Agenda

• The evolving development environment
• Computer Emergency Response Team role in the system development life cycle and supply chain
• Useful strategies and technical solutions
Baseline Questions

• System development background?
  – Acquisition
  – Design
  – Development
  – Operations and maintenance
  – Disposal
Market Drivers

• What’s currently altering the development environment?
  – Regulation and standards
  – Evolution of tools
  – Training requirements and certifications
Regulation

- International Organization for Standardization 27001 and National Institute of Standards and Technology Special Publication 800-64v2
  - Build security into the system throughout development, deployment, operations and maintenance
  - Integrate security into requirements base and functional testing
  - Provides opportunity for incorporating Computer Emergency Response teams into the system development life cycle
Tools

- Current Computer Emergency Response Team tools:
  - Verdasys®
  - Encase®
  - Splunk®
  - Manager of managers (ArcSight®, Tivoli® Netcool®)
- Primarily used during Operations and Maintenance phase
- How can they be employed in other phases of the system development life cycle?

Verdasys is a registered trademark of Verdasys, Inc. in the U.S. and/or other countries.
Encase is a registered trademark of Guidance Software Inc. in the U.S. and/or other countries.
Splunk is a registered trademark of Splunk Inc. in the U.S. and/or other countries.
ArcSight is a registered trademark of ArcSight, Inc. in the U.S. and/or other countries.
Tivoli and Netcool are registered trademarks of International Business Machines Corporation in the U.S. and/or other countries.
Training/Certification

- **Certified Secure Software Lifecycle Professional (CSSLP®)**
  - First certification to explicitly address security in the system development life cycle (SDLC) process
- **SANS Institute Secure Coding Assessment**
  - Focus on specific coding languages rather than SDLC process
- **Department of Defense 8570**
  - Revised in May 2008, expect future updates with more detailed requirements
  - Currently doesn’t require CSSLP for system engineers or developers

CSSLP is a registered trademark of International Information Systems Security Certification Consortium, Inc. in the U.S. and/or other countries.
The System Development Life Cycle

Systems Development Life Cycle (SDLC) Life-Cycle Phases

1. Initiation
   - Begins when a sponsor identifies a need or an opportunity.
   - Concept Proposal is created.

2. System Concept Development
   - Defines the scope or boundary of the concept.

3. Planning
   - Develops a Project Management Plan and other planning documents.
   - Provides the basis for acquiring the resources needed to achieve a solution.

4. Requirements Analysis
   - Analyzes user needs and develops user requirements.
   - Creates a detailed Functional Requirements Document.

5. Design
   - Transforms detailed requirements into complete, detailed System Design Document.
   - Focuses on how to deliver the required functionality.

6. Development
   - Converts a design into a complete information system.
   - Includes acquiring and installing systems environment; creating and testing databases; preparing test case procedures; preparing test files; coding, compiling, refining programs; performing test readiness reviews and procurement activities.

7. Integration and Test
   - Demonstrates that the developed system conforms to requirements as specified in the Functional Requirements Document.
   - Conducted by Quality Assurance staff and users.
   - Produces Test Analysis Reports.

8. Implementation
   - Includes implementation preparation, implementation of the system into a production environment, and resolution of problems identified in the Integration and Test Phase.

9. Operations and Maintenance
   - Describes tasks to operate and maintain information systems in a production environment, includes Post-Implementation and In-Process Reviews.

10. Disposition
    - Describes end-of-system activities. Emphasis is given to proper preservation of data.

Source: Department of Justice SDLC Guidance
- Integration of security in all phases has lead to markedly more secure products
- Focus on eliminating vulnerabilities during design and coding, not through patches after release

System Development Life Cycle (SDLC)/Supply Chain Interdependencies

**Supplier**
- **SC: Plan** → **SDLC: Initiate**
- **SC: Source** → **SDLC: Acquire**
- **SC: Make** → **SDLC: Integrate**
- **SC: Deliver** → **SDLC: O&M**
- **SC: Return** → **SDLC: Dispose**

**Your Company**
- **SDLC:**
  - Initiate
  - Acquire
  - Integrate
  - O&M
  - Dispose
  - Operate & Maintain
  - Integrate/Implement/Assess
  - Deliver
  - Make
  - Plan
  - Acquire/Develop
  - Return

**Customer**
- **SC: Plan** → **SDLC: Initiate**
- **SC: Source** → **SDLC: Acquire**
- **SC: Make** → **SDLC: Integrate**
- **SC: Deliver** → **SDLC: O&M**
- **SC: Return** → **SDLC: Dispose**

O&M = Operations and Maintenance
Role of the Computer Emergency Response Team

• Current role
  – Not considered a valid stakeholder
  – Some basic, relevant baseline requirements introduced in initial phases

• Future role
  – Requirements, Design, Operations and Maintenance, Disposal phases
  – Active role in ongoing requirements development (joint application development/rapid application development)
Discussion Questions

- Anyone currently have a role in development of systems they monitor?
- Opportunities to influence the development of monitoring and forensic capabilities?
- Product Security Incident Response Team versus Computer Emergency Response Team functions? How are they related? Where do they overlap?
Models are needed to depict a managed architecture

...our experience across many IT programs has helped us evolve and develop a common SOA framework that is “technology-agnostic” while bearing forth distinct layers for consideration on any project.
• Inner “V’s” depict development/integration of each configuration item (CI)
• Outer “V” depicts overall systems engineering and integration at the program and portfolio level
• Requires innovation in incident response programs too!

M&S = modeling and simulation  C&A = Certification and Accreditation  DT = demonstration testing
OT = operational testing  LRIP = low-rate initial production  FRP = full-rate production
ADM = Architecture Development Method  SE&I = System Engineering and Integration
Reconciling Viewpoints

Requirements development
• Security requirements are functional requirements
• Security requirements of various government standards are bare minimums, yet are generally considered to be all that are necessary to produce a system with adequate/good security
• Secure system-of-system design/architecture requires moving beyond compliance with bare minimum of requirements

Reconciling viewpoints on security requirements
• Difference in opinion between developers, certifiers and the designated approving authority
• Early agreement necessary to avoid costly changes later in the process

Assignment of requirements to system components
• What’s satisfied locally by the platform/applications?
• Using external security services?
• Reuse of previously approved solutions
A Solution-Oriented Framework

• Usage model:
  – [Design], configure, generate, validate, [alert, index, search, retrieve], archive
  – An example: Splunk® and grep
  – Another example: Verdasys® and Encase® network

Splunk is a registered trademark of Splunk Inc. in the U.S. and/or other countries.
Verdasys is a registered trademark of Verdasys, Inc. in the U.S. and/or other countries.
Encase is a registered trademark of Guidance Software Inc. in the U.S. and/or other countries.
### The Building Security In Maturity Model

#### Ten Core Activities Everybody Does

<table>
<thead>
<tr>
<th>Objective</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>build support throughout organization</td>
<td>create evangelism role/internal marketing</td>
</tr>
<tr>
<td>meet regulatory needs or customer demand with a unified approach</td>
<td>create policy</td>
</tr>
<tr>
<td>promote culture of security throughout the organization</td>
<td>provide awareness training</td>
</tr>
<tr>
<td>see yourself in the problem</td>
<td>create/use material specific to company history</td>
</tr>
<tr>
<td>create proactive security guidance around security features</td>
<td>build/publish security features (authentication, role management, key management, audit/log, crypto, protocols)</td>
</tr>
<tr>
<td>build internal capability on security architecture</td>
<td>have SSG lead review efforts</td>
</tr>
<tr>
<td>drive efficiency/consistency with automation</td>
<td>use automated tools along with manual review</td>
</tr>
<tr>
<td>use encapsulated attacker perspective</td>
<td>integrate black box security tools into the QA process (including protocol fuzzing)</td>
</tr>
<tr>
<td>demonstrate that your organization’s code needs help too</td>
<td>use external pen testers to find problems</td>
</tr>
<tr>
<td>provide a solid host/network foundation for software</td>
<td>ensure host/network security basics in place</td>
</tr>
</tbody>
</table>

#### Three Core Activities that Most Organizations Do

<table>
<thead>
<tr>
<th>Objective</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>understand the organization’s history</td>
<td>collect and publish attack stories</td>
</tr>
<tr>
<td>meet demand for security features</td>
<td>create security standards</td>
</tr>
<tr>
<td>use ops data to change dev behavior</td>
<td>identify software bugs found in ops monitoring and feed back to dev</td>
</tr>
</tbody>
</table>

- Source: www.bsi-mm.com
Concluding Discussion

• Possible solutions:
  – Virtualization
  – Service-oriented architecture
  – Cloud