



# OWASP Cornucopia

## Ecommerce Website Edition

*OWASP Cornucopia - Ecommerce Website Edition helps developers identify security requirements from the OWASP Secure Coding Practices - Quick Reference Guide*

- Colin Watson
- Watson Hall Ltd  
London, United Kingdom
- <https://www.watsonhall.com>

# SAFECode - Practical Security Stories and Security Tasks for Agile Development Environments



## Practical Security Stories and Security Tasks for Agile Development Environments

JULY 17, 2012

### Table of Contents

Problem Statement and Target Audience	2
Overview	2
Assumptions	3
Section 1) Agile Development Methodologies and Security	3
How to Choose the Security-focused Stories and Security Tasks?	3
Story and Task Prioritization Using "Security Debt"	4
Residual Risk Acceptance	4
Section 2a) Security-focused Stories and Associated Security Tasks	5
Section 2b) Operational Security Tasks	29
Section 3) Tasks Requiring the Help of Security Experts	31
Appendix A) Residual Risk Acceptance	32
Glossary	33
References	33
About SAFECode	34

No.	Security-focused story	Backlog task(s)	SAFECode Fundamental Practice(s)	CWE-ID
18	As a(n) architect/developer, I want to ensure <b>AND</b> as QA, I want to verify that cross-site request forgery attacks are prevented	<p>[D] Use one of the many available libraries and frameworks that takes CSRF into account.</p> <p>[D] Defend against cross-site scripting (see Story 17).</p> <p>[A/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.</p> <p>[D/T] Log critical operations and the details of their initiation and arguments.</p> <p>[A/D] Do not use HTTP GET for any method that effects a change in system state.</p>	<ul style="list-style-type: none"> <li>• Use Anti-Cross Site Scripting (XSS) Libraries</li> <li>• Validate Input and Output to Mitigate Common Vulnerabilities</li> <li>• Use Logging and Tracing</li> </ul>	CWE-352
19	As a(n) architect/developer, I want to ensure <b>AND</b> as QA, I want to verify proper neutralization of Special Elements used in an OS Command ('OS Command Injection')	<p>[D] Consider all input as malicious and filter according to the context.</p> <p>[D] Check all arguments to functions like <code>exec()</code> or <code>system()</code> for the expected format before executing.</p> <p>[D] Limit the use of external processes; prefer library calls.</p> <p>[D] Use static code analysis tools.</p> <p>[D] Consider the use of command shells [<code>system()</code>] as opposed to directly calling an executable [<code>exec()</code>] and its implications in command line arguments, like shell expansion.</p> <p>[A/D] Reduce the attack surface by adopting the backlog items of "Execution with Unnecessary Privileges."</p>	<ul style="list-style-type: none"> <li>• Validate Input and Output to Mitigate Common Vulnerabilities</li> <li>• Use Static Analysis Tools</li> <li>• Use Least Privilege</li> </ul>	CWE-78

# OWASP Secure Coding Practices – Quick Reference Guide



## OWASP Secure Coding Practices – Quick Reference Guide

### Copyright and License

Copyright © 2010 The OWASP Foundation.

This document is released under the Creative Commons Attribution ShareAlike 3.0 license. For any reuse or distribution, you must make clear to others the license terms of this work.  
<http://creativecommons.org/licenses/by-sa/3.0/>

Version 2.0

November 2010

November 2010

### 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



# Microsoft Elevation of Privilege (EoP) Card Game

## Elevation of Privilege (EoP) Card Game



Elevation of Privilege (EoP) is the easy way to get started [threat modeling](#), which is a core component of the [design phase](#) in the Microsoft Security Development Lifecycle (SDL).

The EoP card game helps clarify the details of threat modeling and examines possible threats to software and computer systems.

The EoP game focuses on the following threats:

- Spoofing
- Tampering
- Repudiation
- Information Disclosure
- Denial of Service
- Elevation of Privilege



EoP uses a simple point system that allows you to challenge other developers and become your opponent's biggest threat.



# Downloads for EoP



## Elevation of Privilege (EoP) Threat Modeling Card Game



### Quick links

[Overview](#)[System requirements](#)[Instructions](#)

### Looking for support?

[Visit the Microsoft Support site now >](#)

Elevation of Privilege (EoP) is the easy way to get started threat modeling. It is a card game that developers, architects or security experts can play.

### Quick details

Version:	1	Date published:	2/7/2013
Language:	English		

### Files in this download

The links in this section correspond to files available for this download. Download the files appropriate for you.

File name	Size	
EoP_Card Game Images.pdf	6.0 MB	<a href="#">DOWNLOAD</a>
EoP_Cards_Box_Native_files.zip	85.9 MB	<a href="#">DOWNLOAD</a>
EoP_Instructions.pdf	565 KB	<a href="#">DOWNLOAD</a>
EoP_Score Card.pdf	357 KB	<a href="#">DOWNLOAD</a>
EoP_Whitepaper.pdf	271 KB	<a href="#">DOWNLOAD</a>

### Overview

Elevation of Privilege (EoP) is the easy way to get started threat modeling. It is designed to make threat modeling easy and accessible for developers and architects. Threat modeling is a core security practice during the design phase of the Microsoft Security Development Lifecycle (SDL). The EoP card game helps examine possible threats to software and computer system. This game is licensed under the Creative Commons Attribution 3.0 United States License. Native files of the game are made available to allow editing, localization, and printing of the game. To view the full content of this license, visit <http://creativecommons.org/licenses/by/3.0/us/>

[↑ Top of page](#)

### System requirements

**Supported operating systems:** Windows 7, Windows Server 2003, Windows Server 2008, Windows Vista, Windows XP

## More web application relevant



### EoP examples

- An attacker could squat on the random port or socket that the server normally uses
- An attacker can confuse a client because there are too many ways to identify a server
- An attacker can make [your authentication system|client|server] unusable or unavailable [without ever authenticating] [but the problem goes away when the attacker stops|and the problem persists after the attacker goes away] (10 cards)
- An attacker can provide a pointer across a trust boundary, rather than data which can be validated

### Cornucopia examples

- Gary can take over a user's session because there is a long or no inactivity timeout, or a long or no overall session time limit, or the same session can be used from more than one device/location
- Marce can forge requests because per-session, or per-request for more critical actions, strong random tokens or similar are not being used for actions that change state
- Eduardo can access data he does not have permission to, even though he has permission to the form/page/URL/entry point

# More coverage of web security requirements



## EoP suits = STRIDE

- **Spoofing**  
Impersonating something or someone else
- **Tampering**  
Modifying data or code
- **Repudiation**  
Claiming to have not performed an action
- **Information Disclosure**  
Exposing information to someone not authorized to see it
- **Denial of Service**  
Deny or degrade service to users Elevation of Privilege Gain capabilities without proper authorization

## Cornucopia suits

-  **Data validation and encoding**  
Input and output data validation and escaping
-  **Authentication**  
Verification of identity claims and related processes
-  **Session management**  
Maintenance of user state
-  **Authorization**  
User/role permission controls
-  **Cryptography**  
Hashing, digital signatures, encryption and random number generation processes and their usage including key management
-  **Cornucopia (everything else)**  
Everything else including information leakage, data loss, configuration management, denial of service



# Less colourful and less pictorial



## EoP playing cards



## Cornucopia playing cards





## Less vendor specific and **more** webapp/OWASP specific ✓

### EoP examples

- An attacker could take advantage of .NET permissions you ask for, but don't use
- An attacker can alter information in a data store because it has weak ACLs or includes a group which is equivalent to everyone ("all Live ID holders")

### Cornucopia examples

- Bob can influence, alter or affect the application so that it no longer complies with legal, regulatory, contractual or other organizational mandates
- You have invented a new attack of any type

Read more about application security in OWASP's free Guides on Requirements, Development, Code Review and Testing, the Cheat Sheet series, and the Open Software Assurance Maturity Model

- You have invented a new attack against Authorization

Read more about this topic in OWASP's Development and Testing Guides

## More information rich



### EoP

- Suit name (e.g. Denial of Service)
- Attack description
- Ranking (card number)

### Cornucopia

- Suit name (e.g. Authentication)
- Attack description
- Ranking (card number)
- Cross-referencing  
Security requirements, security verification checks, attack detection points, attack patterns and Agile user stories

OWASP SCP  
10, 32, 93, 94, 189

OWASP ASVS  
4.1, 4.2, 4.3, 4.4, 4.6, 4.12

OWASP AppSensor  
ACE3

CAPEC  
25, 39, 74, 162, 166, 207

SAFECode  
8, 10, 11, 12

OWASP Cornucopia Ecommerce Website Edition v1.01

## More individual



### EoP

- An attacker could steal credentials stored on the server and reuse them (for example, a key is stored in a world readable file)
- An attacker can manipulate data because there's no integrity protection for data on the network
- An attacker can provide or control state information
- An attacker can say "I didn't do that," and you'd have no way to prove them wrong

### Cornucopia

- Shamun can bypass input validation or output validation checks because validation failures are not rejected or sanitized
- Kyun can access data because it has been obfuscated rather than using an approved cryptographic function
- Keith can perform an action and it is not possible to attribute it to him



## More individual



### EoP

- An attacker could steal credentials stored on the server and reuse them (for example, a key is stored in a world readable file)
- An attacker can manipulate data because there's no integrity protection for data on the network
- An attacker can provide or control state information
- An attacker can say "I didn't do that," and you'd have no way to prove them wrong

### Cornucopia

- Shamun can bypass input validation or output validation checks because validation failures are not rejected or sanitized
- Kyun can access data because it has been obfuscated rather than using an approved cryptographic function
- Keith can perform an action and it is not possible to attribute it to him

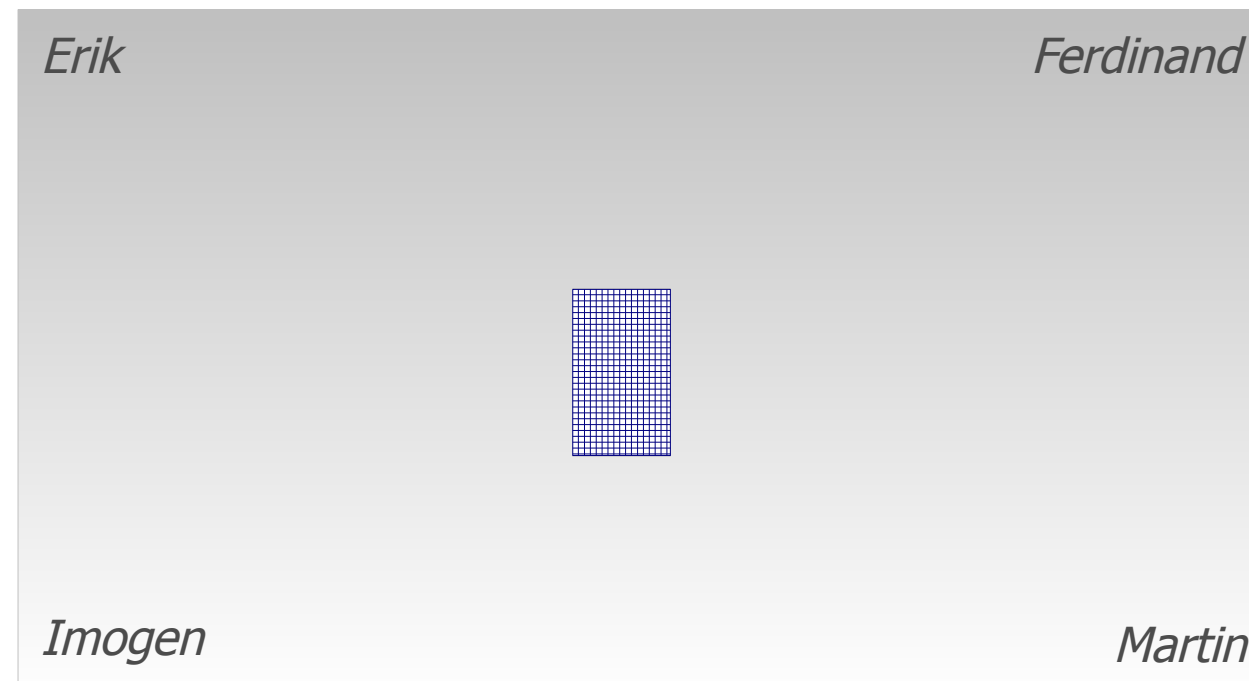
# What's in a name?



The “names” can represent

- External or internal people
- Aliases for computer system components
  - The application itself
  - Other applications
  - Services
  - Operating systems
  - Infrastructure
- Jim can undertake malicious, non-normal, actions without real-time detection and response by the application
- Erik, Ferdinand and Martin are not guilty of doing anything malicious

## Deal the deck of cards



### *Outcomes:*

- *Players have the same number of cards each*
- *Randomly select one player to lead the play for the first round e.g. Ferdinand*



# Identifying requirements with each card played

- Suit and value
- Attack description
- Cross-referencing

AUTHENTICATION

7

Cecilia can use brute force and dictionary attacks against one or many accounts without limit, or these attacks are simplified due to insufficient complexity, length, expiration and re-use requirements for passwords

---

OWASP SCP  
33, 38, 39, 41, 50, 53

---

OWASP ASVS  
2.3

---

OWASP AppSensor  
AE2, AE3

---

CAPEC  
2, 16

---

SAFECode  
27

---

OWASP Cornucopia Ecommerce Website Edition v1.01

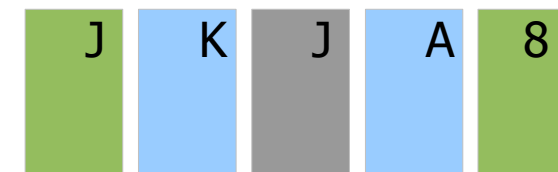
- Is this a viable attack for the function/system under consideration?
- Document the attack
- Subsequently use the cross-references to help create security requirements:
  - User stories
  - Unit tests
  - Configurations
  - etc

# Let play commence – First round

0 Requirements  
0 Rounds



0 Requirements  
0 Rounds



Schedule of requirements



Erik

Ferdinand

Imogen

Martin

- Assume every player Except "Imogen" identified a security requirement, thus 1 point each for the others
- "Ferdinand" won the round with the King so he gets an additional 1 point, and leads the play for the next round



0 Requirements  
0 Rounds



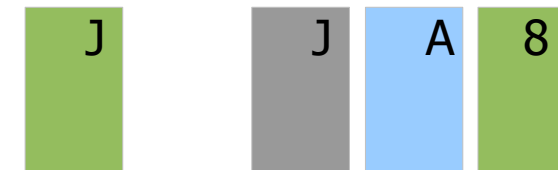
0 Requirements  
0 Rounds

# Second round

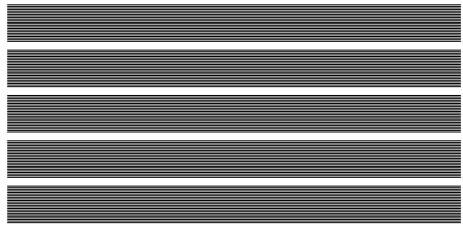
1 Requirements  
0 Rounds



2 Requirements  
1 Rounds



Schedule of requirements



Erik

Ferdinand

Imogen

Martin

- Only "Ferdinand" and "Imogen" identified new requirements and they each receive 1 point
- "Martin" won the round with the Ace so he gets 1 point for that, and leads the play for the next round



0 Requirements  
0 Rounds



1 Requirements  
0 Rounds

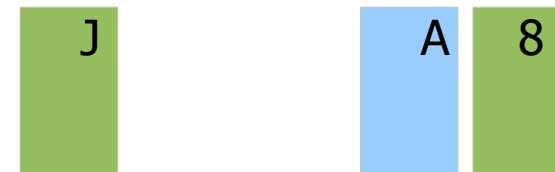


# Third round

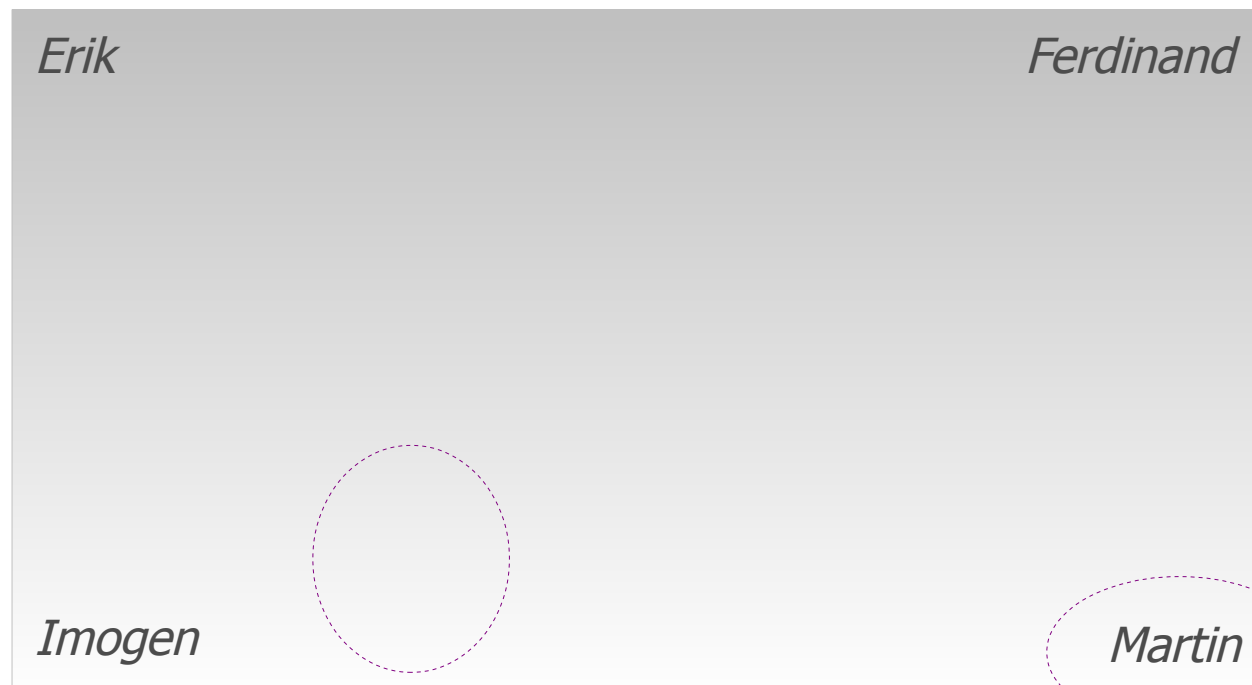
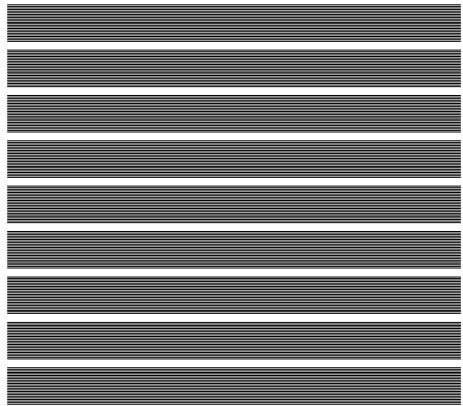
2 Requirements  
0 Rounds



2 Requirements  
1 Rounds



Schedule of requirements



- Everyone identified new requirements and they each receive 1 point
- "Imogen" won the round with the Queen so she gets 1 point for that, and leads the play for the next round



2 Requirements  
0 Rounds



2 Requirements  
1 Rounds

# Fourth round

*2 Requirements*  
*0 Rounds*

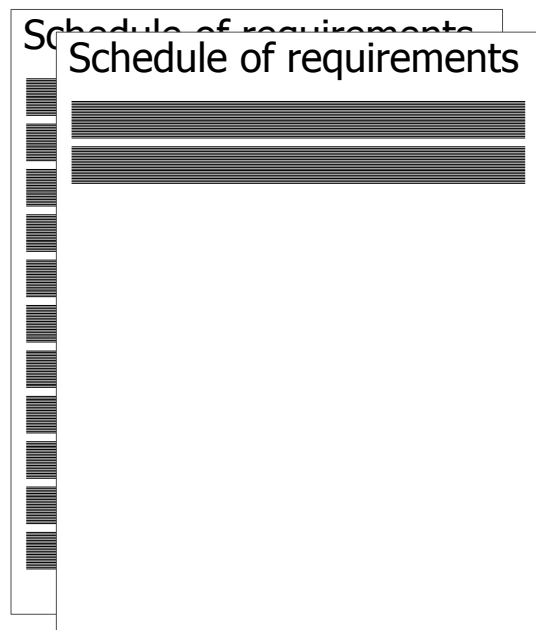
10

7

*3 Requirements*  
*2 Rounds*

J

A



Erik

Ferdinand

Imogen

Martin

8

9

5

2

*2 Requirements*  
*1 Rounds*

*2 Requirements*  
*1 Rounds*

- *Everyone identified new requirements and they again each receive 1 point*
- *"Ferdinand" won the round with the Jack so he gets 1 point for that, and leads the play for the final round – he also has the most points so far*

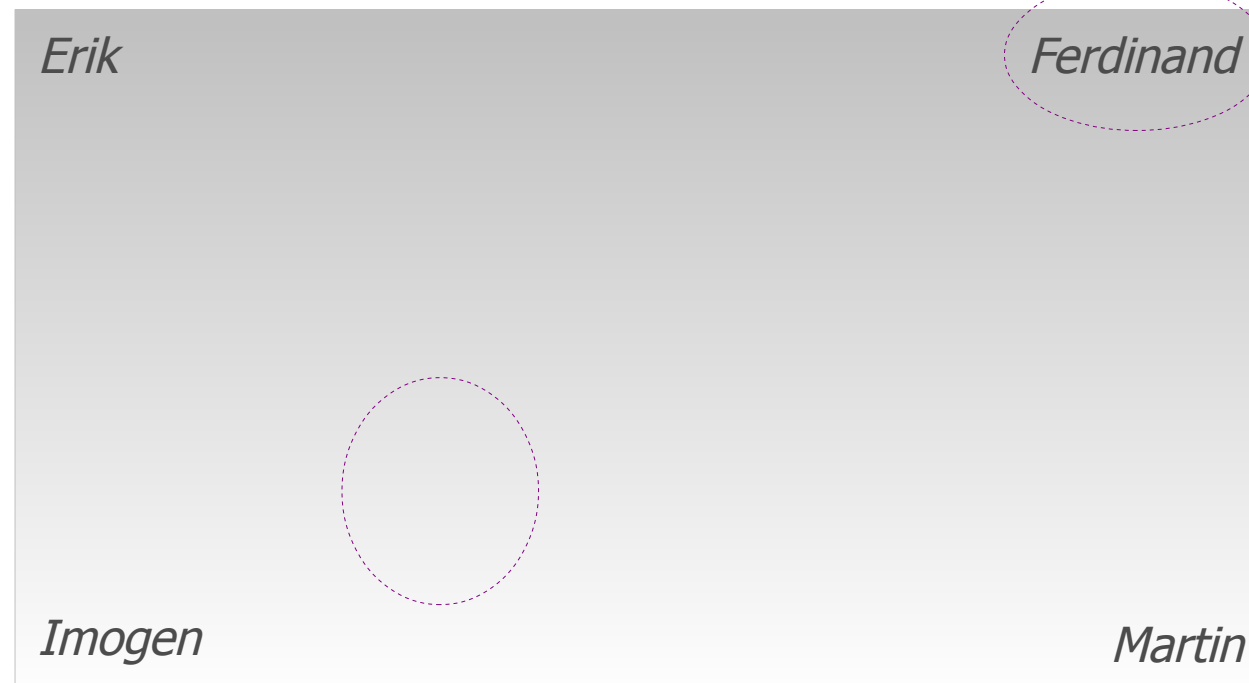
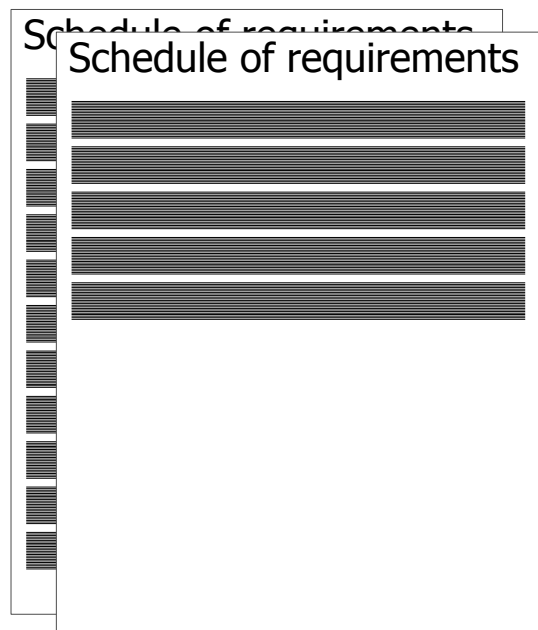
## Fifth and final round

3 Requirements  
0 Rounds

10

4 Requirements  
2 Rounds

A



- Everyone except "Erik" identified new requirements and they each receive 1 point
- "Imogen" won the round with the 8 (trumps) so she gets 1 point for that
- Overall Ferdinand wins the game with a total of 7 points

8

4 Requirements  
1 Rounds


2

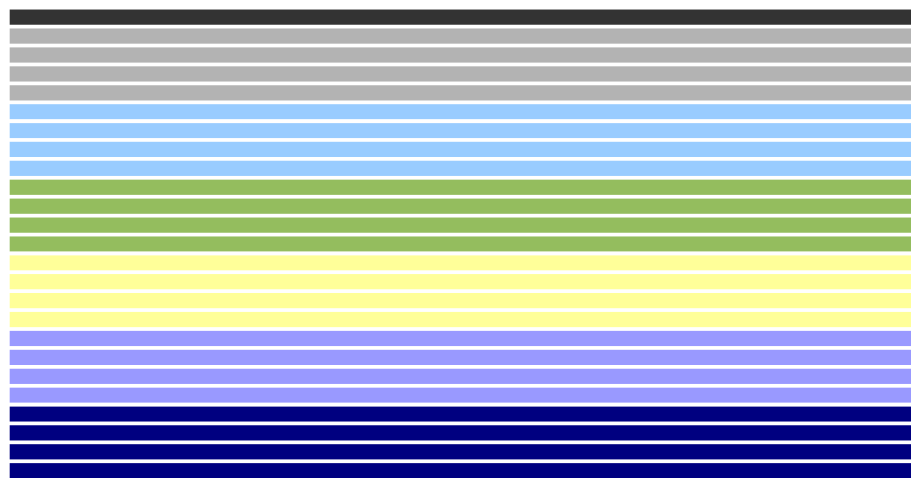
4 Requirements  
1 Rounds



# Choose your deck of cards

## Cornucopia suits

-  **Data validation and encoding**  
Input and output data validation and escaping
-  **Authentication**  
Verification of identity claims and related processes
-  **Session management**  
Maintenance of user state
-  **Authorization**  
User/role permission controls
-  **Cryptography**  
Hashing, digital signatures, encryption and random number generation processes and their usage including key management
-  **Cornucopia (everything else)**  
Everything else including information leakage, data loss, configuration management, denial of service

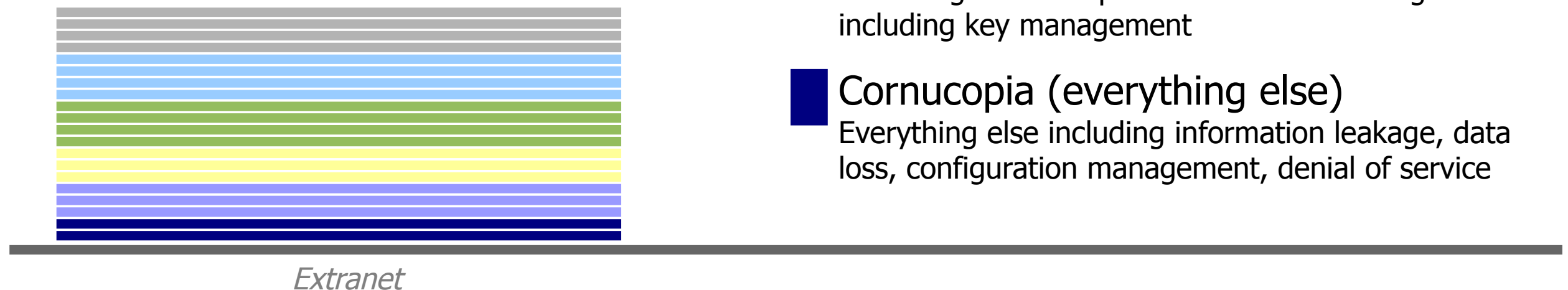


*Full deck*

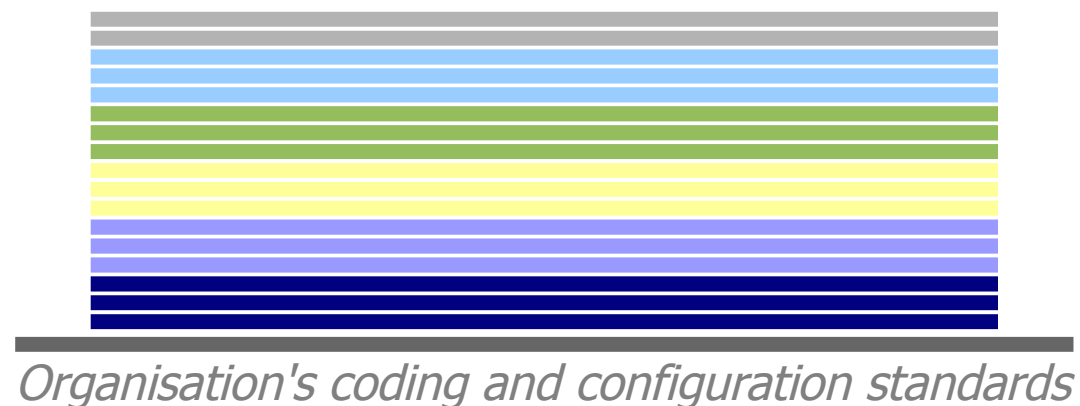
# Application-specific decks

## Cornucopia suits

-  **Data validation and encoding**  
Input and output data validation and escaping
-  **Authentication**  
Verification of identity claims and related processes
-  **Session management**  
Maintenance of user state
-  **Authorization**  
User/role permission controls
-  **Cryptography**  
Hashing, digital signatures, encryption and random number generation processes and their usage including key management
-  **Cornucopia (everything else)**  
Everything else including information leakage, data loss, configuration management, denial of service

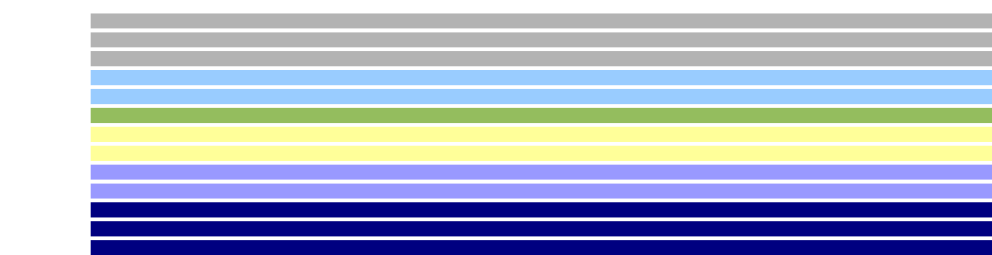


# Development-specific decks



or

*Compliance requirements (e.g. PCIDSS)*



## Cornucopia suits

- Data validation and encoding**  
Input and output data validation and escaping
- Authentication**  
Verification of identity claims and related processes
- Session management**  
Maintenance of user state
- Authorization**  
User/role permission controls
- Cryptography**  
Hashing, digital signatures, encryption and random number generation processes and their usage including key management
- Cornucopia (everything else)**  
Everything else including information leakage, data loss, configuration management, denial of service

# Does Cornucopia matter?



Security  
Standards Council

**Standard:** PCI Data Security  
**Version:** 2.0  
**Date:** January 2013  
**Author:** E-commerce Special  
PCI Security Standards Council

Information Supplement  
PCI DSS E-commerce Guidelines



Information Supplement • PCI DSS E-commerce Guidelines • January 2013

## 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

# Project plan

## Improvements

- Complete framework-specific card decks
- Enhance text and mappings
- Further developer feedback
- Issue further releases
- Graphical design
- Printing and distribution

## Multiple editions

- (Ecommerce website)
- Web services
- Mobile app
- Smart meter



# The project

## OWASP Cornucopia

- [https://www.owasp.org/index.php/OWASP\\_Cornucopia](https://www.owasp.org/index.php/OWASP_Cornucopia)
- [https://lists.owasp.org/mailman/listinfo/owasp\\_cornucopia](https://lists.owasp.org/mailman/listinfo/owasp_cornucopia)

## Download Ecommerce Website Edition v1.01

- [https://www.owasp.org/index.php/File:OWASP-Cornucopia-Ecommerce\\_Website.docx](https://www.owasp.org/index.php/File:OWASP-Cornucopia-Ecommerce_Website.docx)

## Colin Watson

- [colin.watson@owasp.org](mailto:colin.watson@owasp.org)