Uma Modelagem de Risco Centrada em Comportamentos para o Desenvolvimento Seguro de Serviços no Ecossistema Web

Carlo M. R. da Silva[1,2], **Vinícius C. Garcia**[1]

[1] CIn – Universidade Federal de Pernambuco (UFPE)

[2]Campus Garanhuns – Universidade de Pernambuco (UPE)

{cmrs,vcg}@cin.ufpe.br

Abstract. *The aim of this paper is to present a risk modeling for secure development over the Web ecosystem. The proposal also aims to estimate a factor of risk and impact for assets, considering data breaches, human aspects and service compliance, furthermore, considering the behaviors of actors, devices, and resources. In addition, the proposal presents evaluation through "top-threat" catalogs and test cases developed with Java language and BDD techniques. As a result, it is possible to observe applicability to the most emerging risks characterizing itself as an artifact that provides a guided development in the prevention of potential threats to services over the* Web.

Resumo. *O objetivo deste artigo é apresentar uma modelagem de risco para o desenvolvimento de serviços no ecossistema Web. A proposta visa estimar um fator de risco e impacto aos ativos considerando a violação de dados, os aspectos humanos e a conformidade do serviço. Além de considerar os comportamentos de seus atores, dispositivos e recursos. Adicionalmente, a proposta é validada através de catálogos de ataques* top-threats *disponíveis publicamente e testes estruturais com a linguagem Java e técnicas BDD. Com isso, é possível observar sua aplicabilidade sobre os riscos mais emergentes, caracterizando-se como um artefato que proporciona um desenvolvimento guiado na prevenção de ameaças para serviços da Web.*

1. Introdução

Atualmente a *Web* se tornou a plataforma mais empregada para distribuição de serviços nos mais variados segmentos como financeiro, social e até monitoramento de infraestruturas críticas. Em linhas gerais, pode-se afirmar que a *Web* é um ecossistema onde pessoas e empresas, através de dispositivos e serviços, trocam dados (sensíveis ou não) - considerados ativos de informação, que podem ou não conter ou represetar algum valor agregado para o seu proprietário.

Contudo, é cada vez maior o número de riscos[1] aos ativos neste ambiente em consequência de ataques à segurança. De acordo com a Symantec [Symantec 2019], o número de ataques à serviços na *Web* cresceu mais de 30% no ano de 2018. Pior, tais

[1]A norma ISO/IEC 27000:2013 define risco como a chance de uma ameaça se consolidar (probabilidade), resultando em um evento indesejado e possíveis conseqüências para um sistema ou para a organização.

ataques vêm demonstrando motivações variadas, desde ideologias a ganhos financeiros e passando por interesses políticos e espionagem industrial. A WhiteHat [WhiteHat 2016] estima que 50% dos sites possuem vulnerabilidades a serem exploradas.

Uma das formas de minimizar os danos dessas consequências é usando a Modelagem de Risco (*Risk Modeling* - RM), que se traduz como um modelo que auxilia na mitigação e mensuração dos riscos aos ativos de uma aplicação ou serviço, através de uma classificação bem definida [OWASP 2016]. Para tanto, é preciso considerar o ciclo de vida do serviço de ponto a ponto, desde o *datacenter* até o *endpoint*, visto que quando uma ameaça é concretizada em um determinado lado, a consequência pode ser desencadeada na outra extremidade do serviço [Shostack 2014]. Uma modelagem dessa natureza possibiita um maior entendimento dos riscos, considerando causas, consequências e probabilidades [ISO 2009, Gary Stoneburner and Feringa 2002].

Apesar de existirem várias propostas na literatura sobre RM [Shostack 2014, UcedaVelez and Morana 2015, Alvarez and Petrovic 2003], existem três problemáticas quanto ao uso desses artefatos. A primeira é sobre o **domínio de atuação** do serviço. De fato, é interessante considerar um certo nível de abstração na modelagem de riscos a fim de contemplar serviços em segmentos distintos. Contudo, em uma única ameaça existem muitos aspectos a serem considerados, como atores, dispositivos e funcionalidades envolvidas, o que sugere uma moderação na abstração.

O problema é que certas RM tem sua abordagem voltada para o propósito geral [Shostack 2014, UcedaVelez and Morana 2015, Dahl et al. 2007] com alta abstração e baixa profundidade, o que põe em cheque sua aplicabilidade sobre riscos intrínsecos da *Web*, já que seu nível de abstração dificulta a identificação de características como a semântica e a similaridades de ataques que partem de uma ameaça. As similaridades nas variações dos ataques podem ser observadas em ataques do tipo *Buffer Overflow* (BoF) [MITRE 2011], que atuam em diferentes estruturas de dados, mas compartilham o mesmo vetor de ataque e quando explorados reproduzem impactos similares nos ativos.

A segunda problemática diz respeito à **adesão aos modelos** em metodologias de desenvolvimento. As modelagens, em sua maioria, são direcionadas ao processo convencional [Shostack 2014, UcedaVelez and Morana 2015, Saitta et al. 2005] como o modelo em cascata, onde para avançar a uma nova atividades é necessário que a antecessora deva estar totalmente desenvolvida. Diante disso, a utilização da RM fica restrita à atividades como *Design* ou codificação, já que o desenvolvedor terá em mãos requisitos com especificações imutáveis, sugerindo um processo de mão única. Consequentemente, a RM não sugerirá práticas nas demais fases do desenvolvimento, apesar de também serem suscetíveis aos riscos, tornando sua eficiência questionável.

O problema ocorre também nas metodologias ágeis, devido a natureza pouco burocrática e enxuta dos artefatos dessas equipes [Stettina et al. 2012]. Além disso, neste tipo de desenvolvimento, as mudanças são esperadas a todo o momento, o que implica dizer que as fases antecessoras e sucessoras trabalham continuamente de forma iterativa e incremental [Sivanandan and B 2014], gerando resistência as modelagens disponíveis.

Por fim, a terceira problemática remete a **avaliação de impacto** sobre os ativos. É de conhecimento geral que o impacto provocado por um ataque em sistemas que manipulam dados comuns não pode ter um peso equivalente ao de um sistema com

dados sensíveis. Porém, o que chama atenção é que na literatura alguns modelos de mitigação de ameaça e risco tratam o ataque com um peso predefinido de forma generalizada [OWASP 2014, MITRE 2015], indiferente do domínio do serviço. Tal contexto deveria permitir que a equipe de desenvolvimento definisse a criticidade do ataque com base nos comportamentos do seu serviço e analisasse o prejuízo considerando não apenas a confidencialidade dos dados, mas também o fator humano privacidade e os aspectos técnicos e legais que possam trazer consequências ao serviço.

É neste contexto que este artigo tem por objetivo apresentar uma modelagem de risco (RM) à segurança de serviços projetados para o ecossistema Web. Diferente dos modelos convencionais, que são centrados no atacante, nos ativos ou no software, a proposta possui uma ótica hibrida, englobando essas três visões, resultando em um modelo centrado em comportamentos. Também tem o intuito de ser um modelo de documentação leve, para garantir maior aderencia no processo de desenvolvimento em equipes de natureza ágil. Por fim, por não ser de propósito geral, é capaz de considerar atores e recursos intrísecos ao ecossistema *Web*, além de fatores como engenharia social, propagação das ameaças, similaridades entre os ataques e impactos aos ativos.

2. Riscos no Ecossistema Web

Um ecossistema designa a observação do comportamento e interação entre indivíduos em um ambiente [DeRyck et al. 2013]. Moore et al. [Moore 1999] definem um ecossistema como uma comunidade com fins econômicos, apoiado por um conjunto de organizações e indivíduos que estão em constante interação. Na área de Computação, um Ecossistema de Software (ECOS) é definido como um conjunto de atores que funcionam como uma unidade e interagem com um mercado de software [Jansen et al. 2009].

No contexto da *Web*, um ecossistema é a interação entre o conjunto de serviços, dispositivos e recursos que estabelecem um protocolo de comunicação em comum como o HTTP e suas variantes[2] [Berners-Lee et al. 2001].

Na visão deste trabalho, um ecossistema *Web* é composto por:

- Usuário, que consome serviços, geralmente através de um navegador;
- Navegador, software capaz de oferecer a interação entre um usuário e uma aplicação *Web*. Uma vez que praticamente oferta tudo que o usuário precisa, é uma ferramenta com um grande número de responsabilidades;
- Administrador do Serviço, tem o papel de manter os ativos dos usuários, garantindo que os mesmos terão acesso restrito, íntegro e disponível sempre que o proprietário solicitar;
- Atacante, cujo o intuito é executar ações ilícitas sobre os ativos.

É neste cenário que surge o atacante com o intuito de executar ações ilícitas sobre os ativos. As motivações são variadas, como diversão, ideologias, ganhos financeiros ou interesses políticos. Em comparação ao usuário, seu fluxo é mais complexo, pois se foca em buscar comportamentos não esperados pelos envolvidos. Seu primeiro passo é buscar um meio de explorar vulnerabilidades no serviço, navegador Web ou no próprio usuário. Este método de exploração é denominado de vetor de ataque.

[2]Protocolos que derivam HTTP https://www.w3.org/Protocols/rfc2616/rfc2616.txt, como o HTTPS (SSL/TLS) https://tools.ietf.org/html/rfc2818, WS/WSS (WebSocket) https://tools.ietf.org/html/rfc6455, entre outros: https://www.w3.org/Protocols

Quando a vulnerabilidade está no serviço, há o pressuposto de que a responsabilidade com o ativo é totalmente direcionada aos seus administradores. E isso vai de encontro no dilema que é garantir que irão de fato adotar fortes políticas de segurança no desenvolvimento e na manutenção. Não obstante, o atacante pode explorar vulnerabilidades no navegador Web. Nesse contexto, o responsável pela segurança não é bem definido, pois se trata de um ambiente mais suscetível à falta de acuidade de seus usuários.

E, por fim, outra perspectiva é quando o atacante explora engenharia social sobre o usuário ou serviço. Ao contrário das vulnerabilidades anteriormente citadas, ela é direcionada ao descuido no fator humano, seja de forma direta, induzindo técnicas que interceptem o fluxo do usuário, ou indireta, em que o atacante analisa a vida pessoal de suas vítimas para explorar brechas nas políticas dos serviços.

2.1. Classificação dos Comportamentos de Risco

Embora possam existir semelhanças, os ataques no ecossistema **Web** podem comprometer variados atributos, causando impactos distintos nos ativos. Com base nessa consideração e visando auxiliar administradores de serviços *Web*, foi definida uma taxonomia (Figura 1) que divide o ecossistema em três domínios distintos (Serviço, Consumo do Serviço e Engenharia Social), onde se agrupam 8 ameaças que resultam em 20 vetores de ataques. Um documento [3] descreve maiores detalhes sobre a construção desse artefato.

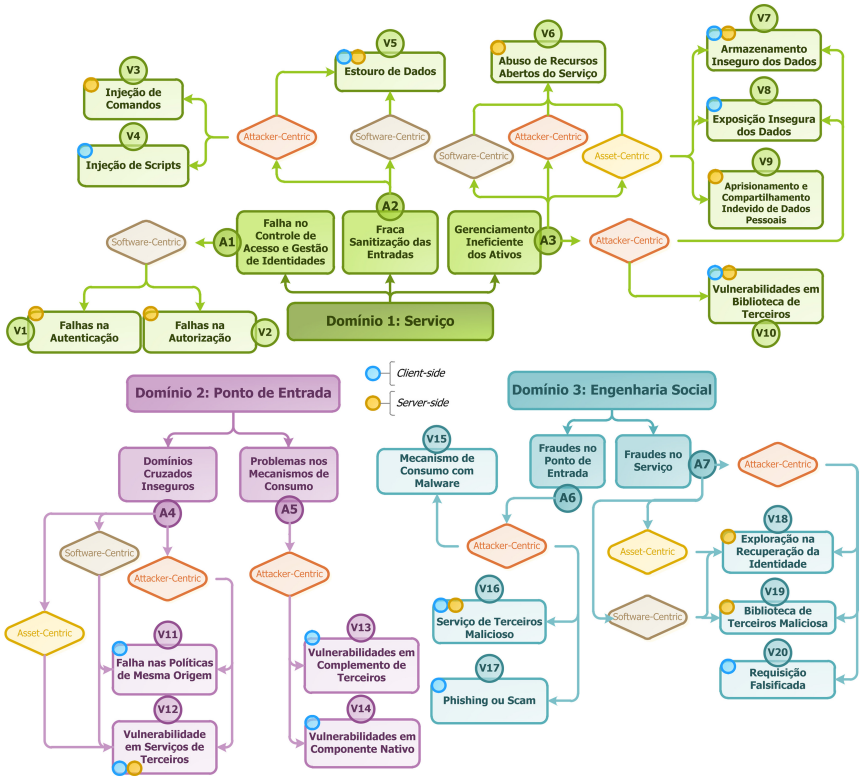

Figura 1. Taxonomia de ameaças do ecossistema Web.

[3] Documento sobre a taxonomia e ontologia de domínio: https://goo.gl/V4JF9x

3. Proposta

Essa seção descreve a metodologia adotada para o desenvolvimento da RM. Primeiramente, é apresentada a Figura 2 que ilustra a combinação dos artefatos com base nos objetivos da proposta. A metodologia da proposta é baseada no projeto da OWASP [OWASP 2016] que auxilia a construção de uma modelagem através de um processo guiado em 3 etapas, sendo que a construção da proposta baseia-se na etapa III, visto que as duas primeiras são focadas em outro artefato.

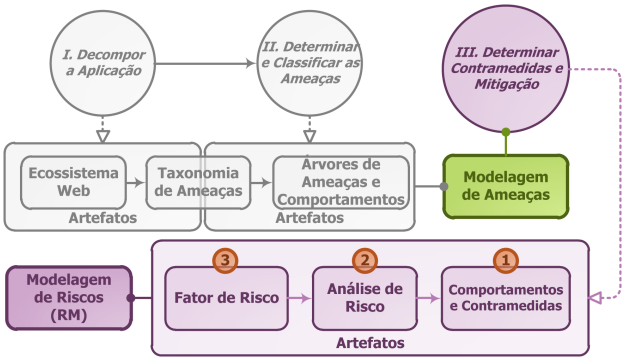

Figura 2. Etapas da metodologia para a construção da RM.

Importante frisar que o processo desse estudo, em comparação com a metodologia da OWASP, sofreu diversas adaptações. A RM proposta possui uma ótica baseada em comportamentos e tem o intuíto de contribuir no cerne do desenvolvimento ágil. Em contrapartida, a metodologia da OWASP é direcionada para o desenvolvimento estruturado, a exemplo do uso de Diagrama de Fluxo de Dados (DFD). Além disso, aspectos como definições, termos técnicos, nomenclaturas e axiomas foram extraídos de um Mapeamento Sistemático da Literatura (MSL)[4] com base em artigos da literatura.

3.1. Sobre as etapas I e II da Metodologia

Quanto ao ecossistema Web, um documento pode ser acessado para maiores detalhes, a exemplo dos atributos de segurança, a saber: Confidencialidade (Co), Integridade (In), Disponibilidade (Di), Privacidade (Pr), Não-Repúdio (Nr), Anonimato (An), Auditabilidade (Ad), Autenticidade (Au) e Responsabilidade (Re)[5], que caracteriza a primeira etapa, denominada "Decompor a Aplicação". A combinação das etapas I e II resulta na Modelagem de Ameaças (TM), esse modelo serve como premissa para a identificação e classificação das ameaças, sendo utilizado pela RM para associar os comportamentos com os riscos, ativos e contramedidas.

3.2. Etapa III: Comportamentos e Contramedidas

Um dos objetivos da RM é oferecer um nível de abstração que identifique similaridades e propagação entre os ataques. Para tanto, sua estrutura descreve a semântica de cada ataque através da granularidade nos vetores e na categorização das ameaças. A abordagem segue uma estrutura de árvores de ameaças e riscos aos ativos, como maneira formal e metódica de descrever o comportamento [Schneier 1999]. A fusão da árvore de ameaças resulta em um mapeamento entre comportamentos e vetores de ataques, conforme a Figura 3.

[4]Documento sobre o Mapeamento Sistemático: https://goo.gl/sfoT5G

[5]Documento sobre o ecossistema e os atributos de segurança: https://goo.gl/v7sIWW

Comportamentos	Prevenções e Contramedidas							Vetores	Contexto
C1. Armazenar Dados Sensíveis	Armazenar de Forma Segura Dados Sensíveis no Servidor	Backup e Redundância	CRM	Hardening (Server-side)	Técnicas para Registro de Evidências	-	-	V7, V9	Recursos Internos do Serviço
C2. Utilizar Bibliotecas de Terceiros	Armazenar de Forma Segura Dados Sensíveis no Servidor	Hardening (Client-side)	Hardening (Server-side)	-	-	-	-	V10, V19	
C3. Tratar Redirecionamentos e Parâmetros da Requisição	Hardening (Server-side)	Filtro Anti-Phishing	Tratamento dos Parâmetros da Requisição	Técnica para Registro de Evidências	-	-	-	V17, V20	
C4. Submeter Formulários	Armazenar de Forma Segura Dados Sensíveis no Cliente	Hardening (Client-side)	Hardening (Server-side)	Prevenção à Exposição de Dados Sensíveis no Cliente	Técnicas para Registro de Evidências	Tratamento das Entradas	-	V3, V4, V5	
C5. Gerir Acesso aos Recursos	Autorização Forte	Hardening (Client-side)	Hardening (Server-side)	Prevenção à Exposição de Dados Sensíveis no Servidor	Técnica para Registro de Evidências	-	-	V1, V2, V7, V18	
C6. Compartilhar Recursos de Forma Aberta	CORS	CRM	Hardening (Client-side)	Hardening (Server-side)	Prevenção à Exposição de Dados Sensíveis no Cliente	Prevenção à Exposição de Dados Sensíveis no Servidor	Técnicas para Registro de Evidências	V6	
C7. Gerir Identidade dos Usuários	Autenticação Forte	Hardening (Client-side)	Hardening (Server-side)	Técnica para Registro de Evidências	-	-	-	V8	
C8. Disponibilizar Mecanismos de Consumo	Acordo de Licença de Usuário Final (EULA)	-	-	-	-	-	-	V9	Mecanismos de Consumo
C9. Utilizar Mecanimos de Consumo	Armazenar de Forma Segura Dados Sensíveis no Cliente	Hardening (Client-side)	Prevenção à Exposição de Dados Sensíveis no Cliente	SOP	-	-	-	V11, V13, V14, V15	
C10. Conceder Gestão da Identidade	Hardening (Client-side)	Prevenção à Exposição de Dados Sensíveis no Cliente	Técnica para Registro de Evidências	-	-	-	-	V1, V2, V12, V18	Serviços de Terceiros
C11. Consumir Recursos Computacionais	Armazenar de Forma Segura Dados Sensíveis no Cliente	Hardening (Client-side)	Prevenção à Exposição de Dados Sensíveis no Cliente	Técnica para Registro de Evidências	-	-	-	V10, V12, V16	
C12. Realizar Transações Financeiras	Armazenar de Forma Segura Dados Sensíveis no Cliente	CRM	Hardening (Client-side)	Prevenção à Exposição de Dados Sensíveis no Cliente	Técnica para Registro de Evidências	-	-	V10, V16, V19	
C13. Compartilhar Dados Pessoais	Armazenar de Forma Segura Dados Sensíveis no Cliente	CRM	Prevenção à Exposição de Dados Sensíveis no Cliente	-	-	-	-	V7, V8, V9	Fator Humano
C14. Gerir o Relacionamento com o Cliente	Acordo de Nível de Serviço (SLA)	Termos de Serviço (ToS)	-	-	-	-	-	V15, V16, V17	
Nível de Codificação			*Nível de Consumo*			*Nível de Infraestrutura*			

Figura 3. Comportamentos considerando as prevenções e ameaças.

O benefício é representar um determinado comportamento ilustrado no nó-raiz, observar diferenças e similaridades através dos nós vizinhos e mensurar comportamento de diferentes maneiras até seus objetivos, indicados através dos nós abaixo, onde identifica os vetores de ataques e suas respectivas contramedidas. Ao todo o estudo levantou 14 comportamentos considerados macros, ou seja, são ações comumente existentes em serviços do ecossistema Web. Um documento disponível pode ser acessado para maiores detalhes sobre os comportamentos e contramedidas. [6]

3.3. Etapa III: Análise de Risco e Fator de Risco

E por fim, o próximo passo é mensurar os impactos aos ativos, relativo ao artefato (v). Isso se faz possível através da descrição dos atributos de segurança envolvidos, conforme ilustrado na Figura 4. A análise tem o intuito de estimar a criticidade de cada vulnerabilidade como forma de dar visibilidade aos administradores sobre os ataques que merecem maior atenção durante o desenvolvimento de seus serviços.

Matriz de Vetor x Impacto		Domínio do Serviço									
		A1. Falhas no Controle de Acesso e Gestão de Identidades		A2. Fraca Sanitização das Entradas			A3. Gerenciamento Ineficiente dos Ativos				
		V1. Falhas na Autenticação	V2. Falhas na Autorização	V3. Injeção de Comandos	V4. Injeção de Scripts	V5. Estouro de Dados	V6. Abuso de Recursos Abertos do Serviço	V7. Armazenamento Inseguro dos Dados	V8. Exposição Insegura dos Dados	V9. Aprisionamento e Compartilhamento Indevido de Dados Pessoais	V10. Vulnerabilidades em Biblioteca de Terceiros
Impacto nos Dados	Confidencialidade	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
	Integridade	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓
	Disponibilidade	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓
Impacto na Identidade do Usuário	Anonimato	✓	✗	✓	✓	✓	✗	✗	✗	✗	✓
	Privacidade	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
	Não-Repúdio	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓
Impacto na Conformidade do Serviço	Auditabilidade	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓
	Autenticidade	✓	✗	✓	✗	✓	✗	✗	✗	✗	✓
	Responsabilidade	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Peso do Vetor		7	5	9	6	9	2	4	4	3	9

Matriz de Vetor x Impacto		Domínio do Ponto de Entrada				Engenharia Social					
		A4. Domínios Cruzados Inseguros		A5. Problemas nos Mecanismos de Consumo		A6. Fraudes no Endpoint			A7. Fraudes no Serviço		
		V11. Falhas nas Políticas de Mesma Origem	V12. Vulnerabilidades em Serviços de Terceiros	V13. Vulnerabilidades em Complemento de Terceiros	V14. Vulnerabilidades em Componente Nativo	V15. Mecanismo de Consumo com Malware	V16. Serviços de Terceiros Maliciosos	V17. Phishing ou Scam	V18. Exploração na Recuperação da Identidade	V19. Biblioteca de Terceiros Maliciosa	V20. Requisição Falsificada
Impacto nos Dados	Confidencialidade	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Integridade	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
	Disponibilidade	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓
Impacto na Identidade do Usuário	Anonimato	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗
	Privacidade	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Não-Repúdio	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Impacto na Conformidade do Serviço	Auditabilidade	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
	Autenticidade	✗	✓	✗	✗	✗	✗	✗	✓	✓	✓
	Responsabilidade	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Peso do Vetor		4	7	4	4	5	6	4	6	9	6

Figura 4. Vetores de Ataques considerando os atributos de segurança.

[6] Documento sobre Comportamentos e Contramedidas https://goo.gl/9pt0LB

É importante salientar que o valor agregado de cada ativo varia bastante, pois cada modelo de negócio tem suas particularidades. Portanto, o atenuante sugerido é uma combinação que considera os aspectos utilizados no método de exploração e o comprometimento dos atributos de segurança relacionados.

A análise de risco e impacto é dividida em três vertentes: correção, destruição e prevenção. A correção representa uma Análise Qualitativa (a_1) que mensura o Poder de Propagação (*PP*) e a Dificuldade de Recuperação (*DR*) do ataque. Já a destruição se enviesa em uma Análise Quantitativa (a_2) com base no peso do vetor, ou seja, a soma dos atributos de segurança violados pelo respectivo vetor, conforme descrito na Figura 4.

E por fim, a prevenção está relacionada à Análise de Probabilidade (a_3) do ataque, que é traduzida pela Facilidade de Execução (*FE*) do ataque por parte do atacante contra a dificuldade de detecção (*DD*) do ataque pelos administradores dos ativos. O resultado de (*R*) é determinado pela raiz quadrada da média aritmética das análises a_1, a_2 e a_3. A justificativa desta raiz quadrada é garantir que o resultado final representa um valor entre as escalas para o fator *R* descrito na Tabela 1.

O cálculo do fator de risco (*R*) de cada ataque é simplificado na equação (1) e detalhado na equação (2). O seu resultado irá constar em uma escala de 1 a 3, sendo 1 para ataques de menor criticidade e 3, de maior impacto.

$$\begin{aligned} R &= \sqrt{a_n} \\ R &= \sqrt{\frac{a_1 + a_2 + a_3}{3}} \\ a_1 &= PP \times DR \\ a_2 &= Co + In + Di + Pr + Nr + An + Ad + Au + Re \\ a_3 &= FE \times DD \end{aligned} \tag{1}$$

$$R = \sqrt{\frac{(PP \times DR) + (Co + In + Di + Pr + Nr + An + Ad + Au + Re) + (FE \times DD)}{3}} \tag{2}$$

Conforme listado na Tabela 1, para todos os índices, com exceção do *R* e a_2, seus fatores variam entre: Baixo, Médio e Alto. Já o a_2 é um conjunto de 9 atributos de segurança que recebem a seguinte regra: valor 0 para não violado e 1 para violado. Obviamente ao menos um atributo de segurança deve estar envolvido, e o resultado de a_2 será a soma de todos os violados. Já o *R* possui uma classificação com maior sensibilidade, variando de: Muito Baixo, Baixo, Médio, Alto a Muito Alto.

4. Validação

Como medida avaliativa da proposta, decidimos utilizar a RM para estabelecer classificações facetadas de diversos ataques publicados na literatura, identificando os atributos de segurança, conforme a Seção 2, violados ou não por um respectivo ataque. Esse tipo de classificação tem o intuito de identificar similaridades entre várias categorias de um determinado assunto, possibilitando dessa forma uma classificação em tabelas de acordo com suas características e comportamentos.

Tabela 1. Escala do nível de cada fator

Escala para os fatores *PP*, *DR*, *FE* e *DD*					
Nível	**Valor**	**Nível**	**Valor**	**Nível**	**Valor**
Baixo	1	Médio	2	Alto	3
Escala para o fator *R*					
Nível	**Valor**	**Nível**	**Valor**	**Nível**	**Valor**
Muito Baixo	De 0,01 até 1,00	Baixo	De 1,01 até 1,50	Médio	De 1,51 até 2,00
Alto	De 2,01 até 2,50	Muito Alto	De 2,51 até 3,00	-	-

Como ponto de partida, foi necessário realizar um levantamento de catálogos de ataques registrados na literatura. Nesta linha, catálogos do tipo *top threats* são geralmente os mais recomendados por apresentarem os ataques emergentes.

O segundo critério foi identificar a presença dos parâmetros necessários para a análise de risco com a RM. Considerando os critérios citados, foi decidido submeter a proposta com base em catálogos como o *OWASP Top Ten*[OWASP 2013] e o *CWE/SANS Top 25 Most Dangerous Software Errors*[MITRE 2011], estes por serem mais consolidado na comunidade de desenvolvimento seguro. O resultado dessa validação pode ser conferido nas Figura 5 e 6.

OWASP Top Ten 2013																		
Classificação			a_1		a_2									a_3			R	
Ataque	Vetor	Propagação	PP	DR	Co	In	Av	Pr	Nr	An	Ad	Au	Re	FE	DD	R		
A1 Injection	V3. Injeção de Comandos	Server-side	3	3	✓	✓	✓	✓	✓	✓	✓	✓	✓	2	1	2,58		
A2 Broken Authentication and Session Management	V1. Falhas na Autenticação	Server-side	2	2	✓	✓	✓	✓	✓	✓	✗	✓	✓	3	2	2,45		
A3 Cross-Site Scripting (XSS)	V4. Injeção de Scripts	Client-Side	3	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	1	2,16		
A4 Insecure Direct Object References	V8. Exposição Insegura dos Dados	Server-side / Client-side	2	2	✓	✓	✓	✓	✗	✓	✗	✓	✓	3	1	2,16		
A5 Security Misconfiguration	V6. Abuso de Recursos do Serviço	Server-side	2	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	1	2,00		
A6 Sensitive Data Exposure	V8. Exposição Insegura dos Dados	Client-Side	3	3	✓	✓	✓	✓	✗	✗	✗	✗	✓	1	2	2,31		
A7 Missing Function Level Access Control	V2. Falhas na Autorização	Server-side	2	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	2	2,24		
A8 Cross-Site Request Forgery (CSRF)	V20. Requisição Falsificada	Client-Side	2	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	2	1	1,91		
A9 Using Components with Known Vulnerabilities	V6. Abuso de Recursos do Serviço	Server-side	3	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	2	3	2,38		
A10 Unvalidated Redirects and Forwards	V17. Phishing ou Scam	Client-Side	1	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	2	1	1,73		

Figura 5. Faceta *OWASP Top Ten 2017*.

O principal diferencial da RM proposta em relação às demais publicadas na literatura é que ela é centrada em vetores de propagação do ataque, ou seja, não está focada nos aspectos que originaram a vulnerabilidade, mas, sim, no método de exploração.

5. Trabalhos Relacionados

Nesta seção, serão descritos trabalhos da literatura que possuem soluções correlatas à proposta deste estudo. Uma das motivações para o desenvolvimento da RM foi minimizar a carência de soluções dessa natureza considerando a Web como um ecossistema. Além disso, é pretendido considerar três lacunas, conforme citado na Seção 1: (i) **domínio de atuação**, (ii) **adesão aos modelos** e (iii) **avaliação de impacto**.

Em (i), devido a proposta se limitar ao ecossistema Web, é possível estabelecer a relação entre dispositivos, atores e domínios envolvidos. Com isso, é possível cobrir ameaças que vão além do domínio da aplicação, a exemplo dos mecanismos de consumo do domínio 2. Não obstante, o *Common Attack Pattern Enumeration and Classification*

CWE/SANS Top 25 Most Dangerous Softwares Errors																			
Classificação			a_1		a_2										a_3		R
Ataque	Vetor	Propagação	PP	DR	Co	In	Av	Pr	Nr	An	Ad	Au	Re	FE	DD	R
CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	V3. Injeção de Comandos	Server-side	3	3	✓	✓	✓	✓	✓	✓	✓	✓	✓	3	3	3,00
CWE-78 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	V3. Injeção de Comandos	Server-side	3	3	✓	✓	✓	✓	✓	✓	✓	✓	✓	3	1	2,65
CWE-120 Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')	V6. Abuso de Recursos do Serviço	Server-side	3	2	✓	✗	✓	✗	✗	✗	✗	✗	✓	3	1	2,00
CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	V4. Injeção de Scripts	Client-Side	3	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	1	2,16
CWE-306 Missing Authentication for Critical Function	V1. Falhas na Autenticação	Server-side	2	2	✓	✓	✓	✓	✓	✓	✗	✓	✓	3	2	2,45
CWE-862 Missing Authorization	V2. Falhas na Autorização	Server-side	3	3	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	2	2,58
CWE-798 Use of Hard-coded Credentials	V1. Falhas na Autenticação	Server-side	1	3	✓	✓	✓	✓	✓	✓	✗	✓	✓	3	2	2,38
CWE-311 Missing Encryption of Sensitive Data	Exposição Insegura dos Dados	Client-Side	2	3	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	1	2,16
CWE-434 Unrestricted Upload of File with Dangerous Type	V6. Abuso de Recursos do Serviço	Server-side	2	3	✓	✓	✓	✓	✓	✓	✓	✓	✓	2	2	2,52
CWE-807 Reliance on Untrusted Inputs in a Security Decision	V2. Falhas na Autorização	Server-side	3	3	✓	✓	✓	✓	✓	✗	✗	✓	✓	3	2	2,71
CWE-250 Execution with Unnecessary Privileges	V2. Falhas na Autorização	Server-side	2	3	✓	✓	✓	✓	✗	✗	✗	✓	✓	3	2	2,45
CWE-352 Cross-Site Request Forgery (CSRF)	V20. Requisições Falsificadas	Client-Side	2	3	✓	✓	✓	✓	✗	✗	✗	✗	✓	2	2	2,24
CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	V2. Falhas na Autorização	Server-side	3	3	✓	✓	✓	✓	✓	✓	✓	✓	✓	3	1	2,65
CWE-494 Download of Code Without Integrity Check	V8. Exposição Insegura dos Dados	Client-Side	1	2	✗	✓	✗	✗	✗	✗	✗	✗	✓	2	2	1,63
CWE-863 Incorrect Authorization	V2. Falhas na Autorização	Server-side	3	3	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	2	2,58
CWE-829 Inclusion of Functionality from Untrusted Control Sphere	V6. Abuso de Recursos do Serviço	Server-side	3	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	2	2,38
CWE-732 Incorrect Permission Assignment for Critical Resource	V2. Falhas na Autorização	Server-side	3	3	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	1	2,38
CWE-676 Use of Potentially Dangerous Function	V6. Abuso de Recursos do Serviço	Server-side	1	3	✓	✓	✓	✓	✗	✗	✗	✗	✓	3	1	1,91
CWE-327 Use of a Broken or Risky Cryptographic Algorithm	V7. Armazenamento Inseguro dos Dados	Server-side	1	3	✓	✓	✗	✓	✗	✗	✗	✗	✓	2	2	1,91
CWE-131 Incorrect Calculation of Buffer Size	V6. Abuso de Recursos do Serviço	Server-side	3	2	✗	✗	✓	✗	✗	✗	✗	✗	✓	3	1	1,91
CWE-307 Improper Restriction of Excessive Authentication Attempts	V6. Abuso de Recursos do Serviço	Server-side	2	2	✓	✓	✓	✓	✓	✓	✗	✗	✓	2	2	2,24
CWE-601 URL Redirection to Untrusted Site ('Open Redirect')	V17. Phishing ou Scam	Client-Side	2	3	✓	✓	✓	✓	✗	✗	✗	✗	✓	2	1	2,08
CWE-134 Uncontrolled Format String	V3. Injeção de Comandos	Server-side	2	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	2	2	2,08
CWE-190 Integer Overflow or Wraparound	V6. Abuso de Recursos do Serviço	Server-side	2	2	✓	✓	✓	✓	✗	✗	✗	✗	✓	2	1	1,91
CWE-759 Use of a One-Way Hash without a Salt	V7. Armazenamento Inseguro dos Dados	Server-side	1	2	✓	✓	✗	✓	✗	✗	✗	✗	✓	3	2	2,00

Figura 6. Faceta *CWE/SANS Top 25 Most Dangerous Software Errors*.

(CAPEC) [MITRE 2015] é um modelo de classificação projetado para software em geral. Embora houvesse um grande número de ataques, o modelo se classifica para propósito geral, o que induz alta abstração e baixa granularidade em determinadas variações de ataques, a exemplo das variações de “Requisições Falsificadas”.

Em contraste ao CAPEC, o projeto *Web Application Security Consortium* (WASC) [WebAppSec 2010] é específico para Web. No entanto, possui um alto nível de abstração das semelhanças de um ataque correspondente, uma vez que não é incomum um grande número de ameaças compartilhar uma mesma classificação, a exemplo do “Abuso de Recursos do Serviço”, resultando assim em uma visão macro para o administrador do serviço que carece de um modelo mais baixo nível.

Em (ii) descreve a adoção de modelagens de risco no desenvolvimento de *software*. Alguns trabalhos trazem apoio ao desenvolvedor, descrevendo o como, quando e onde uma ameaça é atuante [Landwehr et al. 1994] ou buscam investigar implementações inseguras baseando-se em padrões de falhas na codificação [Alvarez and Petrovic 2003, Tsipenyuk et al. 2005] ou condicionada a perspectiva do atacante [Douad and Dahmani 2015]. Contudo, apesar de cobrir a propagação, essas abordagens não relacionam similaridades compartilhadas entre ataques distintos. O principal benefício disso seria auxiliar na prevenção, pois facilita na distribuição das responsabili-

dades em identificar diversos vetores oriundos de uma única ameaça.

RM é um tópico bem debatido na literatura, a exemplo da modelagem DREAD [Shostack 2014], que estima o risco fazendo perguntas em cada uma das categorias, com base na ótica do atacante. Contudo, o DREAD, no molde inicialmente proposto, em 2010 foi desaconselhado por seus autores [SDL 2010a] devido falta de precisão em determinadas situações. Contudo, o mesmo recebeu melhorias quanto ao nível de riscos com base na causa e efeito [SDL 2010b]. Outro exemplo é a modelagem PASTA [UcedaVelez and Morana 2015] que é focada em ativos e provê um processo que considera possíveis cenários de ataque e vulnerabilidades e ataques, mitigando níveis de risco e impacto. Contudo, ambas abordagens são direcionadas ao desenvolvimento convencional, sem uma abordagem com maior consonância ao desenvolvimento ágil.

Por fim, em (iii) relata a capacidade da RM analisar riscos e impactos aos ativos. Alguns trabalhos propõem soluções baseadas em cenários de codificação de softwares [Tsipenyuk et al. 2005] ou aspectos entre os computadores e a rede [Hansman and Hunt 2005]. O principal diferencial da proposta é que sugerimos maior granularidade no modelo de classificação, através da ramificação de 20 vetores. Por estabelecer um agrupamento semântico, a proposta oferece maior precisão sobre a propagação da exploração e a identificação dos atributos de segurança comprometidos, mensurando o impacto aos ativos de forma qualitativa e quantitativa.

A RM apresentada nesse artigo propõe uma abordagem diferenciada, centrada em comportamentos e respectivos vetores de ataques. Com isso, visa delegar responsabilidades mútuas e distintas de acordo com a ameaça envolvida. Além disso, é preciso considerar que nem todas as prevenções estabelecidas em resposta aos incidentes serão bem sucedidas, ou seja, determinados riscos devem ser considerados. Diante disso, é interessante predefinir boas práticas de conduta para o caso de uma ameaça ser concretizada, visando minimizar o prejuízo resultante. Portanto, a RM proposta apresenta contramedidas em consonância com comportamentos e seus riscos.

6. Discussão e Considerações

Nesta seção, serão descritas algumas limitações que precisam ser considerada no estado atual do nosso estudo. A proposta visa preencher as seguintes lacunas: domínio de atuação (i), adesão dos modelos (ii) e avaliação de impacto (iii), descritos na Seção 1.

6.1. Domínio de Atuação

Quanto ao (i), é possível observar uma intersecção de domínios que rodeiam o cerne do ecossistema Web. Dentre os atores, recursos e dispositivos envolvidos, podemos considerar que a TM faz cobertura sobre serviços como rede sociais, transações financeiras, aplicações corporativas, *e-commerce* e *Web clients*.

Em contrapartida, demais serviços se enviesam em um conceito ainda mais amplo e por muitas vezes complexo, já que, apesar de certas tarefas utilizarem o protocolo HTTP, há de se considerar que outras não se limitam ao contexto da Web. Diante disso, a proposta em seu estado atual não deve ser considerada como um modelo autossuficiente para cobrir todos os riscos em certos domínios. Contudo, vislumbramos nestes cenários oportunidades para trabalhos futuros e evolução da proposta. Salientamos os cenários:

Recursos Computacionais: Armazenamento, plataformas de desenvolvimento e software sob demanda são alguns dos muitos recursos computacionais atualmente fornecidos como serviço. A computação em nuvem (CN) oferece esses recursos de forma utilitária, e muitos destes são distribuídos através da Web, sendo certos aspectos mencionados na proposta. Contudo, é preciso salientar que a CN é um paradigma com suas particularidades, em que aspectos como a virtualização e a heterogeneidade, exploram preocupações intrínsecas que não são abordadas na proposta.

Web das Coisas: Na mesma linha, também devemos analisar o contexto como um todo da Internet das Coisas (IoT). O termo "Web das coisas" remete a interface que possibilita as "coisas", dispositivos físicos, interagirem com a Web, mas é importante salientar que nem sempre estes dispositivos tem a Web como único canal de comunicação. Em certos casos esses dispositivos possuem protocolos e interfaces específicas do domínio da IoT, a exemplo do RFID, os quais não são cobertos pelo estudo.

No caso dos dispositivos *smart*, como *smartphones*, as contribuições dessa proposta são mais favoráveis, uma vez que esses utilizam a Web em diversas de suas funcionalidades, a exemplo de seu navegador nativo ou aplicativos que utilizam componentes como o *WebView*[7], que sugerem preocupações como o vetor "Injeção de Scripts".

Monitoramento e Controle Industrial: Outro contexto favorável da IoT é sobre os sistemas SCADA (*Supervisory Control and Data Acquisition*) que se definem como um software para o monitoramento e supervisão de dispositivos com atividades específicas. Eles exercem um papel essencial no controle de infraestruturas de grande porte, como usinas hidroelétricas, termoelétricas, estações para saneamento, entre outros.

Uma vez que precisam ser interoperáveis e façam uso da Web como canal de comunicação, muitos aspectos são cobertos pelo estudo. Não são incomuns os casos de mal-intencionados utilizarem ferramentas como a SHODAN[8] no intuito de obter controle destes dispositivos [Patton et al. 2014].

Malware: A RM contempla riscos e vetores desta natureza que são atuantes na Web, distinguindo suas propagações entre o contexto *server-side* e *client-side*. Todavia, devido à complexidade da ameaça e, condicionado ao cenário aplicado, possa ser necessário analisar certas funcionalidades de *malware* que vão além do escopo do ecossistema Web. A diversidade deste viés pode justificar um maior número de ramificações de seus vetores, distribuindo as preocupações em maior granularidade.

6.2. Adesão dos Modelos

Algumas equipes de desenvolvimento, em especial as de natureza ágil por fazer uso de artefatos mais sucintos, associam toda a responsabilidade do levantamento de riscos e impactos para uma equipes de auditoria externa. Esse tipo de equipe é responsavel em analisar respostas a incidentes, além de estimar consequência das possíveis violações nos recursos do serviço [ISO 2013].

Apesar de ter seus benefícios, como uma visão ampla e externa do serviço, é preciso considerar certas limitações. Para uma eficiente análise do controle dos recursos, é necessário um representante do domínio interno como parte interessada no acordo com

[7] *Android WebView*: http://goo.gl/Qt76Pz

[8] SHODAN, the search engine for Internet-connected devices: https://www.shodan.io/

as práticas exercidas pela equipe de auditoria externa. Inevitavelmente, avaliar tais atividades requer um documento como abordagem formal sobre os recursos envolvidos e respectivas mitigações dos possíveis riscos.

De toda forma, uma auditoria externa é uma solução que, por ser independente, sua aplicabilidade tem baixo impacto já que sua implantação pode ser realizada a qualquer momento por um tercerizado, não sendo necessário participar do ciclo de desenvolvimento e, por fortalecer o ambiente, sua utilização deve ser considerada. Contudo, seu uso não justifica negligenciar a adoção de uma RM como modelo formal, pois nada impede que ambas soluções sejam complementares, fortalecendo a gestão de riscos.

6.3. Avaliação de Impacto

Por fim, em (ii) reflete no comprometimento aos ativos, porém a questão de medição do valor de cada ativo é um tanto complexa devido a tolerância à falhas em determinados domínios. Em certos cenários o valor vai de acordo com a ótica do negócio, como em sistemas críticos, onde o peso é melhor definido. Contudo, serviços de propósito geral apresentam maior desafio em predefinir o valor de suas informações, pois em muitos casos vai de acordo com a ótica de quem usa [Bishop 2009]. Nesses casos, é importante oferecer a equipe de desenvolvimento um artefato que possibilite uma visão sobre os atributos de segurança que estariam potencialmente em risco quando uma determinada ameaça for concretizada [ISO 2013].

7. Conclusões e Trabalhos Futuros

Esse trabalho sugere que na ótica de uma modelagem de riscos, o benefício se enviesa em dois aspectos: na prevenção dos riscos, pois produz apoio na identificação, classificação e entendimento de vulnerabilidades, uma vez que sua abordagem define bem os comportamentos das ameaças existentes, e também oferece resposta a incidentes e contramedidas.

Uma vez que o modelo analisa similaridades, ela também minimiza incidentes provocados por ataques ainda não identificados, e.g. ataques *zero-day*. Consequentemente, ela estabelece uma padronização, fazendo com que sua adoção auxilie os envolvidos no entendimento dos riscos de domínio do software. Aplicamos a proposta em catálogos publicados em veículos que são aderentes à problemática como forma de validá-la. O objetivo foi demonstrar que a modelagem propõe uma abordagem direcionada na exploração dos métodos de execução dos ataques, estabelecendo regras de classificação que refletem ao cenário real.

Como trabalho futuro, além dos cenários descritos na Seção 6.1, temos a intenção de desenvolver uma (*Threat Modeling* - TM), combinada com a RM apresentada, para servir de apoio na elaboração de uma Arquitetura de Referência (*Reference Architecture* - RA), denominada *Security & Safety Driven Development* (S2D2), para o desenvolvimento de serviços *RESTful* alinhado à processos ágeis que envolvem comportamentos, como *Behavior-Driven Development (BDD)*; e também guiada à testes, como o *Test-Driven Development* (TDD), conforme ilustrado na Figura 7.

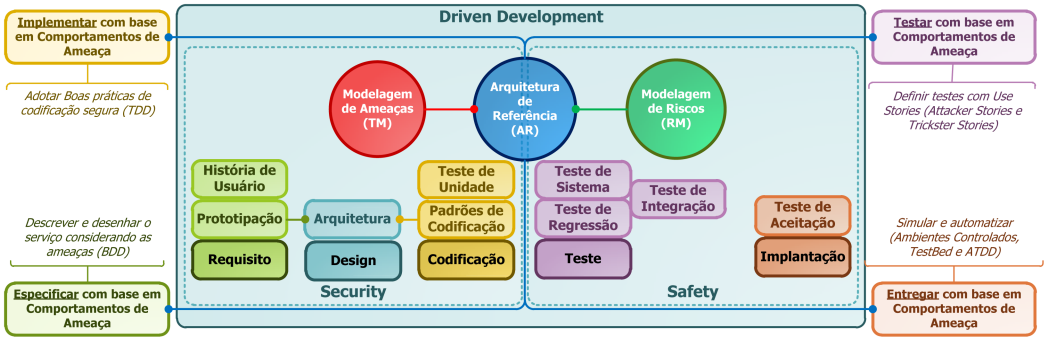

Figura 7. Arquitetura de Referência S2D2

O BDD e TDD garantem a compreensão dos testes de forma ubíqua junto a equipe de desenvolvimento. Uma vez que o TDD tem sua ótica voltada ao código-fonte, acaba relacionando o desenvolvedor e testador. O BDD abrange esse entendimento para os demais envolvidos, inclusive o próprio cliente do *software* em questão, pois associa os testes com uma documentação de maior abstração e não intrínseca ao código-fonte [Sivanandan and B 2014]. A ideia de combinar a TM, RM e RA é prover a equipe de desenvolvimento uma gestão de vulnerabilidades guiada para as prevenções das ameaças (*security*) e resposta a incidentes (*safety*) para um desenvolvimento seguro. A proposta também visa artefatos flexíveis, para maior congruência com o desenvolvimento ágil.

Referências

Alvarez, G. and Petrovic, S. (2003). A new taxonomy of web attacks suitable for efficient encoding. *Computers & Security*, 22(5):435–449.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

Bishop, M. (2009). Some ”secure programming”exercises for an introductory programming class. In *IEEE Security and Privacy*, pages 226–232.

Dahl, H. E. I., Hogganvik, I., and Stølen, K. (2007). Structured semantics for the coras security risk modelling language. *Cooperative and trusted systems*, SINTEF.

DeRyck, P., Desmet, L., Joosen, W., and Muhlberg, J. (2013). Web-platform security guide: Security assessment of the web ecosystem. Technical report, W3.

Douad, M. A. and Dahmani, Y. (2015). Artt taxonomy and cyber-attack framewok. In *New Technologies of Information and Communication*.

Gary Stoneburner, A. G. and Feringa, A. (2002). Risk management guide for information technology systems. *Disponível em: https://goo.gl/kB6yv5*.

Hansman, S. and Hunt, R. (2005). A taxonomy of network and computer attacks. *Computers & Security*, 24(1):31–43.

ISO (2009). Iso/iec 31010 risk management - risk assessment techniques. *Disponível em: https://www.iso.org/standard/51073.html*.

ISO (2013). Iso27001: Information technology - security techniques - information security management systems – requirements. *Disponível em: https://www.iso.org/standard/54534.html*.

Jansen, S., Finkelstein, A., and Brinkkemper, S. (2009). A sense of community: A research agenda for software ecosystems. In *2009 31st International Conference on Software Engineering - Companion Volume*, pages 187–190.

Landwehr, C. E., Bull, A. R., Mcdermott, J. P., William, and Choi, S. (1994). A taxonomy of computer program security flaws. *ACM Computing Surveys*, 26:211–254.

MITRE (2011). Cwe/sans top 25 most dangerous software errors. *Disponível em: http://cwe.mitre.org/top25/*.

MITRE (2015). Common attack pattern enumeration and classification (capec). *Disponível em: https://capec.mitre.org/*.

Moore, J. F. (1999). Creating value in the network economy. In Tapscott, D., editor, *Predators and Prey: A New Ecology of Competition*, pages 121–141, Boston, MA, USA. Harvard Business School Press.

OWASP (2013). Top ten 2013. *Disponível em: https://goo.gl/VKz94B*.

OWASP (2014). Vulnerabilities. *Disponível em: https://goo.gl/xsxX8G*.

OWASP (2016). Threat modeling cheat sheet. *Disponível em: https://goo.gl/tgn772*.

Patton, M., Gross, E., Chinn, R., Forbis, S., Walker, L., and Chen, H. (2014). Uninvited connections: A study of vulnerable devices on the internet of things (iot). In *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*, pages 232–235.

Saitta, P., Larcom, B., and Eddington, M. (2005). Trike v.1. *http://octotrike.org/*.

Schneier (1999). Attack trees. *Dr Dobb's Journal, v.24, n.12. Retrieved 2007-08-16*.

SDL, M. (2010a). Appendix n: Sdl security bug bar (sample). *Disponível em: https://goo.gl/USXuBM*.

SDL, M. (2010b). Security briefs - add a security bug bar to microsoft team foundation server 2010. *Disponível em: https://goo.gl/Qv3smB*.

Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley, 1 edition.

Sivanandan, S. and B, Y. C. (2014). Agile development cycle: Approach to design an effective model based testing with behaviour driven automation framework. In *Advanced Computing and Communications*, pages 22–25.

Stettina, C. J., Heijstek, W., and Fægri, T. E. (2012). Documentation work in agile teams: The role of documentation formalism in achieving a sustainable practice. In *Agile Conference (AGILE), 2012*, pages 31–40.

Symantec (2019). Internet security threat report. *Disponível em: https://www.symantec.com/security-center/threat-report*.

Tsipenyuk, K., Chess, B., and McGraw, G. (2005). Seven pernicious kingdoms: A taxonomy of software security errors. *IEEE Security & Privacy*, 3(6):81–84.

UcedaVelez, T. and Morana, M. (2015). *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. Wiley, 1 edition.

WebAppSec (2010). Wasc. *Disponível em: http://bit.ly/296FhpO*.

WhiteHat (2016). Security predictions 2017. *Disponível em: https://goo.gl/f94Qq8*.