OWASP Cornucopia

Oana Cornea

9th of March 2016
About me

• Oana Cornea
• Leader of the OWASP Bucharest Chapter
• Penetration tester at Dell Secureworks
Agenda

• Introduction
• Agile development and security
• Cornucopia card game
• Playing the game
Application security

- Architecture review
- Penetration testing
- Source code review
Threat modeling

• It can be hard for normal engineers – it requires the “security mindset”
• Software engineers know their code and how it changes
Agile development and security

• security controls rely on a complete picture of the application
• security testing often relies on components outside the current sprint
  • Security tests can be cumulative. One user story can directly cause a security vulnerability in another.
  • While functional testing tends to be defined, security testing does not. There will be a trade-off of accuracy versus completeness in order to fit within a given sprint
OWASP Cornucopia project

- Created by Colin Watson
- Inspired by Microsoft Elevation of Privilege (EoP) Card Game
  - More web application relevant
  - More coverage of web security requirements
  - Less vendor specific and more webapp/OWASP specific
  - More information rich
  - More individual
# SAFECODE - Practical Security Stories and Security Tasks for Agile Development Environments

<table>
<thead>
<tr>
<th>No.</th>
<th>Security-focused story</th>
<th>Backlog task(s)</th>
<th>SAFECODE Fundamental Practice(s)</th>
<th>CWE-ID</th>
</tr>
</thead>
</table>
| 18  | As a(n) architect/developer, I want to ensure **AND** as QA, I want to verify that cross-site request forgery attacks are prevented. | **D** Use one of the many available libraries and frameworks that takes CSRF into account.  
**D** Defend against cross-site scripting (see Story 17).  
**A/V/D** Add business logic and workflow steps to critical processes in the system, and make them out-of-band; send an email in case of password change, send a text message when changing a critical value.  
**D/T** Log critical operations and the details of their initiation and arguments.  
**A/V/D** Do not use HTTP GET for any method that affects a change in system state. |  
- Use Anti-Cross Site Scripting (XSS) Libraries  
- Validate Input and Output to Mitigate Common Vulnerabilities  
- Use Logging and Tracing | CWE-352 |
| 19  | As a(n) architect/developer, I want to ensure **AND** as QA, I want to verify proper neutralization of Special Elements used in an OS Command ('OS Command Injection') | **D** Consider all input as malicious and filter according to the context.  
**D** Check all arguments to functions like exec() or system() for the expected format before executing.  
**D** Limit the use of external processes; prefer library calls.  
**D** Use static code analysis tools.  
**D** Consider the use of command shells (system()) as opposed to directly calling an executable (exec()), and its implications in command line arguments, like shell expansion.  
**A/V/D** Reduce the attack surface by adopting the backlog items of “Execution with Unnecessary Privileges.” |  
- Validate Input and Output to Mitigate Common Vulnerabilities  
- Use Static Analysis Tools  
- Use Least Privilege | CWE-78 |
Authentication and Password Management:

- Require authentication for all pages and resources, except those specifically intended to be public
- All authentication controls must be enforced on a trusted system (e.g., The server)
- Establish and utilize standard, tested, authentication services whenever possible
- Use a centralized implementation for all authentication controls, including libraries that call external authentication services
- Segregate authentication logic from the resource being requested and use redirection to and from the centralized authentication control
- All authentication controls should fail securely
- All administrative and account management functions must be at least as secure as the primary authentication mechanism
- If your application manages a credential store, it should ensure that only cryptographically strong one-way salted hashes of passwords are stored and that the table/file that stores the passwords and keys is write-able only by the application. (Do not use the MD5 algorithm if it can be avoided)
- Password hashing must be implemented on a trusted system (e.g., The server).
- Validate the authentication data only on completion of all data input, especially for sequential authentication implementations
- Authentication failure responses should not indicate which part of the authentication data was incorrect. For example, instead of "Invalid username" or "Invalid password," just use "Invalid"
The cards

Suites:
• Session management
• Cryptography
• Data validation and encoding
• Authentication
• Authorization
• Cornucopia

13 cards per suit, 2 Jokers
Play a round, highest value wins
How to play (1)

• Preparations
  • Identify an application or application process to review; this might be a concept, design or an actual implementation
  • Create a data flow diagram
  • Identify and invite a group of 3-6 architects, developers, testers and other business stakeholders together and sit around a table (try to include someone fairly familiar with application security)
How to play (2)

• Remove the Jokers and a few low-score (2, 3, 4) cards from Cornucopia suit to ensure each player will have the same number of cards
• Shuffle the deck and deal all the cards
• To begin, choose a player randomly who will play the first card - they can play any card from their hand except from the trump suit – Cornucopia
• To play a card, each player must read it out aloud, and explain how (or not) the threat could apply (the player gets a point for attacks that work, and the group thinks it is an actionable bug)
Consider supplementing standard session management with:

• Per-session strong random tokens or parameters.
• Per-request, as opposed to per-session, strong random tokens or parameters.
Other session management card example

- Casey can utilize Adam's session after he has finished, because there is no log out function, or he cannot easily log out, or log out does not properly terminate the session.
Cryptography

The key concept for this card is protection of master cryptographic secrets, within the application and more widely in management processes.
Data validation and encoding

Depending of the target of the attack, the results of this type of input varies widely:

• Information disclosure (error logs, system responses, etc.).
• Operations tampering.
• Denial of Service.
• Spoofing.
• Code execution.
Authentication

- The degree of identity assurance may not be the same for all web application functions. Or the authentication function may be available in a weaker manner in some other mode or channel, thus compromising the web application.
Players can discuss any type of Authentication (AT) attack they think might be possible against the assessment target. It does not matter if the attack relates to another AT card, but if possible try to identify an attack that is fairly unique to the application/functionality/users.
Authorization

Do not make assumptions about the order or previous actions of a user. Re-perform authorization checks at each and every step.
Cornucopia

Application-layer denial of service and other activities that adversely affect the application's users.
How to play (3)

- Someone record the card on the score sheet
- Play clockwise, each person must play a card in the same way; if you have any card of the matching lead suit you must play one of those, otherwise they can play a card from any other suit. Only a higher card of the same suit, or the highest card in the trump suit Cornucopia, wins the hand.
- The person who wins the round, leads the next round (i.e. they play first), and thus defines the next lead suit
- Repeat until all the cards are played
Scorecard
Does Cornucopia matter?

5.10 Resources

Organizations should familiarize themselves with industry-accepted best practices and guidelines for securing e-commerce environments. There are a wide range of resources at varying levels of depth and technical detail. Examples of resources that may provide guidance and technical security data breach reports include:

5.10.1 Information Security Resources

Information security resources provide an in-depth review of topics important to e-commerce, such as secure application development, analysis of attack patterns, and alerts on emerging threats:

- **Open Web Application Security Project (OWASP)** ([www.owasp.org](http://www.owasp.org)). OWASP is a global not-for-profit charitable organization focused on improving the security of web applications. OWASP’s mission is to make application security visible so that individuals and organizations worldwide can make informed decisions about the true risks surrounding application development and security. OWASP provides a number of resources for training and application security awareness, including: podcasts, eBooks, online publications, news feeds, blogs, videos, conferences, and in-person classroom training.

  The **OWASP Development Guide** is a comprehensive reference manual for designing, developing, and deploying secure web services and applications. Individual guides include *Handling E-Commerce Payments*, *Security of Payment Cards (Credit/Debit) In E-commerce Application*, and *Cornucopia E-commerce Web Site Edition*.

- **The SysAdmin, Audit, Network, and Security (SANS) Institute** ([www.sans.org](http://www.sans.org)). The SANS Institute is a privately-held U.S. company providing information security resources, training, and technology.
Reference files

- OWASP SCP requirements
- OWASP ASVS verification IDs
- OWASP AppSensor attack detection point IDs
- CAPEC IDs
- SAFECODE security-focused story IDs
OWASP Resources

• Project page: https://www.owasp.org/index.php/OWASP_Cornucopia

• How to play: https://youtu.be/i5Y0akWj31k
Questions?

Thank you!