



# Cornucopia

## Edição de Ecomércio v1.30-PT-BR

OWASP® Cornucopia é um mecanismo para auxiliar equipes de desenvolvimento de software a identificar requisitos de segurança em processos de desenvolvimento Ágeis, convencionais e formais.

### Autor

Colin Watson

### Líderes de Projeto

Colin Watson and Grant Ongers

### Revisores

Tom Brennan, Johanna Curiel, Darío De Filippis and Timo Goosen

### Agradecimentos

Equipe Microsoft SDL para o "Jogo de Modelagem de Ameaças de Elevação de Privilégios", publicado sob uma licença "Creative Commons Attribution", como inspiração para Cornucopia e do qual muitas ideias, especialmente a teoria dos jogos, foram copiadas.

Keith Turpin e colaboradores do "OWASP Secure Coding Practices - Quick Reference Guide", originalmente doado ao OWASP pela Boeing, que é usado como fonte primária de informações sobre requisitos de segurança para formular o conteúdo dos cartões.

Colaboradores, apoiadores, patrocinadores e voluntários dos projetos OWASP ASVS, AppSensor e "Web Framework Security Matrix", "Common Attack Pattern Enumeration and Classification (CAPEC)" de Mitre e "Practical Security Stories and Security Tasks for Agile Development Environments" de SAFECode que são todos usados nas referências cruzadas fornecidas.

Playgen por fornecer um seminário esclarecedor à tarde sobre gamificação de tarefas, e tartanmaker.com pela ferramenta on-line para ajudar a criar o padrão do verso do cartão.

Blackfoot UK Limited por criar e doar arquivos de design prontos para impressão, Tom Brennan e a Fundação OWASP por instigar a criação de uma caixa e folheto com a marca OWASP, e funcionários da OWASP, especialmente Kate Hartmann, por gerenciar o pedido, estoque e envio de impressos baralhos de cartas. Oana Cornea e outros participantes na cimeira do projeto "AppSec EU 2015" pela sua ajuda na criação do vídeo de demonstração. Colin Watson como autor e co-líder do projeto com Grant Ongers, juntamente com outros voluntários do OWASP que ajudaram de diversas maneiras.

OWASP não endossa nem recomenda produtos ou serviços comerciais © 2012-2024 OWASP Foundation Este documento está licenciado sob a licença "Creative Commons Attribution-ShareAlike 3.0" license



## Introdução

A ideia por trás do Cornucopia é ajudar as equipes de desenvolvimento, especialmente aquelas que usam metodologias Agile, a identificar requisitos de segurança de aplicativos e desenvolver histórias de usuários baseadas em segurança. Embora a ideia já esperasse tempo suficiente para progredir, a motivação final veio quando a SAFECode publicou suas "Practical Security Stories and Security Tasks for Agile Development Environments" em julho de 2012.

A equipe do Microsoft SDL já havia publicado seu super "Elevation of Privilege: The Threat Modeling Game (EoP)", mas isso não parecia abordar o tipo mais apropriado de problemas que as equipes de desenvolvimento de aplicativos da web geralmente precisam resolver. EoP é um ótimo conceito e estratégia de jogo e foi publicado sob uma licença Creative Commons Attribution.

Cornucopia Website App Edition é baseado nos conceitos e ideias de jogos do EoP, mas eles foram modificados para serem mais relevantes para os tipos de problemas que os desenvolvedores de sites de comércio eletrônico encontram. Ele tenta introduzir ideias de modelagem de ameaças em equipes de desenvolvimento que usam metodologias Agile, ou estão mais focadas nos pontos fracos de aplicativos da web do que em outros tipos de vulnerabilidades de software ou não estão familiarizadas com STRIDE e DREAD.

Cornucopia Website App Edition é referenciado como um recurso de informações no Suplemento de informações do PCI Security Standard Council "PCI DSS E-commerce Guidelines", v2, janeiro de 2013.

## O baralho de cartas (pacote)

Em vez dos naipes STRIDE do EoP (conjuntos de cartas com designs correspondentes), os naipes Cornucopia são baseados na estrutura das "OWASP Secure Coding Practices - Quick Reference Guide (SCP)", mas com consideração adicional das seções do "OWASP Application Security Verification Standard", o "OWASP Testing Guide" e os "Principles of Secure Development" de David Rook. Estes forneceram cinco naipes, e um sexto chamado "Cornucópia" foi criado para todo o resto:

- Validação & Codificação de Dados (VE)
- Autenticação (AT)
- Gerenciamento de Sessões (SM)
- Controle de Acessos (AZ)
- Criptografia (C)
- Cornucopia (C)

Semelhante às cartas de pôquer, cada naipe contém 13 cartas (Ás, 2-10, Valete, Dama e Rei), mas, ao contrário do EoP, também existem duas cartas do Coringa. O conteúdo foi extraído principalmente do SCP.

## Mapeamentos

O outro motivador do Cornucopia é vincular os ataques a requisitos e técnicas de verificação. Um objetivo inicial era fazer referência aos IDs de pontos fracos do CWE, mas estes se mostraram muito numerosos e, em vez disso, foi decidido mapear cada placa para IDs de padrão de ataque de software CAPEC, que são mapeados para CWEs, para que o resultado desejado seja alcançado. Cada cartão também é mapeado para as 36 histórias de segurança primárias no documento SAFECode, bem como para o OWASP SCP v2, ASVS v4.0 e AppSensor (detecção e resposta a ataques de aplicativos) para ajudar as equipes a criar sua própria segurança. histórias relacionadas para uso em processos Agile.

## Estratégia de jogo

Além das diferenças de conteúdo, as regras do jogo são virtualmente idênticas às do EoP.

## Imprimindo os cartões

Verifique a página do projeto Cornucópia para saber como obter baralhos pré-impressos em cartão brilhante.

Os cartões podem ser impressos a partir deste documento em preto e branco, mas são mais eficazes em cores. Os cartões nas páginas posteriores deste documento foram dispostos para caber em um tipo de folha de cartão comercial A4 pré-cortada. Esta parecia ser a maneira mais rápida de criar inicialmente cartas de jogar rapidamente. Os códigos de produto Avery C32015 e C32030 foram testados com sucesso, mas quaisquer cartões de 10 até 85 mm x 54 mm em papel A4 devem funcionar com um pequeno ajuste. Outros fornecedores de papelaria como Ryman e Sigel produzem folhas semelhantes. Essas folhas de cartão não são baratas, portanto, deve-se tomar cuidado ao decidir o que imprimir e usar qual mídia e tipo de impressora.

É claro que os cartões podem ser impressos em qualquer tamanho de papel ou cartão e depois cortados manualmente, ou uma impressora comercial seria capaz de imprimir volumes maiores e cortar os cartões no tamanho certo. As linhas de corte são mostradas na penúltima página deste documento, mas a Avery também produz um modelo A4 paisagem (A-0017-01\_L.doc) que pode ser usado como guia.

Imprimir e cortar podem levar cerca de uma hora, e usar uma impressora mais rápida ajuda. Tente imprimir com qualidade superior para aumentar a legibilidade. Um desenho opcional do verso do cartão (em tartan OWASP) foi fornecido na última página deste documento. Não há necessidade de alinhamento especial. A impressão frente e verso requer cuidados especiais. Você pode personalizar as faces ou o verso do cartão de acordo com as preferências da sua organização.

## Costumização

Depois de usar o Cornucopia algumas vezes, você poderá achar que alguns cartões são menos relevantes para seus aplicativos ou que as ameaças são diferentes para sua organização. Edite você mesmo este documento para tornar as cartas mais adequadas para suas equipes ou crie novos baralhos completamente.

## Dar uma resposta

Se você tiver ideias ou comentários sobre o uso do OWASP® Cornucopia, compartilhe-os. Melhor ainda, se você criar versões alternativas dos cartões ou produzir versões profissionais prontas para impressão, compartilhe isso com os voluntários que criaram esta edição e com a comunidade mais ampla de desenvolvimento de aplicativos e segurança de aplicativos.

O melhor lugar para discutir ou contribuir é na lista ou grupo do projeto OWASP:

- Lista/Grupo  
[https://lists.owasp.org/mailman/listinfo/owasp\\_cornucopia](https://lists.owasp.org/mailman/listinfo/owasp_cornucopia)
- Página inicial do projeto  
[https://www.owasp.org/index.php/OWASP\\_Cornucopia](https://www.owasp.org/index.php/OWASP_Cornucopia)

Todos os documentos e ferramentas do OWASP são gratuitos para download e uso. OWASP® Cornucopia está licenciado sob a licença Creative Commons Attribution-ShareAlike 3.0.

## Instruções

O texto em cada carta descreve um ataque, sendo escolhido um nome para o atacante, o texto é único entre todas as cartas do jogo. O nome pode representar um sistema de computador (por exemplo um banco de dados, um sistema de arquivos, outra aplicação qualquer, um serviço relativo, um botnet), um indivíduo (por exemplo um cidadão, um cliente, um colaborador, um criminoso, um espião), ou até mesmo um grupo de pessoas (por exemplo uma organização competitiva, ativistas com uma causa em comum). O atacante pode estar remoto em algum outro aparelho/localização, ou local/interno com acesso ao mesmo aparelho, host ou rede na qual a aplicação está rodando. O atacante sempre é nomeado no começo de cada descrição. Um exemplo segue:

*William tem o controle sobre a geração de identificadores de sessão*

Isso significa que o atacante (William) pode criar novos identificadores de sessão que a aplicação aceita. Os ataques foram inicialmente desenhados a partir dos requisitos listados no SCP 'Secure Coding Practices' v2, suplementados com a verificação de objetivos do OWASP 'Application Security Verification Standard for Web Applications', com histórias focadas em segurança contidas em SAFECode's 'Practical Security Stories and Security Tasks for Agile Development Environments', e finalmente com uma revisão das cartas junto com EOP 'Elevation of Privilege': 'The Threat Modeling Game' criado pelo time da Microsoft SDL.

Um guia mais aprofundado sobre cada carta está disponível no Wiki Deck em

<https://wiki.owasp.org/index.php/Cornucopia - Ecommerce Website Edition - Wiki Deck>

Gabaritos entre os ataques e as cinco fontes estão providas na maioria das cartas:

- Requisitos em "Secure Coding Practices (SCP) - Quick Reference Guide", v2, OWASP®, Novembro 2010 (ref: [OWASP SCP Quick Reference Guide v2.1](#))
- Verification IDs em "Application Security Verification Standard (ASVS) para aplicativos da Web" (ref: [ASVS v3 and v4 downloads](#))
- Attack detection points IDs em "AppSensor", OWASP®, Agosto 2010-2015 (ref: [AppSensor DetectionPoints](#))
- IDs em "Common Attack Pattern Enumeration and Classification (CAPEC)", v2.8, Mitre Corporation, Novembro 2015 (ref: [capec \(31. July 2018\)](#))
- Histórias focadas em segurança em 'Practical Security Stories and Security Tasks for Agile Development Environments', SAFECode, Julho 2012 (ref: [SAFECode Agile Dev Security](#))

Uma "pesquisa" significa que o ataque está incluído no item referenciado, mas não abrange necessariamente toda a sua intenção. Para dados estruturados como CAPEC, é fornecida a referência mais específica, mas às vezes é fornecida uma referência cruzada que também contém exemplos mais específicos (filhos). Não há pesquisas sobre os seis Ases e dois Jokers. Em vez disso, esses cartões contêm algumas dicas gerais em itálico.

É possível jogar Cornucópia de muitas maneiras diferentes. Aqui está uma maneira, demonstrada online em um vídeo em (ref: [ColinWatsonOWASP](#)), que usa a nova folha de pontuação/registro (maio de 2015) em (refL [Cornucopia scoresheet](#))

## A - PREPAROS

- A1. Compre um baralho, ou imprima o seu próprio baralho de cartas Cornucopia (veja a página 2 deste documento)
- A2. Identifique uma aplicação ou o processo de uma aplicação para revisar; podendo ser um conceito, um design ou uma implementação
- A3. Criar um diagrama de fluxo de dados, user stories, ou outros diagramas para ajudar a revisão
- A4. Identificar e convidar um grupo de 3-6 arquitetos, desenvolvedores, testers e outros stakeholders, juntá-los e se sentar ao redor de uma mesa (tente incluir alguém suficientemente familiar com segurança de aplicações)
- A5. Tenha alguns prêmios para entregar (estrelas douradas, pizza, cerveja ou flores dependendo da cultura da sua empresa)

## B - JOGO

Um naipe – Cornucopia – age como o mais forte. Ás de cada naipe ganham de Reis. Fica mais fácil se alguém que não está jogando documente os problemas e as pontuações.

- B1. Retire os coringas e algumas cartas de pontuação baixa (2, 3, 4) do naipe Cornucopia para garantir que cada jogador tenha o mesmo número de cartas.
- B2. Embaralhe o baralho e dê as cartas
- B3. Para começar, escolha aleatoriamente quem irá jogar a primeira carta – pode-se jogar qualquer carta da sua mão com exceção do naipe mais forte – Cornucopia
- B4. Para jogar uma carta, cada jogador deve lê-la em voz alta, e explicar (veja as dicas do Wiki Deck na internet) como a ameaça pode ser aplicada (o jogador ganha um ponto por ataques que podem funcionar o qual o grupo acha que é um possível bug)
- B5. Jogue em sentido horário, cada jogador deve jogar a carta do mesmo modo, se você tem alguma carta do mesmo naipe que foi jogado você deve jogá-la, caso contrário pode-se jogar uma carta de qualquer outro naipe. Apenas a carta mais alta do mesmo naipe, ou a mais alta do naipe Cornucopia ganha a mão
- B6. O jogador que ganhar a rodada, começa a próxima mão, decidindo assim o próximo naipe
- B7. Repita o modo de jogo até que todas as cartas tenham sido jogadas

## C - Pontuação

O objetivo é identificar ameaças aplicáveis e ganhar mãos (rodadas):

- C1. Pontuação +1 para cada cartão que você puder identificar como uma ameaça válida ao aplicativo em consideração
- C2. Marque +1 se você vencer uma rodada
- C3. Depois que todas as cartas forem jogadas, quem tiver mais pontos vence

## D - Fecho

- D1. Revise todas as ameaças aplicáveis e os requisitos de segurança correspondentes
- D2. Crie histórias de usuários, especificações e casos de teste conforme necessário para sua metodologia de desenvolvimento.

## Regras alternativas do jogo

Se você é novo no jogo, remova os Ases e duas cartas do Coringa para começar. Adicione as cartas do Coringa de volta quando as pessoas se familiarizarem com o processo. Além das regras do “jogo de trunfos” descritas acima, que são muito semelhantes às do EoP, o baralho também pode ser jogado como o “jogo de vinte e uma cartas” (também conhecido como “pontão” ou “blackjack”), o que normalmente reduz o número de cartas jogadas em cada rodada.

Pratique em um aplicativo imaginário, ou mesmo em um aplicativo planejado para o futuro, em vez de tentar encontrar falhas nos aplicativos existentes até que os participantes estejam satisfeitos com a utilidade do jogo.

Considere apenas brincar com um naipe para fazer uma sessão mais curta – mas tente cobrir todos os naipes para cada projeto. Ou melhor ainda, apenas jogue uma mão com algumas cartas pré-selecionadas e pontue apenas na capacidade de identificar requisitos de segurança. Talvez faça uma partida de cada naipe por dia durante uma semana ou mais, se os participantes não puderem gastar tempo suficiente para um baralho completo.

Algumas equipes preferiram jogar uma mão inteira de cartas e depois discutir o que há nas cartas após cada rodada (em vez de depois de cada pessoa jogar uma carta).

Outra sugestão é que se um jogador não conseguir identificar se a carta é relevante, permita que outros jogadores sugiram ideias e, potencialmente, deixe-os ganhar o ponto pela carta. Considere permitir pontos extras para contribuições especialmente boas.

Você pode até jogar sozinho. Basta usar os cartões para agir como instigadores de pensamento. Envolver mais pessoas será benéfico.

Nas orientações EoP da Microsoft, eles recomendam trapaça como uma boa estratégia de jogo.

## Desenvolvimento de baralhos de cartas modificados específicos da estrutura

Podem ser integrados controles de segurança em algumas linguagens e frameworks comumente usados para desenvolvimento de aplicativos web e móveis. Com certas ressalvas, é útil considerar como o uso desses controles pode simplificar a identificação de requisitos adicionais – desde que, é claro, os controles estejam incluídos, habilitados e configurados corretamente.

Considere remover cartas dos baralhos se tiver certeza de que elas serão abordadas pela maneira como você está usando a linguagem/estrutura. Os itens entre parênteses são “talvez”.

## Padrões e bibliotecas de codificação interna

Adicione sua própria lista de cartões excluídos com base nos padrões de codificação da sua organização (desde que sejam confirmados por etapas de verificação apropriadas no ciclo de vida de desenvolvimento).

### Seus padrões de codificação e bibliotecas

Data validation and encoding [su lista]	Session management [su lista]	Cryptography [su lista]
Authentication [su lista]	Authorization [su lista]	Cornucopia [su lista]

## Conjuntos de requisitos de conformidade

Crie um baralho menor incluindo apenas cartas para um requisito de conformidade específico.

### Requisito de conformidade

Data validation and encoding [lista]	Session management [lista]	Cryptography [lista]
Authentication [lista]	Authorization [lista]	Cornucopia [lista]

## Perguntas frequentes

### 1. Posso copiar ou editar o jogo?

Sim claro. Todos os materiais do OWASP são gratuitos para você usar como quiser, desde que cumpra a licença Creative Commons Attribution-ShareAlike 3.0. Talvez se você criar uma nova versão, você possa doá-la para o Projeto Cornucópia OWASP?

### 2. Como posso participar?

Por favor, envie ideias ou ofertas de ajuda para a mailing list do projeto.

### 3. Como foram escolhidos os nomes dos agressores?

EoP começa cada descrição com palavras como 'Um invasor pode...'. Estas devem ser formuladas como um ataque, mas eu não gostei da terminologia anônima, querendo algo mais envolvente e, portanto, usei nomes pessoais. Eles podem ser considerados pessoas externas ou internas ou apelidos para sistemas de computador. Mas em vez de apenas nomes aleatórios, pensei em como eles poderiam refletir o aspecto da comunidade OWASP. Portanto, além de 'Alice e Bob', eu uso os (primeiros) nomes dos funcionários e membros do Conselho atuais e recentes da OWASP (atribuídos sem ordem) e, em seguida, selecionei aleatoriamente os 50 nomes restantes da lista atual de pagadores, membros individuais da OWASP. Nenhum nome foi usado mais de uma vez e, nos casos em que as pessoas forneceram dois nomes pessoais, deixei cair uma parte para tentar garantir que ninguém pudesse ser facilmente identificado. Os nomes não foram atribuídos deliberadamente a nenhum ataque, defesa ou requisito específico. A mistura cultural e de gênero reflecte simplesmente estas fontes de nomes e não pretende ser representativa mundial. Na v1.20, o nome do VE-10 foi alterado para refletir o novo co-líder do projeto – esta carta também é a única com dois nomes no ataque.

### 4. Por que não há imagens nas faces dos cartões?

Há muito texto nos cartões e as referências cruzadas também ocupam espaço. Mas seria ótimo incluir elementos de design adicionais. Qualquer voluntário

### 5. Os ataques são classificados pelo número da carta?

Apenas aproximadamente. O risco dependerá da aplicação e da organização, devido aos diversos requisitos de segurança e conformidade, portanto, sua própria classificação de gravidade poderá colocar os cartões em alguma ordem diferente da dos números nos cartões.

### 6. Quanto tempo leva para jogar uma rodada de cartas usando o baralho completo?

Isso depende de la portée de l'application, da quantidade de discussão e da familiaridade dos participantes com os conceitos de segurança de aplicativos. Mas talvez permita de 1,5 a 2,0 horas para 4 a 6 pessoas.

### 7. Que tipo de pessoas deveriam jogar?

Sempre tente ter uma combinação de funções que possam contribuir com perspectivas alternativas. Mas inclua alguém que tenha um conhecimento razoável da terminologia de vulnerabilidade de aplicativos. Caso contrário, tente incluir uma mistura de arquitetos, desenvolvedores, testadores e um gerente de projeto ou proprietário de empresa relevante.

### 8. Quem deve tomar notas e registrar as pontuações?

É melhor que outra pessoa, que não esteja jogando, faça anotações sobre os requisitos identificados e as questões discutidas. Isso poderia ser usado como treinamento para um desenvolvedor júnior ou realizado pelo gerente de projeto. Algumas organizações fizeram uma gravação para revisão posterior, quando os requisitos forem redigidos de forma mais formal.

### 9. Devemos sempre usar o baralho completo?

Não. Um deck menor é mais rápido de jogar. Comece seu primeiro jogo com cartas suficientes apenas para duas ou três rodadas. Sempre considere remover cartões que não sejam apropriados para o aplicativo ou função alvo que está sendo revisado. Nas primeiras vezes que as pessoas jogam, geralmente é melhor remover os Ases e os dois Jokers. Também é comum jogar sem nenhum naipe de trunfo até que as pessoas estejam mais familiarizadas com a ideia.

### 10. O que os jogadores devem fazer quando têm uma carta Ás que diz "inventaram um novo ataque X"?

O jogador pode inventar qualquer ataque que considere válido, mas deve corresponder ao naipe da carta, por exemplo, "data validation and encoding"). Com jogadores novos no jogo, pode ser melhor removê-los para começar (veja também a FAQ 9).

### 11. Não entendo o que significa o ataque em cada carta - há informações mais detalhadas?

Sim, o Wiki Deck online foi criado para ajudar os jogadores a entender os ataques. Ver

[https://www.owasp.org/index.php/Cornucopia - Ecommerce\\_Website\\_Edition - Wiki\\_Deck](https://www.owasp.org/index.php/Cornucopia - Ecommerce_Website_Edition - Wiki_Deck)

### 12. Minha empresa deseja imprimir sua própria versão do OWASP® Cornucopia, a qual licença precisamos nos referir? Consulte a resposta completa a esta pergunta nas páginas web do projeto em

[https://www.owasp.org/index.php/OWASP\\_Cornucopia - tab=FAQs](https://www.owasp.org/index.php/OWASP_Cornucopia - tab=FAQs)

VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS
<b>A</b> Você inventou um novo ataque contra a Validação & Codificação de Dados	<i>Leia mais sobre este tópico em OWASP Cheat Sheets. Pesquise sobre validação dos dados de entrada, Prevenção de XSS(Cross-site Scripting), Prevenção do DOM baseado em XSS, Prevenção de SQL Injection e Parametrização de Consultas</i>	(Nenhum Cartão)	Brian consegue reunir o básico de informações sobre a utilização e configuração de base de dados, lógica, codificação, além da utilização de softwares, serviços e infraestrutura nas mensagens de erro ou em mensagens de configuração, ou na presença de arquivos de instalação (padrões ou antigos), ou em evidências de testes, ou em backups ou em exposição de código fonte	<b>2</b>	Robert consegue inserir dados maliciosos pois o formato de protocolo não foi checado, ou duplicações são aceitas, ou a estrutura não está sendo verificada, ou os dados individuais não foram validados por formato, tipo, intervalo, tamanho e por uma lista de caracteres ou formatos possíveis
			OWASP SCP <u>69</u> , 107-109, 136, 137, 153, 156, 158, 162 OWASP ASVS <u>1.6.4</u> , 2.10.4, 4.3.2, 7.1.1, 10.2.3, 14.1.1, <u>14.2.2</u> , 14.3.3 OWASP APPSENSOR <u>HT1-3</u> CAPEC <u>54</u> , 541 SAFECODE <u>4</u> , 23 \${Common_Title_full}		OWASP SCP <u>-</u> OWASP ASVS <u>1.5.3</u> , 5.1.1-4, 13.2.1, 14.1.2, 14.4.1 OWASP APPSENSOR <u>RE7-8</u> , AE4-7, IE2-3, CIE1, CIE3-4, <u>HT1-3</u> CAPEC <u>28</u> , 48, 126, 165, 213, 220-221, 261-262, <u>271-272</u> SAFECODE <u>3</u> , 16, 24, 35 \${Common_Title_full}
<b>VALIDAÇÃO &amp; CODIFICAÇÃO DE DADOS</b>	<b>VALIDAÇÃO &amp; CODIFICAÇÃO DE DADOS</b>	<b>VALIDAÇÃO &amp; CODIFICAÇÃO DE DADOS</b>	Jee consegue ignorar as rotinas centralizadas de codificação de saída pois elas não estão sendo usadas em todos os lugares, ou a codificação errada está sendo usada	<b>6</b>	Jason consegue ignorar as rotinas centralizadas de validação de dados de entrada pois elas não estão sendo usadas em todos os campos de entrada de dados
			OWASP SCP <u>3</u> , 15, 18-22, 168 OWASP ASVS <u>1.1.6</u> , 1.5.3, 5.1.3, 13.2.2, 13.2.5 OWASP APPSENSOR <u>-</u> CAPEC <u>28</u> , 31, 152, 160, 468 SAFECODE <u>2</u> , 17 \${Common_Title_full}		OWASP SCP <u>4-5</u> , 7, 150 OWASP ASVS <u>1.5.3</u> , 13.2.2, 13.2.5 OWASP APPSENSOR <u>IE2</u> , IE3 CAPEC <u>28</u> , 153, 165 SAFECODE <u>3</u> , 16, 24 \${Common_Title_full}

VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS	VALIDAÇÃO & CODIFICAÇÃO DE DADOS
<b>8</b> Oana consegue ignorar as rotinas centralizadas de tratamento (sanitização) pois elas não estão sendo usadas de forma abrangente  OWASP SCP 15, 169 OWASP ASVS 1.1.6, 5.2.2, 5.2.5 OWASP APPSENSOR - CAPEC 28, 31, 152, 160, 468 SAFECODE 2, 17 \${Common_Title_full}	<b>9</b> Shamun consegue ignorar as verificações de validação de entrada ou de saída porque as falhas de validação não são rejeitadas e/ou tratadas (sanitização)  OWASP SCP 6, 21-22, 168 OWASP ASVS 7.1.3 OWASP APPSENSOR IE2, IE3 CAPEC 28 SAFECODE 3, 16, 24 \${Common_Title_full}	<b>10</b> Dario consegue explorar a confiabilidade da aplicação em fonte de dados (ex: dados definidos pelo usuário, manipulação de dados armazenados localmente, mudança do estado dos dados em dispositivos clientes, falta de verificação da identidade durante uma validação de dados, como Dario pode fingir ser Colin)  OWASP SCP 2, 19, 92, 95, 180 OWASP ASVS 1.12.2, 5.1.3, 9.2.3, 12.2.1, 12.3.1-3, 12.4.2, 12.5.2, 14.5.3 OWASP APPSENSOR IE4, IE5 CAPEC 12, 51, 57, 90, 111, 145, 194-195, 202, 218, 463 SAFECODE 14 \${Common_Title_full}	<b>J</b> Toby tem o controle sobre validações de entrada de dados, validações de saída de dados ou codificação de saída ou rotinas que ele consegue ignorar/burlar  OWASP SCP 1, 17 OWASP ASVS 1.5.3 OWASP APPSENSOR RE3, RE4 CAPEC 87, 207, 554 SAFECODE 2, 17 \${Common_Title_full}	
<b>Q</b> Xavier consegue injetar dados num dispositivo ou num interpretador no lado do cliente porque uma interface parametrizada não foi usada, ou não foi implementada corretamente, ou os dados não foram codificados corretamente para o contexto proposto, ou não há uma política restritiva para a codificação ou a inclusão de dados  OWASP SCP 10, 15-16, 19-20 OWASP ASVS 5.2.1, 5.2.5, 5.3.3, 5.5.4 OWASP APPSENSOR IE1, RP3 CAPEC 28, 31, 152, 160, 468 SAFECODE 2, 17 \${Common_Title_full}	<b>K</b> Gabe consegue injetar dados num interpretador no lado do servidor (ex: SQL, comandos para o sistema operacional, Xpath, Server JavaScript, SMTP) porque uma interface parametrizada não foi usada ou não foi implementada corretamente  OWASP SCP 15, 19-22, 167, 180, 204, 211, 212 OWASP ASVS 5.2.1, 5.2.2, 5.3.4, 5.3.7-10 OWASP APPSENSOR CIE1, CIE2 CAPEC 23, 28, 76, 152, 160, 261 SAFECODE 2, 19-20 \${Common_Title_full}		(Nenhum Cartão)	(Nenhum Cartão)

AUTENTICAÇÃO	A	AUTENTICAÇÃO	(Nenhum Cartão)	AUTENTICAÇÃO	2	AUTENTICAÇÃO	3				
	Você inventou um novo ataque contra a Autenticação				James pode assumir as funções de autenticação sem que o usuário real esteja ciente do uso destas funções (ex: tentar fazer login, logar com credenciais, redefinir a senha)		Muhammad consegue obter a senha de um usuário ou outros dados, pela observação durante a autenticação, ou cache local, ou pela memória, ou pelo tráfego de dados, ou pela leitura de algum local desprotegido, ou porque isto é amplamente conhecido, ou porque não há expiração de dados, ou por que o usuário não consegue trocar sua própria senha				
AUTENTICAÇÃO	<i>Leia mais sobre este tópico em OWASP Authentication Cheat Sheet</i>			AUTENTICAÇÃO	<p>OWASP SCP 47, 52</p> <p>OWASP ASVS 2.5.2, 7.1.2, 7.1.4, 7.2.1, 8.2.1, 8.2.2, 8.2.3, 8.3.6</p> <p>OWASP APPSENSOR UT1</p> <p>CAPEC</p> <p>-</p> <p>SAFECODE 28</p> <p>{Common_Title_full}</p>		AUTENTICAÇÃO	<p>OWASP SCP 36-37, 40, 43, 48, 51, 119, 139-140, 146</p> <p>OWASP ASVS 2.5.2, 2.5.3</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 37, 546</p> <p>SAFECODE 28</p> <p>{Common_Title_full}</p>			
AUTENTICAÇÃO	4	AUTENTICAÇÃO	5	AUTENTICAÇÃO	6	AUTENTICAÇÃO	7				
	Sebastien pode identificar facilmente nomes de usuários ou consegue elencar quem eles são		Javier pode usar credenciais padrões (default), de teste ou facilmente adivinhadas para autenticação, ou consegue autenticar através de contas inativas ou autentica-se por contas não necessariamente da aplicação		Sven consegue reutilizar uma senha temporária porque o usuário não precisa troca-la no primeiro acesso, ou o tempo de expiração é muito longo, ou o tempo de expiração não existe, ou não é usado um método de entrega out-of-band (ex: aplicação mobile, SMS)		Cecilia consegue usar força bruta e ataques de dicionário (dictionary attacks) contra uma ou muitas contas sem limitação, ou estes ataques são simplificados pois as senhas tem baixa complexidade, tamanho reduzido, inexistência de expiração e regras para reuso				
AUTENTICAÇÃO	<p>OWASP SCP 33, 53</p> <p>OWASP ASVS 2.2.1, 4.1.5</p> <p>OWASP APPSENSOR AE1</p> <p>CAPEC 383</p> <p>SAFECODE 28</p> <p>{Common_Title_full}</p>		<p>OWASP SCP 54, 175, 178</p> <p>OWASP ASVS 4.1.5</p> <p>OWASP APPSENSOR AE12, IT3</p> <p>CAPEC 70</p> <p>SAFECODE 28</p> <p>{Common_Title_full}</p>		<p>OWASP SCP 37, 45-46, 178</p> <p>OWASP ASVS 2.5.6</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 50</p> <p>SAFECODE 28</p> <p>{Common_Title_full}</p>		<p>OWASP SCP 33, 38-39, 41, 50, 53</p> <p>OWASP ASVS 2.1.2, 2.1.7, 2.1.10, 2.2.1</p> <p>OWASP APPSENSOR AE2, AE3</p> <p>CAPEC 2, 16</p> <p>SAFECODE 27</p> <p>{Common_Title_full}</p>				
							OWASP Cornucopia Ecommerce Website Edition v1.20-EN				

AUTENTICAÇÃO	8	AUTENTICAÇÃO	9	AUTENTICAÇÃO	10	AUTENTICAÇÃO	J
	Kate consegue ignorar a autenticação porque isto não é uma falha de segurança (ex: o acesso sem autenticação está assinalado como padrão)		Claudia consegue assumir funções críticas porque os requisitos de autenticação são muito fracos (ex: não é usado autenticação com força de senha), ou não é um requisito revalidar a autenticação com frequência		Pravin consegue ignorar controle de autenticação porque não está sendo usado um módulo/framework/serviço de autenticação que seja centralizado, testado, comprovado e aprovado para gerir requisições		Mark consegue acessar recursos ou serviços porque não há requisitos de autenticação, ou, por engano, um outro sistema ou outra ação realizou autenticação
AUTENTICAÇÃO	OWASP SCP 28	AUTENTICAÇÃO	OWASP SCP 55-56	AUTENTICAÇÃO	OWASP SCP 25-27	AUTENTICAÇÃO	OWASP SCP 23, 32, 34
	OWASP ASVS 4.1.5		OWASP ASVS 1.4.5, 2.1.6, 2.2.4, 4.1.3, 4.3.3		OWASP ASVS 1.1.6, 1.4.4		OWASP ASVS 1.4.5, 4.3.1
AUTENTICAÇÃO	OWASP APPSENSOR ~	AUTENTICAÇÃO	OWASP APPSENSOR ~	AUTENTICAÇÃO	OWASP APPSENSOR ~	AUTENTICAÇÃO	OWASP APPSENSOR ~
	CAPEC 115		CAPEC 21		CAPEC 90, 115		CAPEC 115
AUTENTICAÇÃO	SAFECODE 28	AUTENTICAÇÃO	SAFECODE 14, 28	AUTENTICAÇÃO	SAFECODE 14, 28	AUTENTICAÇÃO	SAFECODE 14, 28
	\${Common_Title_full}		\${Common_Title_full}		\${Common_Title_full}		\${Common_Title_full}
AUTENTICAÇÃO	Q	AUTENTICAÇÃO	K				
	Johan consegue ignorar a autenticação porque não é aplicado o mesmo rigor para todas as funções de autenticação (ex: logar, troca de senha, recuperação de senha, logout, acesso administrador) ou não é aplicado o mesmo rigor nos diversos locais de acesso e versões do sistema(ex:mobile website, mobile app, full website, API, call center)		Olga consegue influenciar ou alterar o código ou a rotina de autenticação e com isto ignorar a autenticação		(Nenhum Cartão)		(Nenhum Cartão)
AUTENTICAÇÃO	OWASP SCP 23, 29, 42, 49	AUTENTICAÇÃO	OWASP SCP 24		OWASP SCP 24		OWASP SCP 23, 32, 34
	OWASP ASVS 1.4.5, 2.5.6, 2.5.7, 4.3.1		OWASP ASVS 4.1.1, 10.2.3, 10.2.4-6		OWASP ASVS 4.1.1, 10.2.3, 10.2.4-6		OWASP ASVS 1.4.5, 4.3.1
AUTENTICAÇÃO	OWASP APPSENSOR ~	AUTENTICAÇÃO	OWASP APPSENSOR ~		OWASP APPSENSOR ~		OWASP APPSENSOR ~
	CAPEC 36, 50, 115, 121, 179		CAPEC 115, 207, 554		CAPEC 115, 207, 554		CAPEC 115
AUTENTICAÇÃO	SAFECODE 14, 28	AUTENTICAÇÃO	SAFECODE 14, 28		SAFECODE 14, 28		SAFECODE 14, 28
	\${Common_Title_full}		\${Common_Title_full}		\${Common_Title_full}		\${Common_Title_full}

GERENCIAMENTO DE SESSÕES	GERENCIAMENTO DE SESSÕES	GERENCIAMENTO DE SESSÕES	GERENCIAMENTO DE SESSÕES	GERENCIAMENTO DE SESSÕES
A	(Nenhum Cartão)		William tem o controle sobre a geração de identificadores de sessão	Ryan consegue usar uma única conta em paralelo, pois as sessões simultâneas são permitidas
Leia mais sobre este tópico em OWASP Session Management Cheat Sheet e prevenção de ataques do tipo Cross Site Request Forgery (CSRF)			OWASP SCP 58-59	OWASP SCP 68
			OWASP ASVS 3.7.1	OWASP ASVS 3.3.3, 3.3.4
			OWASP APPSENSOR SE2	OWASP APPSENSOR -
			CAPEC 31, 60-61	CAPEC -
			SAFECODE 28	SAFE CODE 28
			{\$Common_Title_full}	{\$Common_Title_full}
4	GERENCIAMENTO DE SESSÕES	5	GERENCIAMENTO DE SESSÕES	GERENCIAMENTO DE SESSÕES
Alison consegue configurar identificadores de cookies em outras aplicações web porque o domínio ou o caminho não são suficientemente limitados		John consegue prever ou adivinhar identificadores de sessão porque estes não são alterados quando uma regra de usuário é alterada (ex: antes e depois da autenticação) e quando uma troca entre meios de comunicação criptografados e não criptografados acontece, ou os identificadores são curtos e não randômicos, ou não são modificados periodicamente	Gary consegue ter o controle da sessão de um usuário porque o tempo de encerramento(timeout) da sessão é longo ou inexiste, ou o tempo limite da sessão é longo ou inexiste, ou a mesma sessão pode ser usada para mais de um dispositivo/local	Graham consegue utilizar a sessão de Adam depois dele ter finalizado o uso da aplicação, porque a função de logout inverte, ou Adam não fez logout, ou a função de logout não termina a sessão de forma adequada

GERENCIAMENTO DE SESSÕES	8	GERENCIAMENTO DE SESSÕES	9	GERENCIAMENTO DE SESSÕES	10	J
	<p>Matt consegue utilizar longas sessões porque a aplicação não solicita uma nova autenticação de forma periódica para validar se os privilégios do usuário foram alterados</p> <p>OWASP SCP 96</p> <p>OWASP ASVS 3.3.2, 3.6.1</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 21</p> <p>SAFECODE 28</p> <p>{Common_Title_full}</p>	<p>Ivan consegue roubar identificadores de sessão porque estes são transmitidos em canais inseguros, ou estão logados, ou são exibidos em mensagens de erros, ou estão em URLs, ou são acessíveis pelo código que o atacante consegue alterar ou influenciar</p> <p>OWASP SCP 69, 75-76, 119, 138</p> <p>OWASP ASVS 1.9.1, 3.1.1, 7.1.1, 7.1.2, 7.2.1, 9.1.3, 9.2.2</p> <p>OWASP APPSENSOR SE4-6</p> <p>CAPEC 31, 60</p> <p>SAFECODE 28</p> <p>{Common_Title_full}</p>	<p>Marce consegue inventar requisições porque tokens randômicos e fortes (ou seja, tokens anti-CSRF) ou similares não estão sendo usados para ações que mudam estado. Estas requisições podem ser por sessão ou por requisição (request) em ações mais críticas</p> <p>OWASP SCP 73-74</p> <p>OWASP ASVS 4.2.2</p> <p>OWASP APPSENSOR IE4</p> <p>CAPEC 62, 111</p> <p>SAFECODE 18</p> <p>{Common_Title_full}</p>	<p>Jeff consegue reenviar uma interação de repetição idêntica (ex: requisição HTTP, sinal, botão pressionado) e ela é aceita, sem rejeição</p> <p>OWASP SCP -</p> <p>OWASP ASVS 11.1.1, 11.1.2, 11.1.3</p> <p>OWASP APPSENSOR IE5</p> <p>CAPEC 60</p> <p>SAFECODE 12, 14</p> <p>OWASP Cornucopia Ecommerce Website Edition v1.20-EN</p>		
GERENCIAMENTO DE SESSÕES	Q	GERENCIAMENTO DE SESSÕES	K	(Nenhum Cartão)	(Nenhum Cartão)	
	<p>Salim consegue ignorar o gerenciamento de sessão porque este não é aplicado de forma abrangente e consistente por toda a aplicação</p> <p>OWASP SCP 58</p> <p>OWASP ASVS 1.1.6, 3.7.1</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 21</p> <p>SAFECODE 14, 28</p> <p>{Common_Title_full}</p>	<p>Peter consegue ignorar o controle de gerenciamento de sessão porque este foi autoconstruído e/ou é fraco, ao invés de ter sido usado a estrutura padrão de um framework ou um módulo testado e aprovado</p> <p>OWASP SCP 58, 60</p> <p>OWASP ASVS 1.1.6</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 21</p> <p>SAFECODE 14, 28</p> <p>{Common_Title_full}</p>				

CONTROLE DE ACESSOS	A	CONTROLE DE ACESSOS	(Nenhum Cartão)	2	CONTROLE DE ACESSOS	3
	Você inventou um novo ataque contra Controle de Acessos  <i>Leia mais sobre este tópico em OWASP Development Guide e OWASP Testing Guide</i>			Tim consegue alterar nomes/endereços (paths) onde os dados são enviados ou encaminhados para alguém		Christian consegue acessar informações, que ele não deveria ter permissão, por meio de outro mecanismo que tenha permissão (ex: indexador de pesquisa, log, relatórios) ou porque a informação está armazenada em cache, ou mantida por mais tempo do que o necessário, ou outra vazamento de informação
CONTROLE DE ACESSOS	4	CONTROLE DE ACESSOS	5	CONTROLE DE ACESSOS	6	CONTROLE DE ACESSOS
	Kelly consegue ignorar controles de acesso porque estes não falham seguramente (ex: a permissão de acesso está assinalada como padrão)		Chad consegue acessar recursos que não deveria ter acesso devido a inexistência de uma autorização ou por concessão de privilégios excessivos (ex: não usar o princípio de menor privilégio possível). Os recursos podem ser serviços, processos, AJAX, Flash, vídeo, imagens, documentos, arquivos temporários, dados de sessão, propriedades do sistema, dados de configuração, logs	Eduardo consegue acessar dados que ele não tem permissão embora ele tem permissão em formulários, páginas, URL ou pontos de entrada		Yuanjing consegue acessar funções, telas e propriedades do aplicativo, a qual ele não está autorizado a ter acesso
CONTROLE DE ACESSOS	OWASP SCP 79-80	CONTROLE DE ACESSOS	OWASP SCP 70, 81, 83, 84, 87-9, 99, 117, 131, 132, 142, 154, 170, 179	OWASP SCP 81, 88, 131	CONTROLE DE ACESSOS	OWASP SCP 81, 85-86, 131
	OWASP ASVS 4.1.5		OWASP ASVS 1.2.2, 4.1.1, 4.1.3, 4.2.1	OWASP ASVS 4.1.3, 4.2.1		OWASP ASVS 4.1.3, 4.2.1
CONTROLE DE ACESSOS	OWASP APPSENSOR -	CONTROLE DE ACESSOS	OWASP APPSENSOR ACE1, ACE2, ACE3, ACE4, HT2	OWASP APPSENSOR ACE1-4	CONTROLE DE ACESSOS	OWASP APPSENSOR ACE1-4
	CAPEC 122		CAPEC 75, 87, 95, 126, 149, 155, 203, 213, 264-265	CAPEC 122		CAPEC 122
CONTROLE DE ACESSOS	SAFECODE 8, 10-11	CONTROLE DE ACESSOS	SAFECODE 8, 10-11, 15	SAFECODE 8, 10-11	CONTROLE DE ACESSOS	SAFECODE 8, 10-11
	§Common_Title_full}		§Common_Title_full}	§Common_Title_full}		§Common_Title_full}

CONTROLE DE ACESSOS	8	CONTROLE DE ACESSOS	9	CONTROLE DE ACESSOS	10	CONTROLE DE ACESSOS	J
	<p>Tom consegue ignorar regras de negócios alterando o fluxo/sequência usual do processo, ou realizando o processo na forma incorreta, ou manipulando valores de data e hora usados pela aplicação, ou usando recursos válidos para fins não intencionais, ou pela manipulação incorreta do controle de dados</p> <p>OWASP SCP 10, 32, 93-94, 189</p> <p>OWASP ASVS 4.1.2, 4.2.1, 4.3.3, 7.3.4, 11.1.1, 11.1.2</p> <p>OWASP APPSENSOR ACE3</p> <p>CAPEC 25, 39, 74, 162, 166, 207</p> <p>SAFECODE 8, 10-12</p> <p>{Common_Title_full}</p>		<p>Mike consegue usar indevidamente uma aplicação quando uma funcionalidade é usada de forma muito rápida, ou com muita frequência, ou de outra maneira a qual a funcionalidade não se destina, ou pelo consumo de recursos da aplicação ou pela condição de corrida (race conditions) ou utilização excessiva da funcionalidade</p> <p>OWASP SCP 94</p> <p>OWASP ASVS 11.1.3, 11.1.4</p> <p>OWASP APPSENSOR AE3, FIO1-2, UT2-4, STE1-3</p> <p>CAPEC 26, 29, 119, 261</p> <p>SAFECODE 1, 35</p> <p>{Common_Title_full}</p>		<p>Richard consegue ignorar os controles de acesso centralizados pois estes não estão sendo utilizados de forma abrangente em todas as interações</p> <p>OWASP SCP 78, 91</p> <p>OWASP ASVS 1.1.6, 4.1.1</p> <p>OWASP APPSENSOR ACE1-4</p> <p>CAPEC 36, 95, 121, 179</p> <p>SAFECODE 8, 10-11</p> <p>{Common_Title_full}</p>		<p>Dinis consegue acessar informações referente a configurações de segurança ou consegue acessar a lista de controle de acesso</p> <p>OWASP SCP 89-90</p> <p>OWASP ASVS 4.1.2, 10.2.3-6</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 75, 133, 203</p> <p>SAFECODE 8, 10-11</p> <p>{Common_Title_full}</p>
CONTROLE DE ACESSOS	Q	CONTROLE DE ACESSOS	K				
	<p>Christopher consegue injetar um comando que a aplicação vai executar no mais alto nível de privilégio</p> <p>OWASP SCP 209</p> <p>OWASP ASVS 5.3.8</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 17, 30, 69, 234</p> <p>SAFECODE 8, 10-11</p> <p>{Common_Title_full}</p>		<p>Ryan consegue influenciar ou alterar controles de acesso e permissões e consegue ignorá-los</p> <p>OWASP SCP 77, 89, 91</p> <p>OWASP ASVS 4.1.1, 4.1.2, 10.2.3-6</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 207, 554</p> <p>SAFECODE 8, 10-11</p> <p>{Common_Title_full}</p>		(Nenhum Cartão)		(Nenhum Cartão)

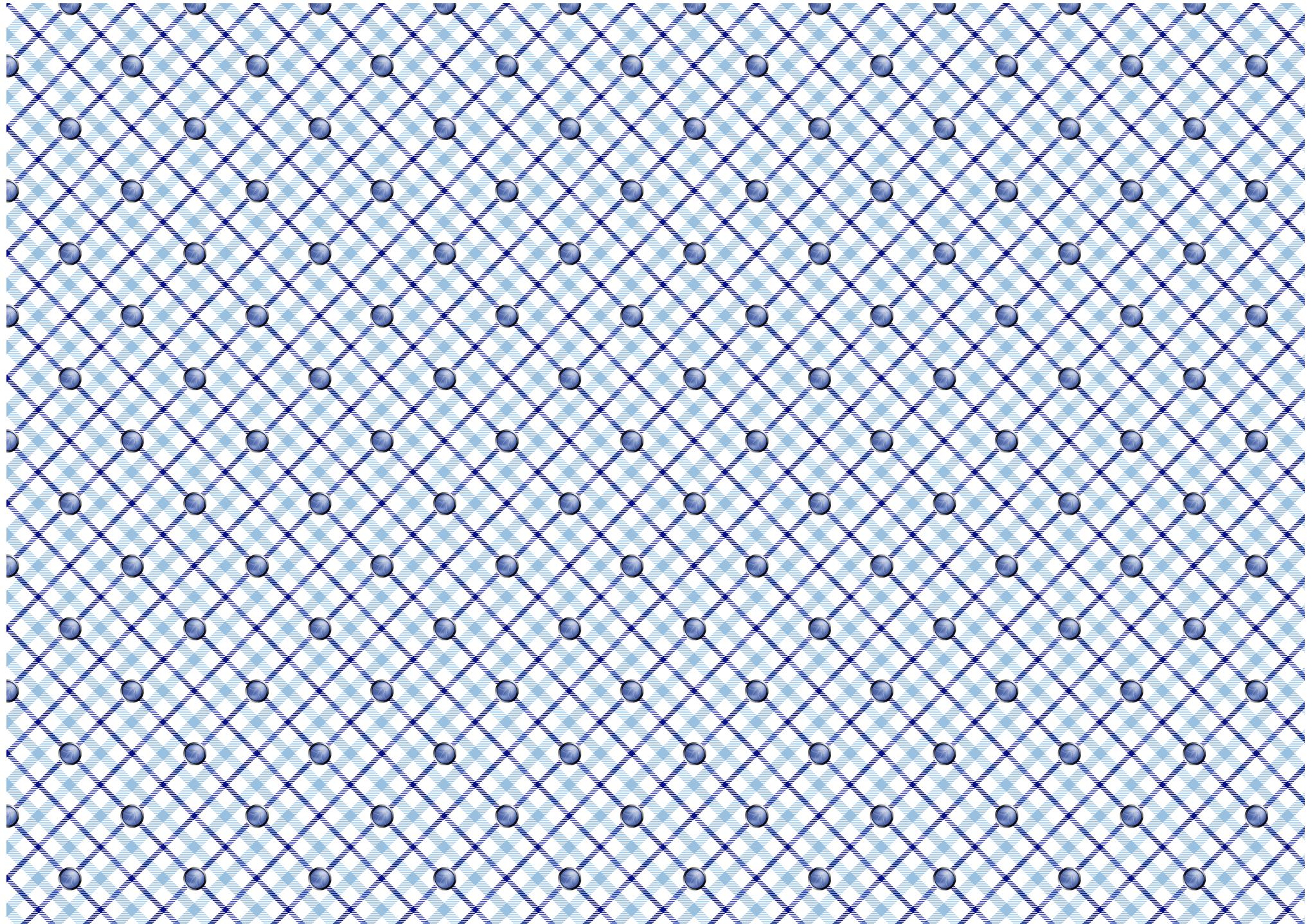
PRÁTICAS DE CRIPTOGRAFIA	<p><b>A</b></p> <p>Você inventou um novo ataque contra Criptografia</p> <p><i>Leia mais sobre este tópico em OWASP Cryptographic Storage Cheat Sheet e OWASP Transport Layer Protection Cheat Sheet</i></p>	PRÁTICAS DE CRIPTOGRAFIA	<p>(Nenhum Cartão)</p>	PRÁTICAS DE CRIPTOGRAFIA	<p>Kyun consegue acesso a dados porque isto foi ocultado/ofuscado/escondido ao invés de ser usada uma função de criptografia aprovada</p>
PRÁTICAS DE CRIPTOGRAFIA	<p><b>4</b></p> <p>Paulo consegue acesso a dados transitórios não criptografados, embora o canal de comunicação esteja criptografado</p> <p>OWASP SCP 37, 88, 143, 214</p> <p>OWASP ASVS 8.3.4, 9.1.1</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 185-187</p> <p>SAFECODE 14, 29-30</p> <p>Common_Title_full}</p>	PRÁTICAS DE CRIPTOGRAFIA	<p><b>5</b></p> <p>Kyle consegue ignorar controles criptográficos porque eles não falham de forma segura (ex: eles são desprotegidos por padrão)</p> <p>OWASP SCP 103, 145</p> <p>OWASP ASVS 1.9.1, 6.2.1, 9.1.3, 9.2.2</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC</p> <p>-</p> <p>SAFECODE 21, 29</p> <p>Common_Title_full}</p>	PRÁTICAS DE CRIPTOGRAFIA	<p><b>6</b></p> <p>Romain consegue ler e modificar dados descriptografados que estão na memória ou são transitórios (ex: credenciais, identificadores de sessão, dados pessoais e comercialmente relevantes), em uso ou em comunicação dentro da aplicação, ou entre aplicação e usuário, ou entre a aplicação e sistemas externos</p> <p>OWASP SCP 36-37, 143, 146-147</p> <p>OWASP ASVS 1.9.1, 2.2.5, 2.5.1, 8.3.4, 8.3.6, 9.1.3, 9.2.2</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 31, 57, 102, 157-158, 384, 466, 546</p> <p>SAFECODE 29</p> <p>Common_Title_full}</p>
PRÁTICAS DE CRIPTOGRAFIA	<p><b>7</b></p> <p>Gunter consegue interceptar ou modificar dados criptografados em trânsito porque o protocolo está mal implantado, ou configurado de forma fraca, ou os certificados estão inválidos, ou os certificados não são confiáveis, ou a conexão pode ser deteriorada para uma comunicação mais fraca ou descriptografada</p> <p>OWASP SCP 75, 144-145, 148</p> <p>OWASP ASVS 1.9.2, 6.2.7, 9.1.1, 9.2.1, 9.2.4, 14.4.5</p> <p>OWASP APPSENSOR</p> <p>IE4</p> <p>CAPEC 31, 216</p> <p>SAFECODE 14, 29-30</p> <p>Common_Title_full}</p>	PRÁTICAS DE CRIPTOGRAFIA	<p><b>2</b></p> <p>Axel consegue modificar dados que estão armazenados ou que são temporários ou transitórios, ou consegue modificar código fonte, ou consegue modificar patches/atualizações, ou alterar dados de configuração, pois a integridade não foi checada</p> <p>OWASP SCP 92, 205, 212</p> <p>OWASP ASVS 10.2.3-6, 10.3.1, 10.3.2, 14.1.1, 14.1.4, 14.1.5</p> <p>OWASP APPSENSOR</p> <p>SE1, IE4</p> <p>CAPEC 31, 39, 68, 75, 133, 145, 162, 203, 438-439, 442</p> <p>SAFECODE 12, 14</p> <p>Common_Title_full}</p>	PRÁTICAS DE CRIPTOGRAFIA	<p><b>3</b></p>

PRÁTICAS DE CRIPTOGRAFIA	8	PRÁTICAS DE CRIPTOGRAFIA	9	PRÁTICAS DE CRIPTOGRAFIA	10	J
	<p>Eoin consegue acesso a dados de negócios armazenados (ex: senhas, identificadores de sessão, informações de identificação pessoal - PII, dados de titular de cartão) pois estes dados não estão criptografados de forma segura ou com segurança</p> <p>OWASP SCP 30-31, 70, 133, 135</p> <p>OWASP ASVS 2.4.1, 6.2.2, 6.2.3, 8.3.4</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 31, 37, 55</p> <p>SAFECODE 21, 29, 31</p> <p>{Common_Title_full}</p>	<p>Andy consegue ignorar a geração de números aleatórios/randômicos, ou ignorar a geração aleatória de GUID, ou ignorar as funções de criptografia e hashing porque eles são fracos ou foram autoconstruídos</p> <p>OWASP SCP 60, 104-105</p> <p>OWASP ASVS 6.2.2, 6.2.3, 6.3.1, 6.3.3</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 97</p> <p>SAFECODE 14, 21, 29, 32-33</p> <p>{Common_Title_full}</p>	<p>Susanna consegue quebrar a criptografia em uso pois a criptografia não é forte o suficiente para oferecer a proteção exigida, ou esta não é forte o suficiente para tratar a quantidade de esforço que o atacante está disposto a fazer</p> <p>OWASP SCP 104-105</p> <p>OWASP ASVS 6.3.3</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 97, 463</p> <p>SAFECODE 14, 21, 29, 31-33</p> <p>{Common_Title_full}</p>		<p>OWASP SCP 35, 90, 171-172</p> <p>OWASP ASVS 1.6.1, 1.6.2, 1.6.4, 2.10.4, 6.4.1, 6.4.2</p> <p>OWASP APPSENSOR -</p> <p>CAPEC 116</p> <p>SAFECODE 21, 29</p> <p>{Common_Title_full}</p>	
PRÁTICAS DE CRIPTOGRAFIA	Q	PRÁTICAS DE CRIPTOGRAFIA	K			

CORNUCOPIA	A			2	3
	Você inventou um novo ataque de qualquer tipo	(Nenhum Cartão)		Lee consegue ignorar os controles do aplicativo pois foram usadas funções arriscadas da linguagem de programação ao invés de opções seguras, ou há erros de conversão, ou porque o aplicativo está inseguro quando um recurso externo está indisponível, ou há race condition, ou há problemas na inicialização ou alocação de recursos, ou quando há sobrecarga	Andrew consegue acessar o código fonte, ou descompilar o aplicativo, ou consegue acessar a lógica do negócio para entender como a aplicação funciona e quais segredos ela contém
CORNUCOPIA	<p><i>Leia mais sobre segurança da aplicação nos guias da OWASP (Requirements, Development, Code Review and Testing) e na série OWASP Cheat Sheet, e no modelo de maturidade Open SAMM (Software Assurance Maturity Model)</i></p>			<p>OWASP SCP 194-202, 205-209</p> <p>OWASP ASVS 14.1.2</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 25-26, 29, 96, 123-124, 128-129, 264-265</p> <p>SAFECODE 3, 5-7, 9, 22, 25-26, 34</p> <p>{Common_Title_full}</p>	<p>OWASP SCP 134</p> <p>OWASP ASVS 14.1.1</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 189, 207</p> <p>SAFECODE</p> <p>-</p> <p>{Common_Title_full}</p>
	4	CORNUCOPIA	5	CORNUCOPIA	6
CORNUCOPIA	Keith consegue realizar uma ação e isto não é atribuído a ele		Larry consegue induzir a confiança de outras partes, incluindo usuários autenticados, ou violar esta confiança em outro lugar (ex: em outro aplicativo)	Aaron consegue ignorar os controles porque a manipulação de erros/exceções é perdida/ignorada, ou é implementada de forma inconsistente ou parcial, ou não há negação de acesso por padrão (ex: erros devem terminar o acesso/execução da funcionalidade), ou depende do tratamento por algum outro serviço ou sistema	As ações de Mwengu não podem ser investigadas porque não há um registro correto de eventos de segurança com precisão, ou não há uma trilha de auditoria completa, ou estas podem ser alteradas ou excluídas pelo Mwengu, ou não existe um serviço de registro centralizado
	<p>OWASP SCP 23, 32, 34, 42, 51, 181</p> <p>OWASP ASVS 7.2.1, 7.2.2</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC</p> <p>-</p> <p>SAFECODE</p> <p>-</p> <p>{Common_Title_full}</p>		<p>OWASP SCP</p> <p>-</p> <p>OWASP ASVS 1.9.2, 9.1.1, 5.1.5, 9.2.1, 9.2.4</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 89, 103, 181, 459</p> <p>SAFECODE</p> <p>-</p> <p>{Common_Title_full}</p>	<p>OWASP SCP 109-112, 155</p> <p>OWASP ASVS 4.1.5, 7.1.4</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 54, 98, 164</p> <p>SAFECODE 4, 11, 23</p> <p>{Common_Title_full}</p>	<p>OWASP SCP 113, 114, 115, 117, 118, 121-130</p> <p>OWASP ASVS 7.1.2, 7.1.4, 7.2.1, 7.2.2, 7.3.1, 7.3.3, 8.3.5, 9.2.5</p> <p>OWASP APPSENSOR</p> <p>-</p> <p>CAPEC 93</p> <p>SAFECODE 4</p> <p>{Common_Title_full}</p>

CORNUCOPIA	8	CORNUCOPIA	9	CORNUCOPIA	10	CORNUCOPIA	J
	<p>David consegue ignorar o aplicativo para obter acesso aos dados porque a infraestrutura de rede e servidores e os serviços suportados não foram configurados de forma segura, as configurações não são verificadas periodicamente e os patches de segurança não são aplicados, ou os dados armazenados localmente não são fisicamente protegidos</p> <hr/> OWASP SCP 151, 152, 156, 160, 161, 173-177		<p>Michael consegue ignorar o aplicativo para obter acesso aos dados porque ferramentas ou interfaces administrativas não estão adequadamente seguras</p> <hr/> OWASP SCP 23, 29, 56, 81, 82, 84-90		<p>Spyros consegue contornar os controles do aplicativo porque os códigos fontes tanto dos frameworks, como de bibliotecas e componentes utilizados contêm código malicioso ou vulnerabilidades</p> <hr/> OWASP SCP 57, 151-152, 204-205, 213-214		<p>Roman consegue explorar o aplicativo pois este foi compilado usando ferramentas desatualizadas ou configurações não seguras como padrão ou informações de segurança não foram documentadas e passadas para o time operacional</p> <hr/> OWASP SCP 90, 137, 148, 151-154, 175-179, 186, 192
CORNUCOPIA	Q	CORNUCOPIA	K	CURINGA	Joker	CURINGA	Joker
	<p>Jim pode realizar ações mal-intencionadas, não normais, sem detecção e resposta em tempo real pela aplicação</p> <hr/> OWASP SCP -		<p>Grant pode utilizar o aplicativo para negar o serviço a alguns ou a todos os usuários</p> <hr/> OWASP SCP 41, 55		<p>Alice consegue utilizar a aplicação para realizar ataques a dados e usuários do sistema</p> <hr/> <i>Você pensou em se tornar membro individual da OWASP? Todas as ferramentas, guias e reuniões locais são gratuitas para todos, mas ser um membro individual apoia o trabalho da OWASP</i>		<p>Bob pode influenciar, alterar ou mudar a aplicação para que ela não cumpra os propósitos legais, regulamentadores, contratuais ou outras diretrizes organizacionais</p> <hr/> <i>Examine as vulnerabilidades e descubra como elas podem ser solucionadas através do OWASP Juice Shop, Security Shepherd, ou usando o desafio online OWASP Hacking-lab. Ambos são gratuitos</i>

Cut  
here



## Change Log

Version / Date		Comments
0.1	30 Jul 2012	Original Draft
0.2	10 Aug 2012	Draft reviewed and updated
0.3	15 Aug 2012	Draft announced OWASP SCP mailing list for comment.
0.4	25 Feb 2013	Play rules updated based on feedback during workshops. Added reference to PCI SSC Information Supplement: PCI DSS E-commerce Guidelines. Descriptive text extended and updated. Added contributors section, page numbering, FAQs and change log.
1	25 Feb 2013	Release.
1.01	03 Jun 2013	Framework-specific card deck discussion added. Additional FAQs created. Descriptive text updated. New cover image, and previous cover image moved to back. Cut lines added. FAQs 5 and 6 added. Attack descriptions on cards with tinted backgrounds changed to black (from dark grey). Project contributors added.
1.02	14 Aug 2013	Warning about time to print added. Additional alternative game rules added (twenty-one, play a deck over a week, play full hand and then discuss). Compliance deck concept added. FAQs 5 and 6 added. Attack descriptions on cards with tinted backgrounds changed to black (from dark grey). Project contributors added.
1.03	18 Sep 2013	Minor attack wording changes on two cards. OWASP SCP and ASVS cross-references checked and updated. Code letters added for suits. All remaining attack descriptions on cards changed to black (from dark grey) and background colours amended to provide more contrast and increase readability.
1.04	01 Feb 2014	Text “password change, password change,” corrected to “password change, password recovery,” on Queen of Authentication card.
1.05	21 Mar 2014	Updates to alternative game rules. Additional FAQs created. Contributors updated. Podcast and video links added.
1.1	04 Mar 2015	Change log date corrected for v1.05. Cross-references updated for 2014 version of ASVS. Contributors updated. Minor text changes to cards to improve readability.
1.2	29 Jun 2016	Video mentioned/linked Separate score sheet mentioned/linked. Previous embedded score sheet pages deleted. Correction (identified by Tom Brennan) and addition to text on card 8 Authentication. Oana Cornea and other participants at the AppSec EU 2015 project summit added to list of contributors. Dario De Filippis added as project co-leader. Wiki Deck link added. Cross-references updated for ASVS v3.0.1 and CAPEC v2.8. Minor text changes to a small number of cards. Added “-EN” to version number in preparation for “-ES” version. Susana Romaniz added as a contributor to the Spanish translation. Minor text changes to instructions and FAQs.
1.3	01 Jan 2023	Cross-references updated from ASVS v3.0.1 to ASVS v4.0 by Johan Sydseter.

## Colaboradores do projeto

Todos os projetos OWASP dependem dos esforços voluntários de pessoas nos setores de desenvolvimento de software e segurança da informação.

Eles contribuíram com seu tempo e energia para fazer sugestões, fornecer feedback, escrever, revisar e editar documentação, encorajar, testar o jogo e promover o conceito.

Sem todos os seus esforços, o projeto não teria progredido até este ponto.

Por favor, entre em contato diretamente com a lista de discussão ou com os líderes do projeto, se alguém estiver faltando nas listas abaixo.

- Simon Bennetts
- Sebastien Gioria
- Mark Miller
- Tom Brennan
- Tobias Gondrom
- Cam Morris
- Fabio Cerullo
- Timo Goosen
- Susana Romaniz
- Oana Cornea
- Anthony Harrison
- Ravishankar Sahadevan
- Johanna Curiel
- John Herrlin
- Tao Sauvage
- Todd Dahl
- Jerry Hoff
- Stephen de Vries
- Luis Enriquez
- Marios Kourtesis
- Colin Watson
- Ken Ferris
- Antonis Manaras
- Johan Sydseter
- Dario De Filippis
- Jim Manico

- Os funcionários esforçados da OWASP.
- Participantes das reuniões dos capítulos da OWASP Londres, OWASP Manchester, OWASP Holanda e OWASP Escócia, e do encontro de Gamificação em Londres, que fizeram sugestões úteis e fizeram perguntas desafiadoras
- Blackfoot UK Limited para presentear arquivos de design prontos para impressão e centenas de baralhos de cartas impressos profissionalmente para distribuição por correio e nas reuniões do capítulo da OWASP
- OWASP NYC por criar um design de caixa OWASP e distribuir pacotes na AppSec USA 2014.

## Podcasts e vídeos

Os seguintes recursos de suporte do OWASP® Cornucopia estão disponíveis online:

- Vídeo - Usando os cartões, criado durante a cúpula do projeto AppSec EU 2015, 20 de maio de 2015  
<https://www.youtube.com/watch?v=i5Y0akWj31k>
- Entrevista em podcast, canal OWASP 24/7 Podcast, 21 de março de 2014  
<http://trustedsoftwarealliance.com/2014/03/21/the-owasp-cornucopia-project-with-colin-watson/>
- Vídeo de apresentação, OWASP EU Tour 2013 Londres, 3 de junho de 2013  
[https://www.youtube.com/watch?v=Q\\_LF-8xNXVk](https://www.youtube.com/watch?v=Q_LF-8xNXVk)



Consulte o site do projeto para mais informações e materiais de apresentação.