SOFTWARE SECURITY ASSURANCE SUMMIT

September 12, 2011 | HP Protect at Gaylord National | Washington D.C.

Evolution of Application Security

From Breach to Mobile Applications

John South
Chief Security Officer
Heartland Payment Systems

T.E.N. TECH EXEC NETWORKS *presented by* hp HP Enterprise Security

- Who is Heartland Payment Systems?
- Overview of the Breach
- Strategic Asymmetry
- Securing the Application Threat Space
- Securing the Mobile Threat Space
- Partnering for Success

- Publicly traded, NYSE: HPY
- *FORTUNE 1000* company
- Fifth largest processor in the US
- Processes close to 11 million transactions a day
- Serves more than 250,000 businesses nationwide
- More than 2,700 employees
- Ten offices throughout the US and Canada

- Credit/debit/prepaid card processing
- Mobile payments
- E3™ technology
- Payroll services
- Gift marketing and loyalty programs
- Check management
- Online payments
- Give Something Back Network OneCard
- MicroPayments
- K-12 school lunch payments

- Major markets served:
  - Restaurant
  - Lodging
  - Healthcare
  - Retail
  - Petroleum
  - Community Banks

Overview of the Breach

- Very Late 2007 – SQL Injection via a customer-facing web page in our corporate (non-payments) environment. Bad guys were in Heartland’s corporate network.
- Early 2008 – Hired largest approved QSA to perform penetration testing of corporate environment
- Spring 2008 – CEO learned of sniffer attack on Hannaford’s, created a dedicated Chief Security Officer position and filled that position
- April 30, 2008 – Passed sixth consecutive “Annual Review” by largest QSA
- Very Late 2007 – Mid-May 2008 – Unknown period but it is possible that bad guys were studying the corporate network
- Mid-May 2008 – Penetration of Heartland’s payments network

- Late October 2008 – Informed by a card brand that several issuers suspected a potential breach of one or more processors. We received sample fraud transactions to help us determine if there was a problem in our payments network. Many of these transactions never touched our payments network.
- No evidence could be found of an intrusion despite vigorous efforts by Heartland employees and then two forensics companies to find a problem.
- January 9, 2009 – We were told by QIRA that "no problems were found" and that a final report reflecting that opinion would be forthcoming.
- January 12, 2009 – January 20, 2009 – Learned of breach, notified card brands, notified law enforcement and made public announcement.

Strategic Asymmetry

A One-sided Game

- **SQL injection via a customer-facing web page in our corporate (non-payments) environment. Bad guys were in our corporate network**

- Why are applications the targets *du jour*?
    - Network and device security have been focus of vendors and security teams for a number of years
    - Applications are often portals
        - Directly to sensitive data itself, or
        - Unknowingly, to soft underbelly of internal network
- Applications used to be much less of a threat

- This is a classic case of manipulating a strategic asymmetry
  - Strategic use of asymmetric technologies to exploit asymmetric advantages and counter asymmetric weaknesses*
  - Two sides in the battle
    - Corporations, medium-sized enterprises, small businesses, individuals, vs.
    - Professional cybercriminals

- Though not captured in these terms in the past, this is the classic information security struggle – though evolved

*See Nshetri, Kir, The Global Cybercrime Industry, Chapter 6. Springer-Verlag. Pg 119

- Corporations, medium-sized enterprises, small businesses, individuals
  - Large, diverse networks
  - Often multiple hierarchies of responsibility and accountability
  - Constrained by budgets, SLAs, project delivery deadlines and limited human capital

vs.

- Professional cybercriminals who, in almost all cases, are:
  - Very intelligent (at least of their subject matter) and better trained
  - Better financed
  - Better prepared
  - Have a time advantage
  - And … have nation-state protection

- Who are the Bad Actors?
  - Cybercriminals
    - Crime “families” – Russian Business Network
    - Specialists – Bot herders
  - Cyberterrorists
    - Stuxnet
    - Hydraq
  - Hactivists
    - Attacks against military and intelligence organizations
    - Corporations (particularly those who impact their funding model)
- What do each of these have in common?
  - Extensive target research
- Malicious insiders

- Rub of strategic asymmetry
  - Entities least prepared to establish a strong defensive position are least prepared to establish proactive threat modeling
  - With today’s threat space:
    - You cannot fight something if you cannot see it
    - You cannot prevent something if you cannot predict it
    - You cannot secure something that was not built to be secure*
- In our case, the application that was breached was compliant with its functional specifications

*Roger Thornton, CTO & Founder, Fortify Software, Presentation at the 2011 BITS-FS-ISAC Conference, “Increase Your Security Intelligence: Manage Application Security in Context with the Business”.

Securing the Application Threat Space

Where Heartland Found Itself

- Software paradigms have evolved from computer-centric to very distributed models over time
  - Evolving and expanding attack surface
- Another classic example of asymmetry
  - In order to do business, applications and portals have to be:
    - Easily accessible
    - Easy to use
    - Operate transparently to users
  - Expands security scope and oversight
- Adage – "company has to find all security holes in the applications and portals, malicious actors only have to find one"

- You cannot fight something if you cannot see it – visibility
  - First part of the problem for Heartland was two-fold
    - What applications are on our networks?
      - External facing
      - Internal-only
    - Which applications are problematic from security perspective?
  - What access models were being used by various apps?

- Visibility to the application threat space is a critical first step
  - Have to look at all applications
  - Utilities, business intelligence apps, etc.

- How complex is our application security space today?
- Complete a full inventory of application space
  - Internal- vs external-facing applications
  - PC vs mobile platforms
  - Software as a Service
  - Application ownership
  - Authentication mechanisms
  - Account maintenance
- Completely documented data flows
  - Transmission of data
  - Data stores
  - Access to data

- Application Security Framework
  - Developed a baseline of secure coding functionality to be incorporated into coding
  - Requirements grouped by type of application being developed
    - Application Security Baseline – apply to all applications
    - Browser-based Application Baseline – apply to web applications
    - Web Service Application Baseline – apply to all web services
    - Confidential: Restricted Baseline – apply to all applications that store, process, or forward Confidential: Restricted information
  - Trained all developers on the Framework
  - Software leads have first line responsibility that developers adhere to Framework
- Framework a functional part of the SDLC

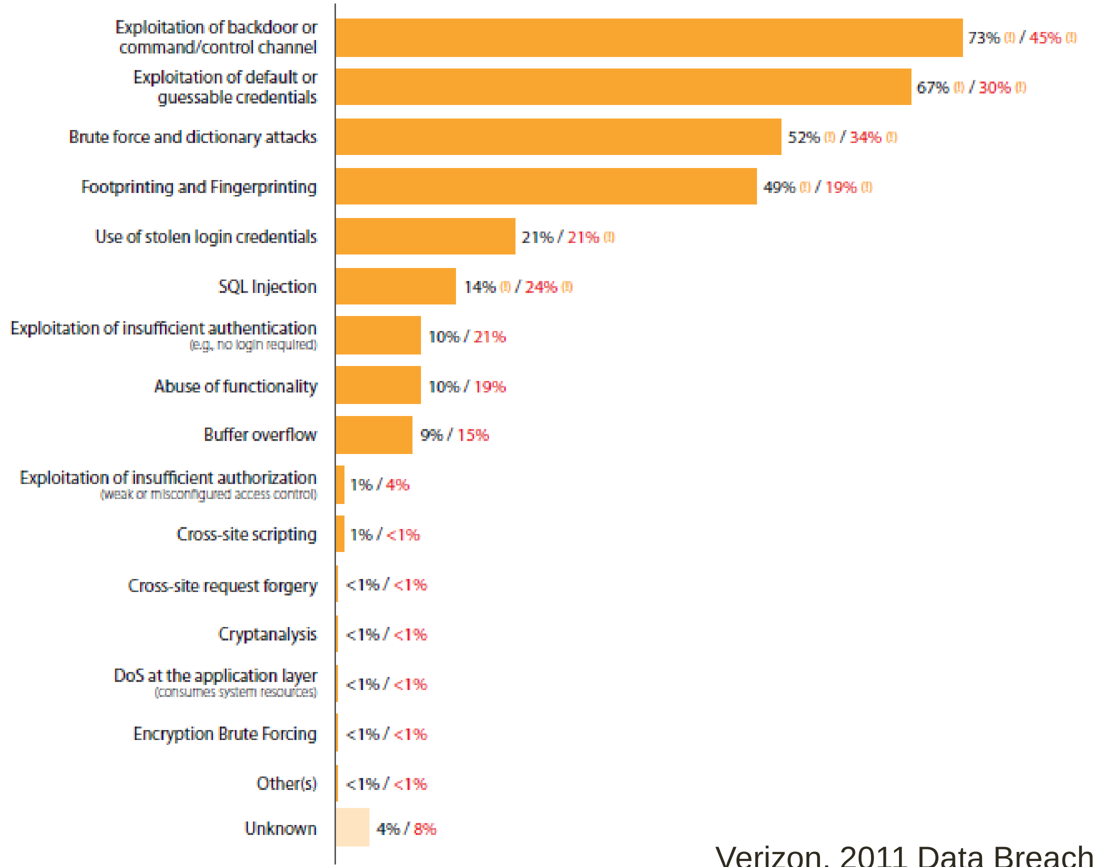
- You cannot prevent something if you cannot predict it – predictability
  - Look to analytics to increase knowledge of threats
  - Ties threat space to the threats that may impact it

- Number of sources of threat intel
  - Much of information is publicly available (but needs to be current)
  - Threat intel specific to your industry – FS-ISAC is an example
  - Important to develop relationships with local and federal law enforcement
    - Some portion of our personnel need to be cleared for this to be effective
    - No need for attribution

Software Security Assurance Summit

September 12, 2011 | HP Protect at Gaylord National | Washington D.C.

Figure 22. Types of hacking by percent of breaches within Hacking and percent of records



Verizon, 2011 Data Breach Investigations Report, pg 32

presented by

HP Enterprise Security

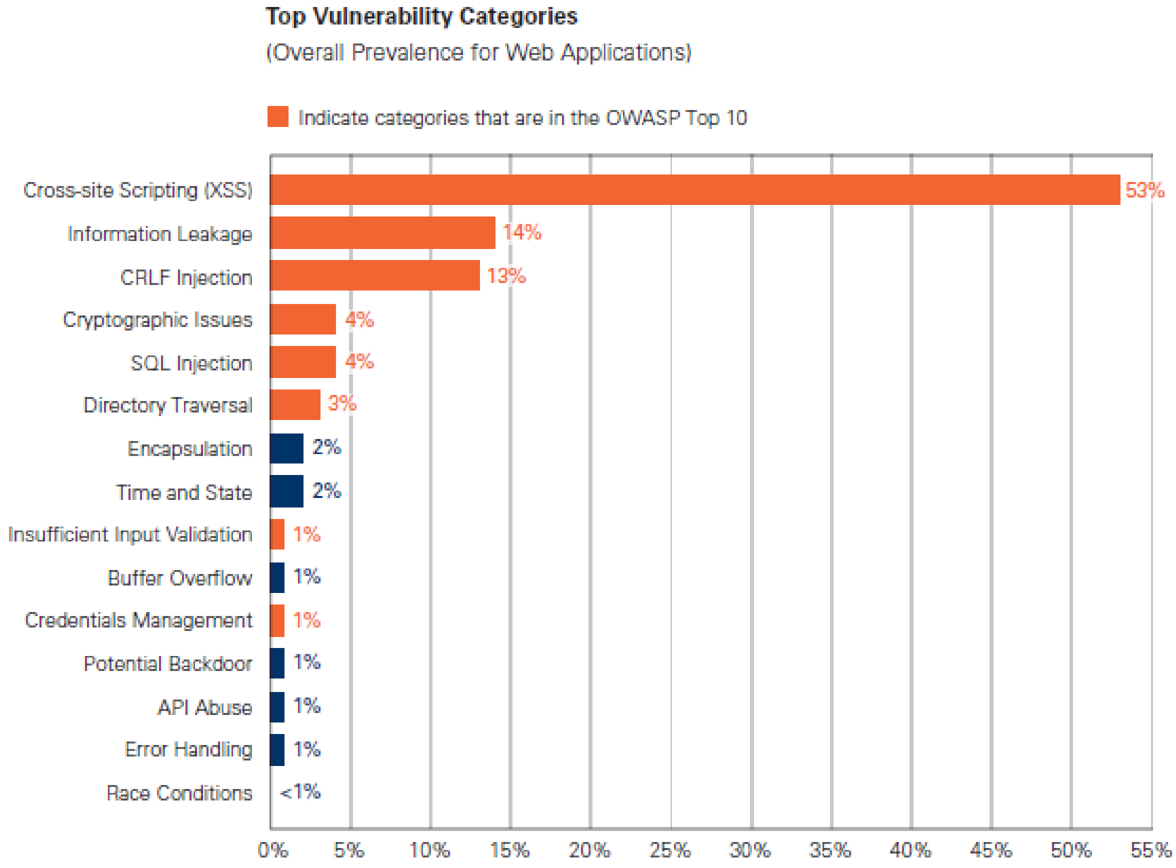
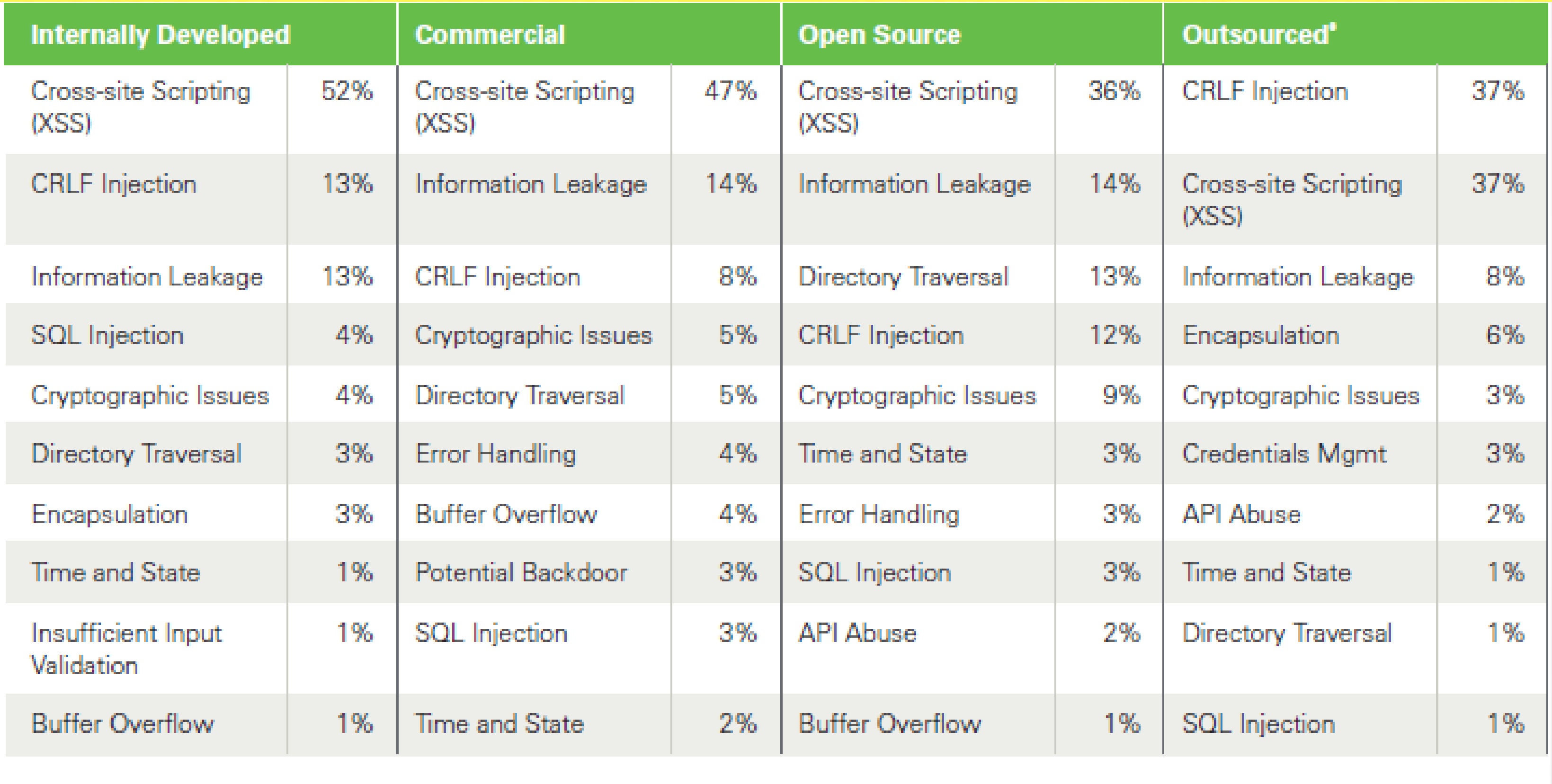


Figure 17: Top Vulnerability Categories (Overall Prevalence for Web Applications)

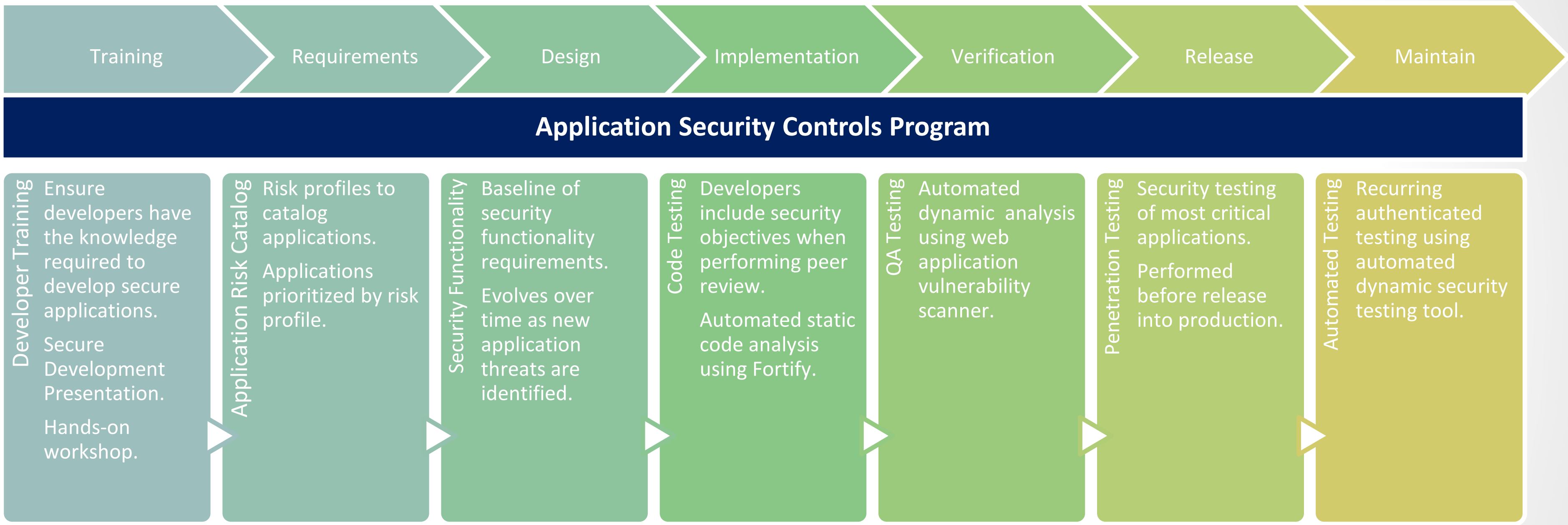
Veracode, State of Software Security Report: The Intractable Problem of Insecure Software, Apr 2011, pg 25



Internally Developed		Commercial		Open Source		Outsourced*	
Cross-site Scripting (XSS)	52%	Cross-site Scripting (XSS)	47%	Cross-site Scripting (XSS)	36%	CRLF Injection	37%
CRLF Injection	13%	Information Leakage	14%	Information Leakage	14%	Cross-site Scripting (XSS)	37%
Information Leakage	13%	CRLF Injection	8%	Directory Traversal	13%	Information Leakage	8%
SQL Injection	4%	Cryptographic Issues	5%	CRLF Injection	12%	Encapsulation	6%
Cryptographic Issues	4%	Directory Traversal	5%	Cryptographic Issues	9%	Cryptographic Issues	3%
Directory Traversal	3%	Error Handling	4%	Time and State	3%	Credentials Mgmt	3%
Encapsulation	3%	Buffer Overflow	4%	Error Handling	3%	API Abuse	2%
Time and State	1%	Potential Backdoor	3%	SQL Injection	3%	Time and State	1%
Insufficient Input Validation	1%	SQL Injection	3%	API Abuse	2%	Directory Traversal	1%
Buffer Overflow	1%	Time and State	2%	Buffer Overflow	1%	SQL Injection	1%

Veracode, State of Software Security Report: The Intractable Problem of Insecure Software, Apr 2011, pg 18

- You cannot secure something that was not built to be secure
  - Static and dynamic code analysis – credentialed and non-credentialed attacks
  - Web application firewalls
- Testing code before it is put into production
  - This can’t be last step before code into production – too late
  - Security testing has to be an integral part of development process



Training → Requirements → Design → Implementation → Verification → Release → Maintain

Application Security Controls Program

Developer Training

- Ensure developers have the knowledge required to develop secure applications.
- Secure Development Presentation.
- Hands-on workshop.

Application Risk Catalog

- Risk profiles to catalog applications.
- Applications prioritized by risk profile.

Security Functionality

- Baseline of security functionality requirements.
- Evolves over time as new application threats are identified.

Code Testing

- Developers include security objectives when performing peer review.
- Automated static code analysis using Fortify.

QA Testing

- Automated dynamic analysis using web application vulnerability scanner.

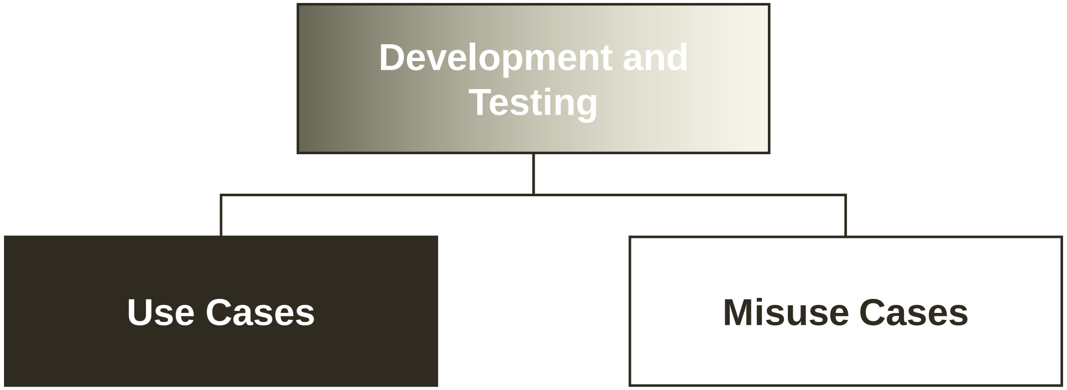
Penetration Testing

- Security testing of most critical applications.
- Performed before release into production.

Automated Testing

- Recurring authenticated testing using automated dynamic security testing tool.

- How does application security fit into the development lifecycle?
  - Functional testing is ensuring that all application functions perform as expected during normal user interaction.
  - Security testing is ensuring that all application functions perform as expected during *abnormal* user interaction.



Securing the Mobile Threat Space

- How complex is the mobile application security space today?
- Looking at this issue from non-applications perspective
  - Physical security – high likelihood of being lost, stolen or co-opted for some other use
  - Data stored on device is more valuable than device itself
- Malware
- Phishing
- Any device driver that has not been secured could be a weakness introduced into architecture of underlying OS
- Application and data isolation – prevent unwanted access to data

- Turn on Transport Layer Security (TLS) or Secure Sockets Layer (SSL)
- Follow secure programming practices
  - Secure coding guidelines (OWASP)
  - Security frameworks
- Validate input
- Leverage the permissions model of underlying OS
  - Permissions models on iPhone and Android generally isolate one app from another
- Store sensitive information properly
  - iPhone and Android have the ability to store sensitive information in non-clear text
- Sign the application code

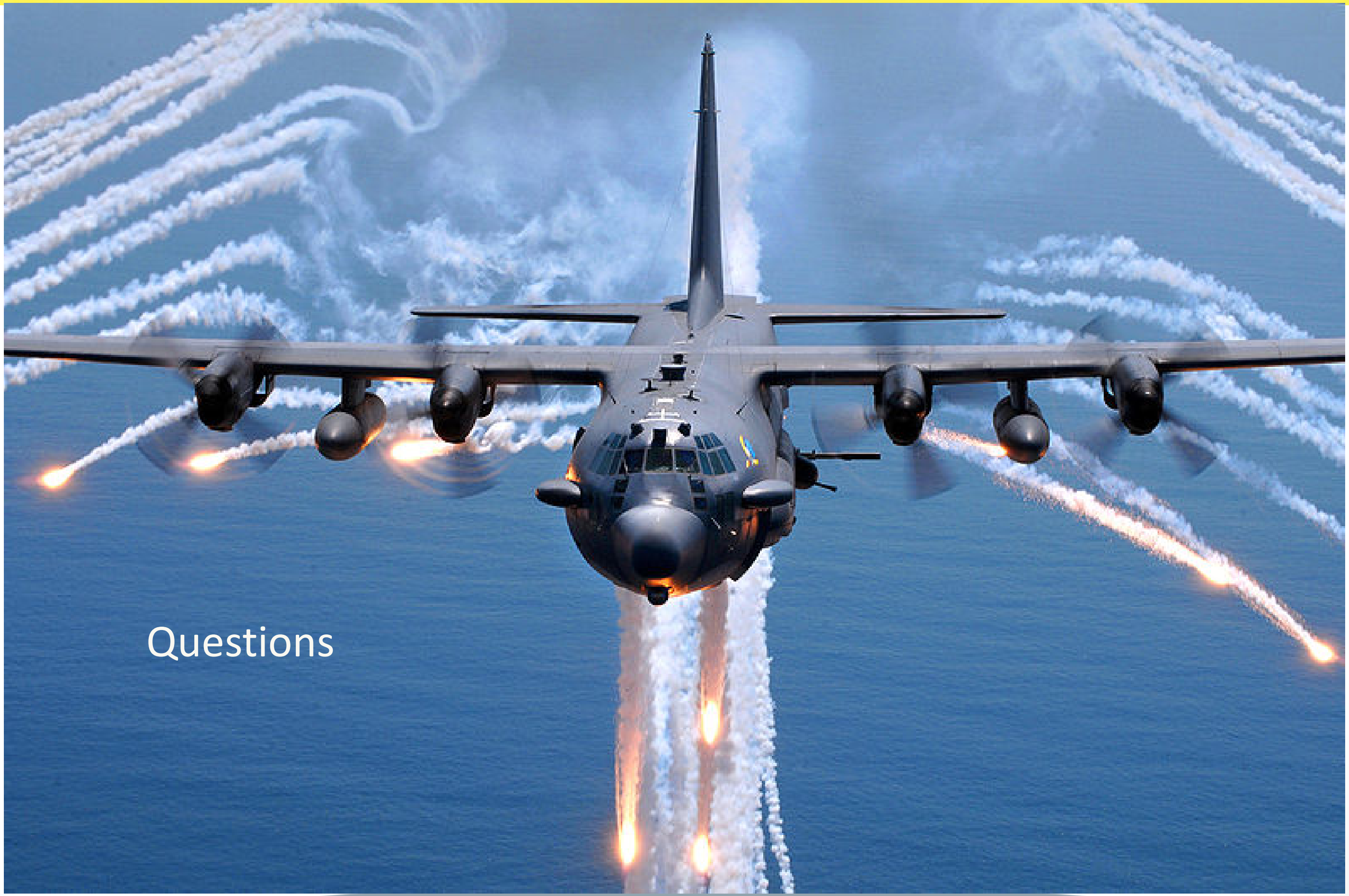
See Dwivedi, H, Clark, C., Thiel, D. Mobile Application Security. McGraw Hill pp 2-13

- Threat modeling for risk reduction
  - Thoroughly vet pros and cons of mobile architectures
    - Security models
    - Weaknesses
    - Securing administrative access
  - Pinpoint all input points in application design
    - Ensure that each of these is included in test plans for input validation
    - Map all data flows
      - Understand where data is stored
      - Understand who has access to data and why
      - Test access and authentication
- Ensure test plans are comprehensive

- Systematic testing
  - Static code analysis
  - Dynamic code analysis
  - Manual review
- Static code analysis can be problematic
  - Android is a Linux-based OS
  - Java-based coding
  - Tools like Fortify work exceptionally well
  - iPhone uses Objective-C coding
  - Most static code analyzers don’t cover this language
    - Flawfinder (www.dwheeler.com/flawfinder)
    - Clang Static Analyzer (clang-analyzer.llvm.org)

- Dynamic code analysis
  - Allows credentialed and non-credentialed testing
  - Very much like the attack might see application
- Manual review
  - Not all problems can be isolated using analyzers
  - Sometimes the best way to look at logic flow is to look at code and programs manually
  - Example: passing of parameters in the URLs
- Distributing the analysis process to development teams

- Conclusions
  - Moving into the mobile application space doesn't inherently mean that we had to change our software development techniques to secure the application
  - Techniques had to morph a bit to meet different threat models
  - Basic SDLC processes are much the same
  - Biggest challenge is in the handling of sensitive data flows when using mobile devices that in themselves have physical and logical security challenges
  - Need specialists who understand the hardware and software architectures of target devices
  - Remain entrepreneurial, but maintain a security focus



Questions

- Dwidvedi, H., Clark, C., Thiel, D. Mobile Application Security. McGraw Hill, 2010 ISBN 978-0-07-163356-7
- Cannings, R., Dwivedi, H., Lackey, Z. Hacking Web 2.0 Exposed McGraw Hill Osborne, 2008 ISBN 978-0-07-149461-8
- Veracode, State of Software Security Volume 3. Available online at: www.veracode.com/reports/index.html
- Verizon, 2011 Data Breach Investigations Report. Available online at: www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2011_en_xg.pdf