CLASP, SDL and Touchpoints compared

Bart De Win
DistriNet, Dept. of Computer science
K.U.Leuven
bart.dewin@cs.kuleuven.be

OWASP Day
Belgium

6 Sep 2007

The OWASP Foundation
http://www.owasp.org/

Agenda

- Introduction
- Phase-wise comparison
- Discussion

Introduction

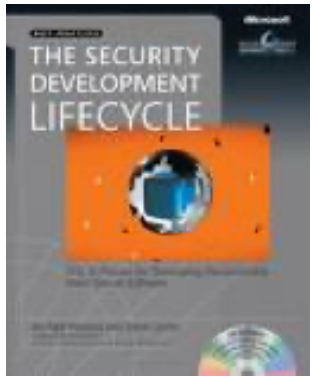
- Processes for secure software development have become available
  - CLASP, SDL, Touchpoints, Correctness by Construction, …
  - Shown to considerably improve the security level of software in practice
- It is not so easy to pick the most suited one
  - How do they compare ?
  - What are their strong and weaker points ?
  - Can they be combined ?
  - Is there room for improvement ?
- Highlights of a *theoretical* comparison of three candidates: CLASP, SDL and Touchpoints
  - Difficult and time-consuming job
  - Activity-wise analysis
- Joint work with Riccardo Scandariato, Koen Buyens, Johan Grégoire and Wouter Joosen

Common Lightweight Application Security Process (CLASP)

- Originally defined by Secure Software, later donated to OWASP
- Key players: Pravir Chandra (project lead), John Viega
- Most recent version: 1.2, version 2007 is announced
- Core is a set of 24 activities

- General characteristics
  - Security at center stage
  - Loose structure
  - Role-based
  - Rich in resources

Secure Development Lifecycle (SDL)



- Result of Microsoft's commitment to trustworty computing (from 2002 onwards)
- Book written by Michael Howard and Steve Lipner (2006)
- The core process is organized in 12 stages

- General characteristics
  - Security as a supporting quality
  - Well-defined process
  - Good guidance
  - Management perspective

Touchpoints (TP)

- Based on the book by Gary McGraw (2007)
- Set of best practices, grouped into 7 touchpoints.

- General characteristics
  - Risk management
  - Black-hat versus white-hat
  - Prioritization of touchpoints (quick wins)
  - Resource and knowledge management

How to compare in more detail ?

- Problem:
  - Different setup
  - Different activities
- Our approach
  - Identify activities
  - Optimize hierarchy
  - Link similar activities
  - Organize into phases (5+1)
  - Result: **activity matrix**
- Used as a vehicle for evaluation and comparison

Project Inception Phase			
Activity	*SDL*	*CLASP*	*Touch points*
2.1. Build security			
2.1.1. Build security team	✓	✘	✓
2.1.2. Assign security advisor	✓	✓	✘
2.1.3. Institute accountability for security issues	✘	✓	✘
2.2. Determine whether the application is covered by methodology	✓	✘	✘
2.3. Initial security			
2.3.1. Provide tools to track security issues	✓	✘	✘
2.3.2. Determine the bug bar	✓	✘	✘
2.4. Monitor security metrics			✓
2.4.1. Identify metrics to collect & identify how they will be used	✘	✓	?
2.4.2. Institute data collection and reporting strategy	✘	✓	?
2.4.3. Periodically collect and evaluate metrics *(ongoing during entire lifecycle)*	✘	✓	?
2.5. Institute rewards	✓	✓	✘
2.6. Identify global security policy			
2.6.1 Identify global project security policy, if necessary	✘	✓	✘
2.6.2. Determine suitability of global requirements to project	✘	✓	✘
2.7. Build an improvement program	✘	✘	✓
2.8. Execute continuous improvement program	✘	✘	✓

Education and awareness

- Common baseline
  - Basic and specific education
  - Increase the awareness of the problem and the specific environment
- Differentiators
  - For CLASP, education is basis for accountability
  - In SDL, attention is given to track attendance and measure effectiveness of courses
  - Briefly mentioned in Touchpoints

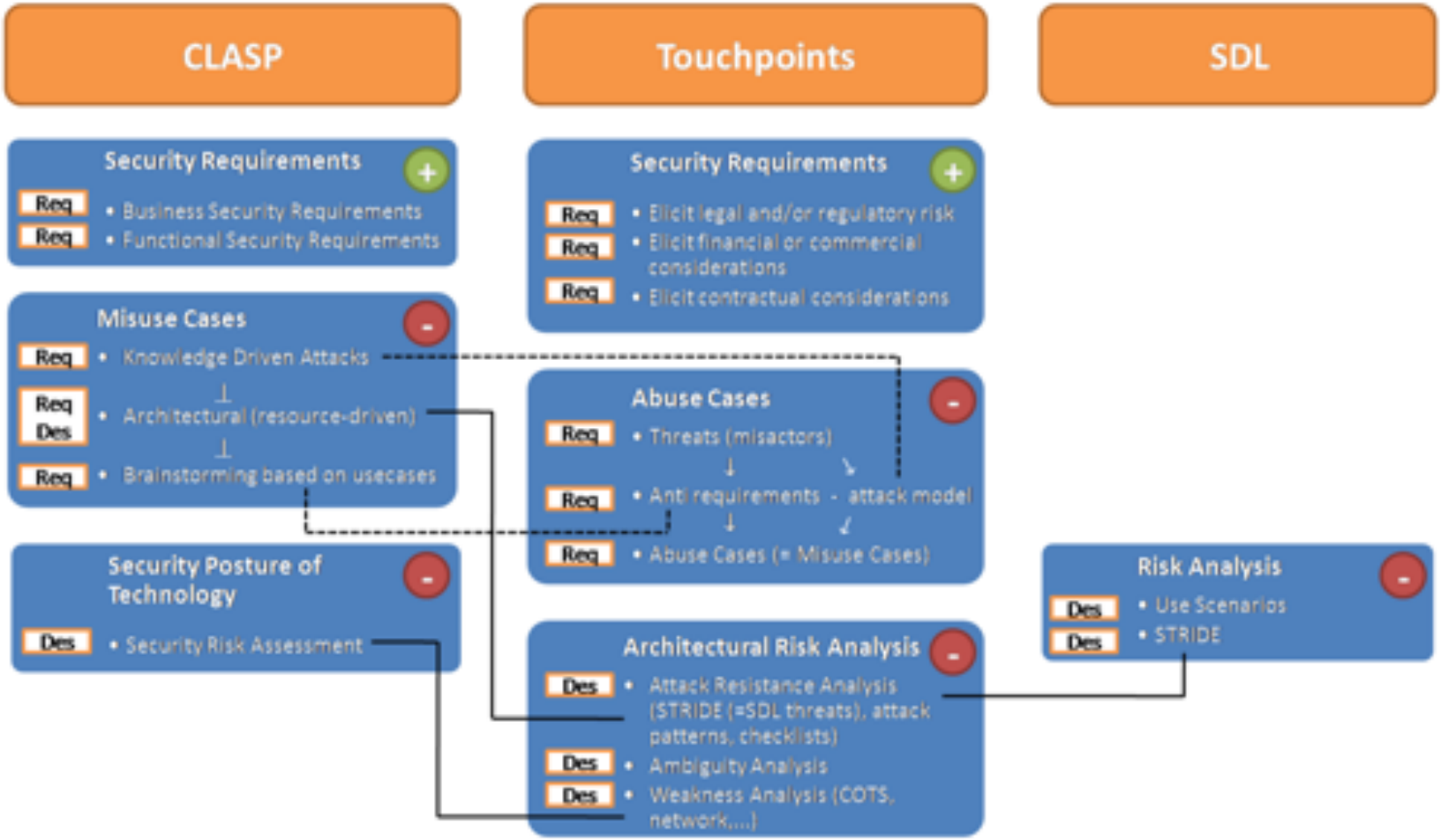
Project inception

- Common baseline
  - Installation of the security team
  - Identification of security metrics
  - Logistics and tools
- Differentiators
  - Extent of the security team
  - SDL explicitly sets the "bug bar"
  - CLASP identifies the global organizational policy (an important source for requirements)
- Discussion
  - CLASP is the most thorough in discussing metrics, but still much room for improvement
  - Upfront determination of security goals ?

Analysis

- Common baseline
  - Threat modeling and requirements specification
- Differentiators
  - See figure
- Discussion
  - Combination of CLASP and TP might benefit analysis-level threat modeling
    - CLASP: attack-driven, resource-driven, UC-driven
    - TP: actor * anti-requirement * attack model => MUC
  - Threat modeling for conceptual resources (assets) ?
  - How to deal with the *threat explosion problem*

Analysis (ctd.)



Design

- Common baseline
  - Attack surface scrubbing (not in TP)
  - Product risk assessment
  - Architectural threat analysis
- Differentiators
  - Only CLASP focuses on constructive design
    - Annotate class design, security principles in design
  - Microsoft's STRIDE provides thorough and systematic threat modeling
- Discussion
  - Little support for architectural design

Implementation and Testing

- Common baseline
  - Secure coding guidelines (not in TP)
  - Security analysis & code review
  - Security testing
  - Addressing security issues (not in TP)
- Differentiators
  - CLASP: includes implementation activities
  - SDL: creation of tools for configuration and audit
  - Security testing: black-hat versus white-hat, unit versus system, black-box versus white-box, …
- Discussion
  - Test generation and automation
  - Difficulty of determining test coverage (esp. black-hat)

Deployment and support

- Common baseline
  - Documentation and security guides
  - Response planning and execution
- Differentiators
  - Code sign-off (SDL) & code signing (CLASP)
  - SDL: elaborate response planning and execution
- Discussion
  - Focus on support rather than deployment

Synthesis and discussion

- The three processes are similar and they can be mapped to each other
  - CLASP has the widest scope. When fully (and properly) applied, it is probably the heaviest candidate (despite being named lightweight)
  - SDL is more focused and, hence, it often provides the most concrete activities
  - Touchpoints is well suited from an audit perspective. It has interesting ideas, but is often too descriptive.
- The main goal of a process should be to increase systematicity, predictability and coverage.
- *Advise:* start with the one that suits your goal best and augment where necessary with elements from the others.

Possible improvements

- Activities:
  - Method: not *what* to do, but *how* to do it
  - Systematic (no 100% security, but know what you're doing)
  - Description: input – method – output + resources
  - Good mix of construction – verification - management
- *Integration* of activities
  - Output Act.1 -> input Act.2 for all constructive activities
- Security metrics to measure progress
  - Activity-wise and process-wise
- Integrated support for security principles
- Security patterns are relevant at all levels
  - Vulnerabilities, requirements, design, testing, …
- Further experience !

Questions ?

Requirements Elicitation

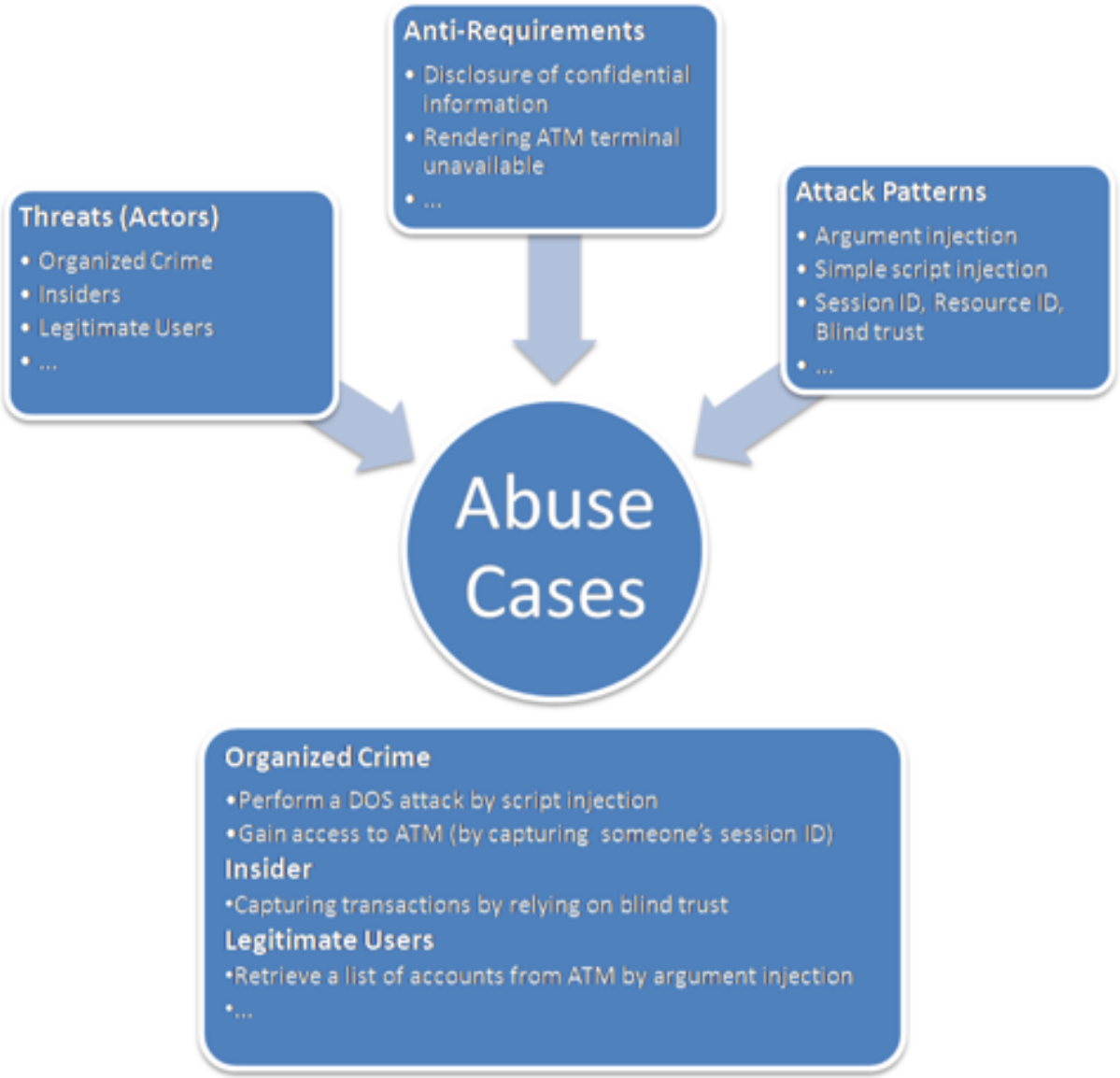
Class	Resource	Requirement
User-confidential	Customer Information	1. User-confidential data is only created by the banking company, the banking system or the ATM terminal.
Banking System Processes	Banking Service	2. Start/Stop/Restart actions are only executed by the Banking System Administrator.
...	...	...

Coverage Verification

Class	Resource	Capability	Covered Requirement
User-confidential	Customer Information	Add(create)	1
User-confidential	Transaction Information	Create	1
User-confidential	Transaction Information	Set Ownership	NO
User-confidential	Transaction Information	Read Meta-attributes	NO
Banking System Processes	Banking Service	Start/Stop/Restart	2
...	...	...	...

Elicit Special Requirements

Resource	Capability	Requirement
Transaction Log File	Set Ownership	The ownership of the transaction log file is only set by the security administrator.
Transaction Log File	Read Meta-attributes (last time database modified)	The meta-attributes of the transaction log file are only read by the bank auditor.
...	...	...


Anti-Requirements
• Disclosure of confidential information
• Rendering ATM terminal unavailable
• ...
Threats (Actors)
• Organized Crime
• Insiders
• Legitimate Users
• ...
Attack Patterns
• Argument injection
• Simple script injection
• Session ID, Resource ID, Blind trust
• ...
Abuse Cases
Organized Crime
•Perform a DOS attack by script injection
•Gain access to ATM (by capturing someone's session ID)
Insider
•Capturing transactions by relying on blind trust
Legitimate Users
•Retrieve a list of accounts from ATM by argument injection
•...