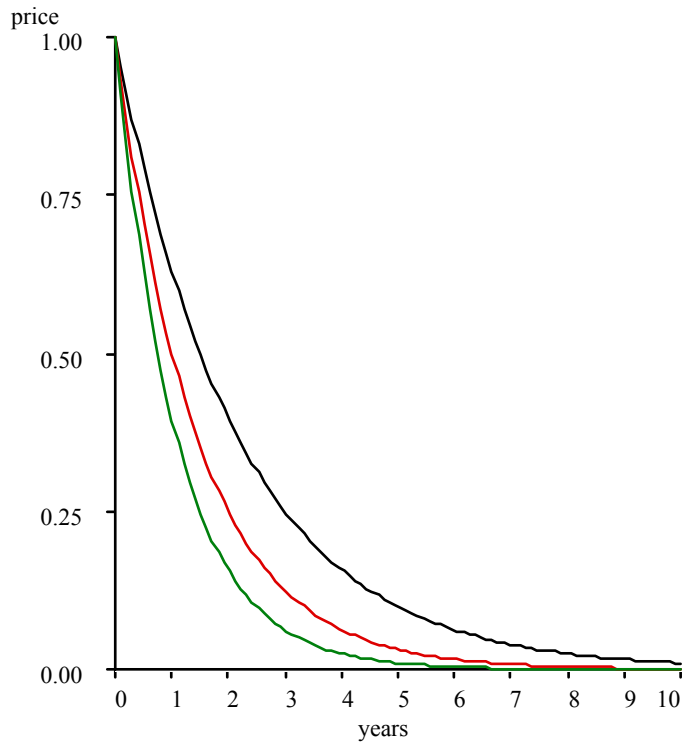
Security trend 2
Perimeter defense is increasingly diseconomic

- **"Shared wire" supplants "shared model"**
  - XML is the great equalizer
  - SOAP and XML-RPC specifically designed to go through firewalls
  - Emerging web services
- **Firewalls stop nuisance attacks, not application traffic**
  - Everyone leaves ports 80 and 443 open
- **As a result, the threat model mutates**
  - More attacks through HTTP, at application level
  - More attacks targeted at specific application components
  - Attacks on applications require lower skill levels

Security trend 3
Data, data everywhere

- **Data storage needs increasing exponentially**
  - More new data produced in next 3 years than in all of human history
  - Corporate IT spending 4% in 1999 v. 17% in 2003 (Forrester)
- **Form factors proliferating**
  - Local storage
  - Storage arrays
  - Appliances/network-attached storage

Moore's Law, 18mo doubling
Storage, 12mo doubling
Bandwidth, 9mo doubling



Corresponding business realities

- **Risk management has won**
- **Anticipate failure or be damned**
- **Demand for security expertise exceeding supply**

But most importantly,

- **The future belongs to the quants**

Quantitative decision support for risk management

- **Annualized Loss Expectancy**

  = ∑ (probability * business impact) } **Before investment, and after**

- **Net Present Value**

  Increased Revenues
  - Improved Uptime
  - Transactional Frequency
  - New Referrals

  Decreased Direct Costs
  - Developer Re-work
  - System Administrator Labor
  - Patch Release Costs
  - Customer Retention

  Cost Avoidance (soft costs)
  - Media/Legal

  } **Future cash flows discounted by cost of funds**

= Net Investment Return

Treat application security as you would quality

Relative cost to fix issues, by stage

Design	1
Implementation	6.5
Testing	15
Maintenance	100

Source: *Implementing Software Inspections*, IBM Systems Sciences Institute, IBM, 1981

Software development costs, by stage

Design	15%
Implementation	60%
Testing	25%

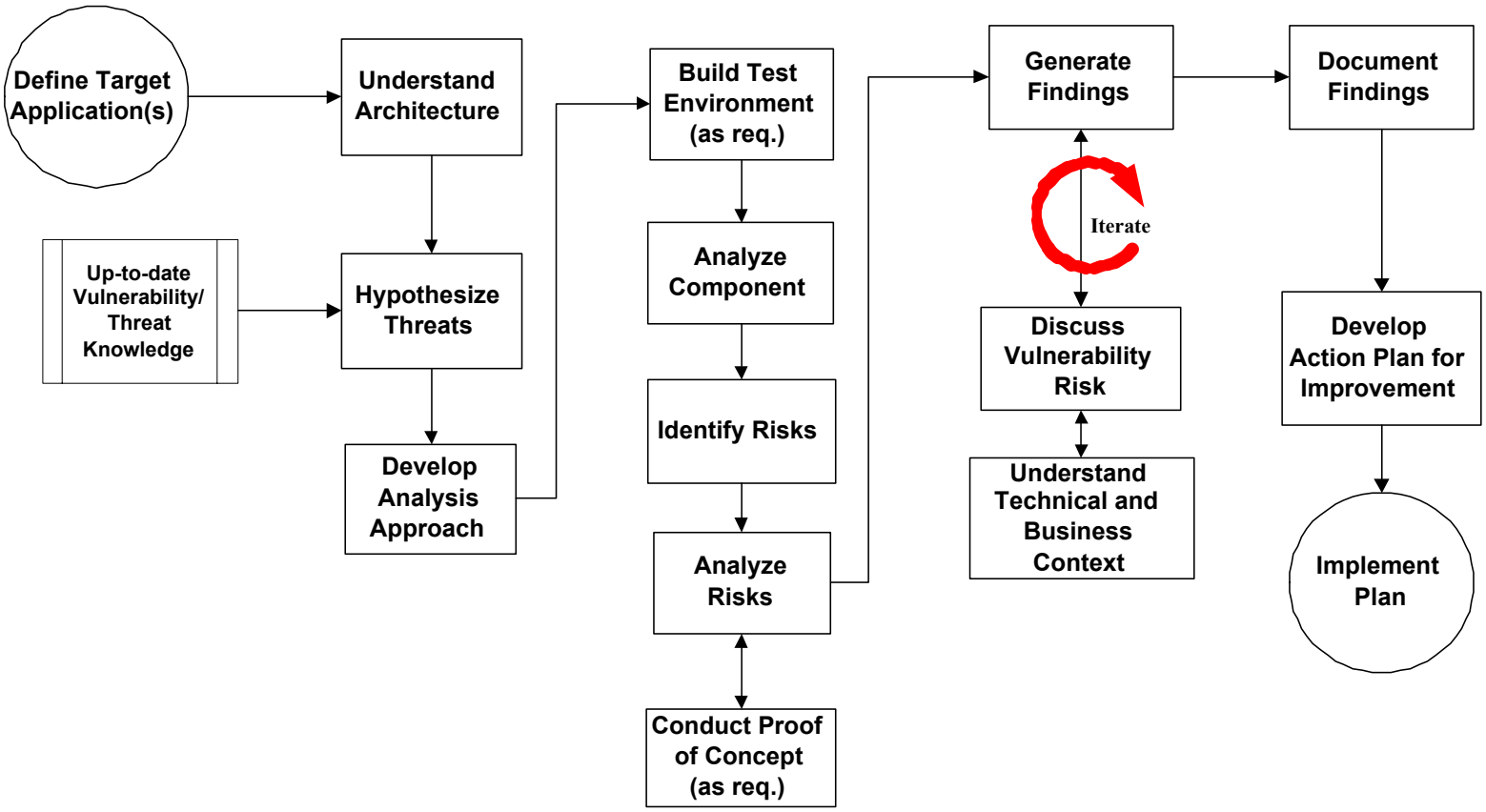
Source: *Architectures for Software Systems*, course Notes, Garlan & Kazman, CS, CMU, 1998

A little example of pooled data

Security evaluation of major applications treated as a source of summary numbers and shared intelligence

All data are real, pooled and hence anonymized within a trust relationship, and modeled as normative

Application Penetration Testing Approach



Finding 1/4: *Security defects are common*

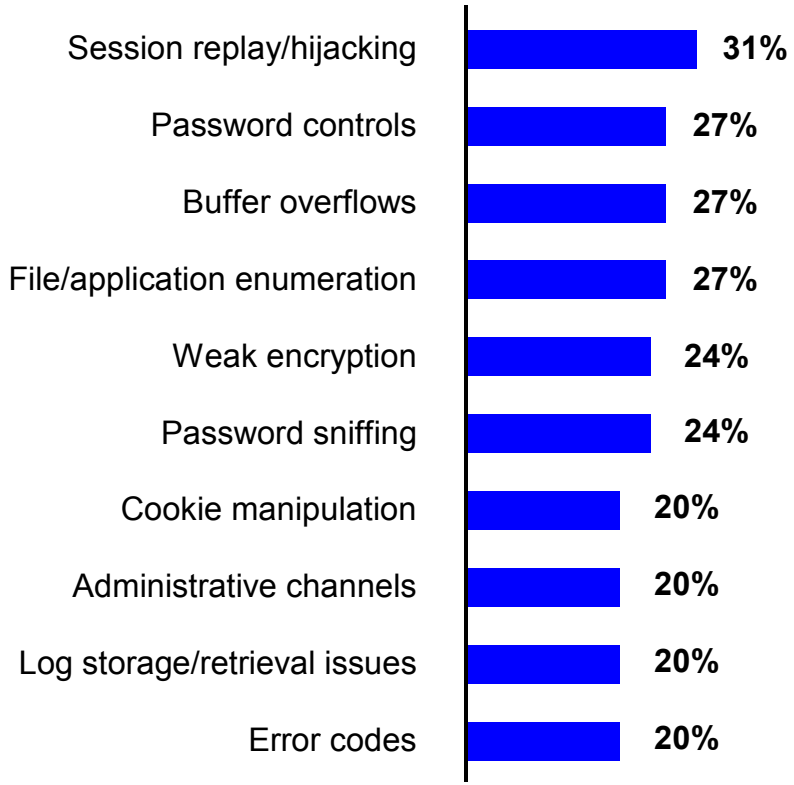
Security Defects by Category

Category	Engagements where observed	Design related	Serious design flaws*
Administrative interfaces	31%	57%	**36%**
Authentication/access control	62%	89%	**64%**
Configuration management	42%	41%	**16%**
Cryptographic algorithms	33%	93%	**61%**
Information gathering	47%	51%	**20%**
Input validation	71%	50%	**32%**
Parameter manipulation	33%	81%	**73%**
Sensitive data handling	33%	70%	**41%**
Session management	40%	94%	**79%**
Total	**45**	**70%**	**47%**

*Scores of 3 or higher for exploit risk *and* business impact

Source: 2002 @stake - The Hoover Project (n=45)

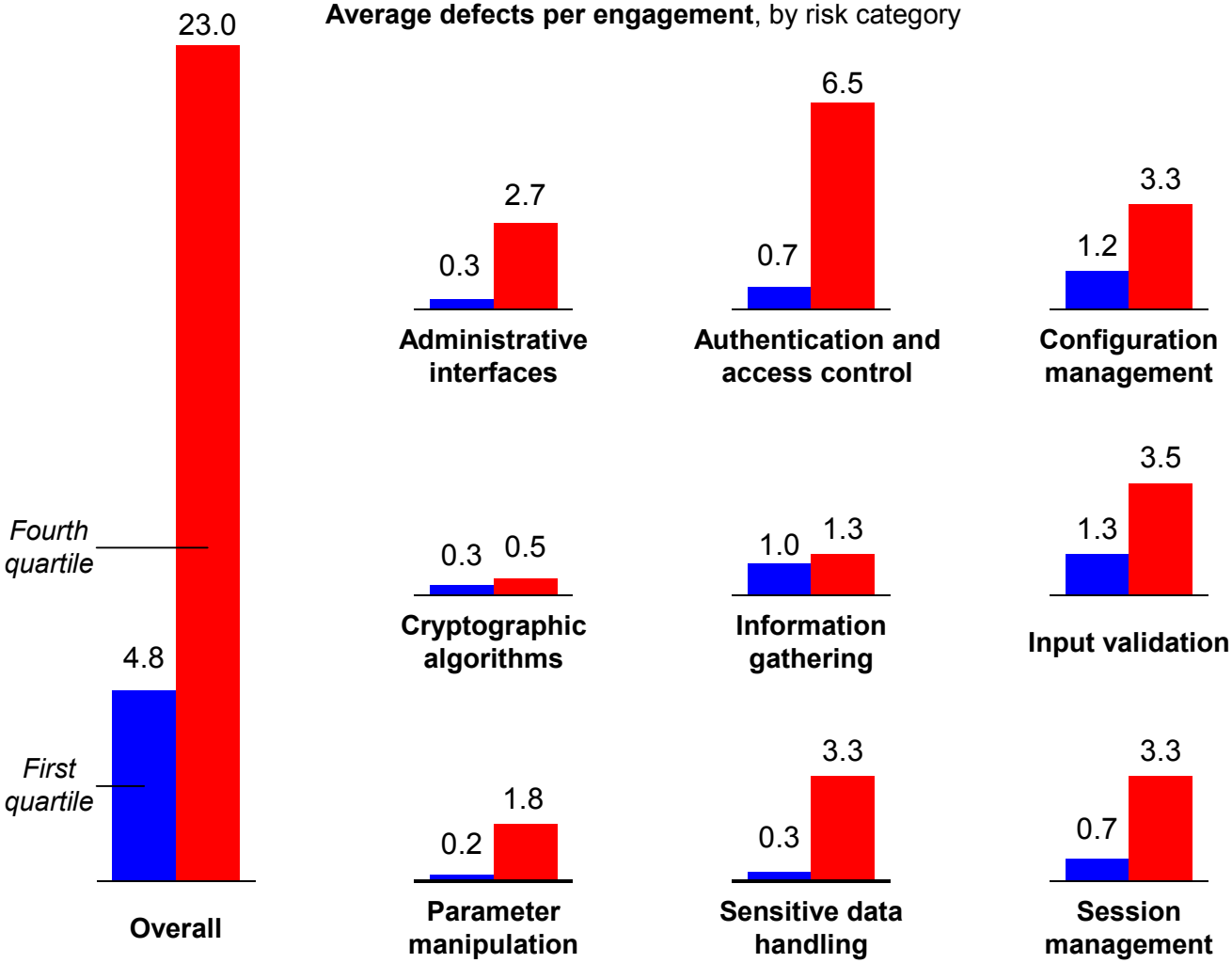
Top 10 Application Security Defects



Assessments where encountered, percent

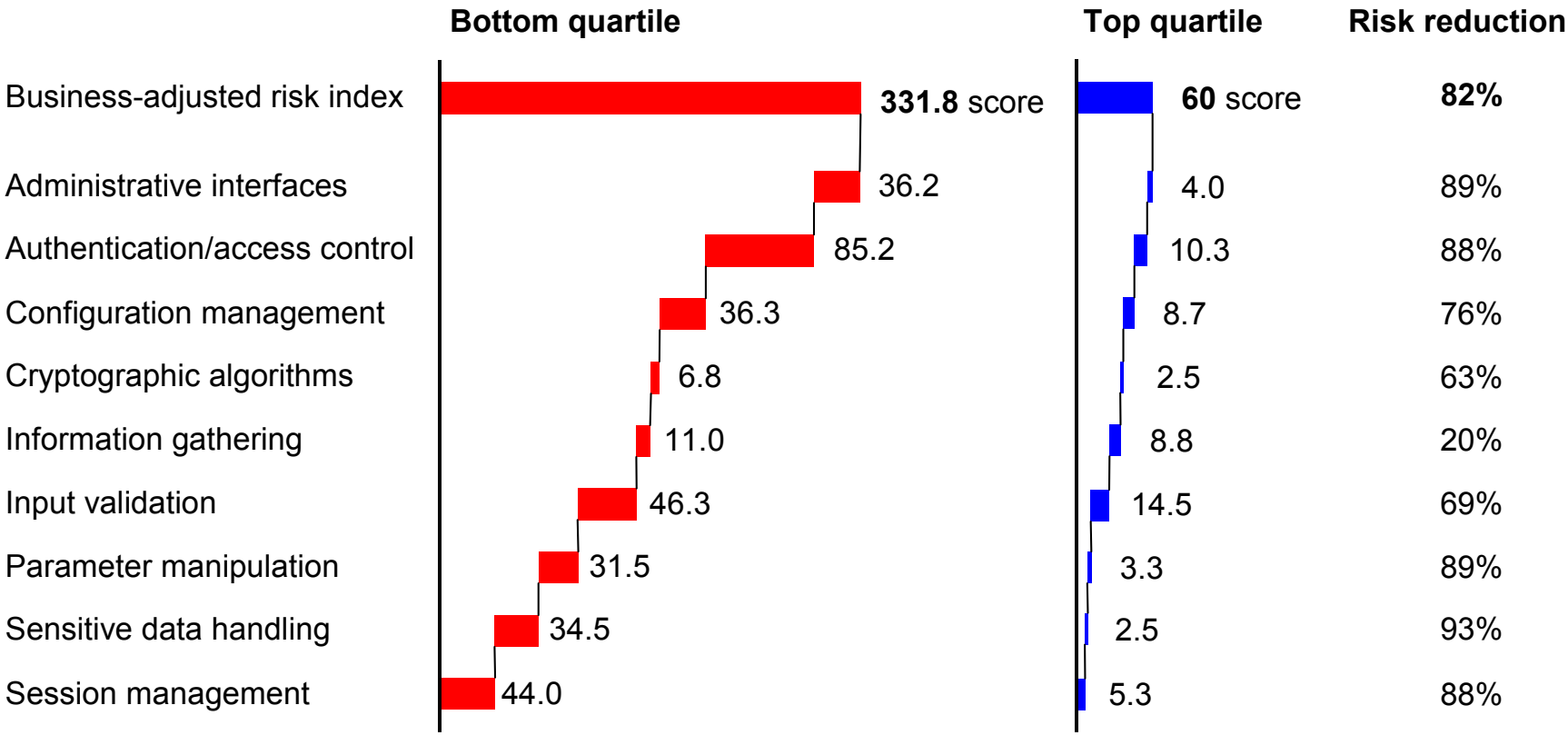
Finding 2/4: *Leaders have fewer defects*

Average defects per engagement, by risk category



Source: 2002 @stake - The Hoover Project (n=23)

Finding 3/4: *Leaders carry less risk*

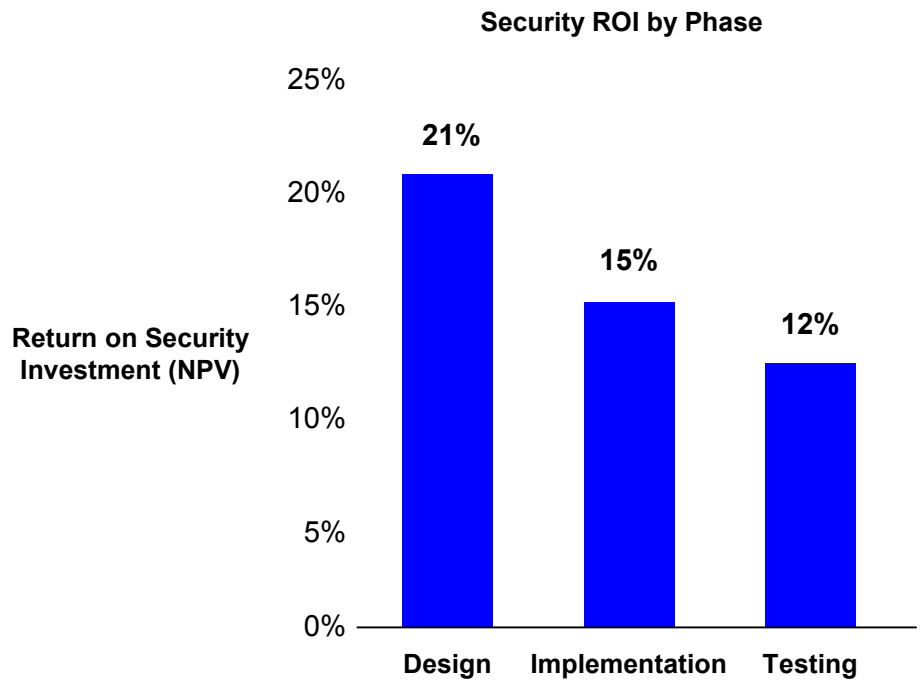


Average business-adjusted risk (BAR) index per engagement, with breakdown by risk category

Source: 2002 @stake - The Hoover Project (n=23).
BAR index = sum of all defects' individual BAR scores, where each defect's score = exploit risk (5 point scale) x business impact (5 point scale).

Finding 4/4: *Fixing security defects earlier pays off*

- **Although benefits can be found throughout the lifecycle, earlier involvement is most beneficial**
- **Vulnerabilities are harder to address post-design**
- **System-wide changes may be required at later stages**
- **Enabling improvements can be made at design state**



Source: 2002 @stake - The Hoover Project

Repeating: *Applications are where the action is*

- **Security trends say so**
- **Business realities say so**
- **Risk management means quantitative decision support**
- **Application pen-tests can yield that support**

And if they don't, what's the point?

Questions?