UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

LUCIANO GONÇALVES DE CARVALHO

Requisitos e testes de segurança para brinquedos inteligentes

São Paulo
2017

LUCIANO GONÇALVES DE CARVALHO

Requisitos e testes de segurança para brinquedos inteligentes

Versão original

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de Concentração: Metodologia e Técnicas da Computação

Orientador: Prof. Dr. Marcelo Medeiros Eler

São Paulo
2017

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)

Carvalho, Luciano Gonçalves de
Requisitos e testes de segurança para brinquedos inteligentes /
Luciano Gonçalves de Carvalho ; orientador, Marcelo Medeiros Eler,
2017.

71 f. : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-
Graduação em Sistemas de Informação, Escola de Artes,
Ciências e Humanidades, Universidade de São Paulo.
Versão original

1. Segurança de computadores. 2. Segurança de software. 3.
Brinquedos inteligentes. 4. Internet das coisas. I. Eler, Marcelo
Medeiros, orient. II. Título.

CDD 22.ed.— 005.8

Dissertação de autoria de Luciano Gonçalves de Carvalho, sob o título "**Requisitos e testes de segurança para brinquedos inteligentes**", apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em _____ de _____ de _____ pela comissão julgadora constituída pelos doutores:

Prof. Dr. _____
Instituição: _____
Presidente

Prof. Dr. _____
Instituição: _____

Prof. Dr. _____
Instituição: _____

Prof. Dr. _____
Instituição: _____

Em memória dos professores Dr. Sdnei de Brito Alves e Dr. Jair Cunha Filho.

Agradecimentos

Aos meus professores da graduação Dr. Alexandre Carlos Brandão Ramos e Dr. Edison Oliveira de Jesus pela formação que me proporcionaram e pelas cartas de recomendação fornecidas para que fosse possível o prosseguimento de meus estudos na USP.

Agradeço a meu amigo e colega de trabalho Dr. Rodrigo Rocha Silva pelo incentivo, conselhos e carta de recomendação.

Ao Centro Paula Souza e Fatec de Mogi das Cruzes por terem me proporcionado afastamento parcial de minhas aulas durante parte do período de duração de meu mestrado.

Ao meu orientador, Dr. Marcelo Medeiros Eler, que acreditou em meu potencial e realmente me ajudou durante todo o meu mestrado.

Por fim, agradeço a meus filhos Mateus e Lucas, e à minha esposa Mariângela por todo o apoio e suporte.

“if I have seen further, it is by standing on ye shoulders of giants.”

(Isaac Newton)

Resumo

CARVALHO, Luciano Gonçalves de. **Requisitos e testes de segurança para brinquedos inteligentes**. 2017. 71 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2017.

Os brinquedos são uma parte essencial de nossa cultura e têm evoluído ao longo do tempo. Atualmente, encontramos no mercado brinquedos dotados de circuitos eletrônicos e sensores, capazes de coletar dados do ambiente e informações pessoais dos usuários. Além disso, eles podem se conectar automaticamente a redes de comunicação por meio de protocolos de rede sem fio para acessar serviços móveis com o objetivo de personalizar a experiência de jogo para cada usuário. Conhecidos como brinquedos inteligentes, estes fazem parte de um ambiente denominado de computação para brinquedos, composto pelo brinquedo físico, um dispositivo móvel, que pode ser um *tablet* ou *smartphone*, e um aplicativo móvel (app), que pode controlar o brinquedo físico e compartilhar informações com serviços móveis. Esse novo tipo de brinquedo, que pertence a um novo tipo de ambiente e que também carrega características da Internet das Coisas, traz consigo questões relacionadas à segurança da informação que não existiam nos brinquedos convencionais. Essas questões, portanto, devem ser tratadas de forma a evitar prejuízos aos usuários dessa tecnologia. Para isso, o presente trabalho apresenta vinte e dois (22) requisitos de segurança gerais identificados por meio da utilização do processo Security Development Lifecycle (SDL) da Microsoft e da técnica de modelagem de ameaças suportada pelo modelo de ameaça STRIDE (Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service e Elevation of privilege). Os requisitos apresentados endereçam questões de segurança que os brinquedos inteligentes devem atender para evitar as principais ameaças existentes. As questões de segurança consideradas foram extraídas da Children's Online Privacy Protection Act (COPPA), General Data Protection Regulation (GDPR) e Personal Information Protection and Electronic Documents Act (PIPEDA). Além dos requisitos de segurança, um conjunto de testes de segurança gerais foram identificados com base no processo SDL para verificar o adequado atendimento a esses requisitos. Uma análise dos brinquedos inteligentes atualmente disponíveis no mercado e suas falhas de segurança relatadas publicamente dão indícios da importância de atender aos requisitos de segurança e executar os testes propostos neste trabalho para evitar problemas de segurança variados.

Palavras-chave: Brinquedo inteligente. Computação para brinquedos. Segurança da informação. Requisitos de segurança. Testes de segurança. Internet das coisas.

Abstract

CARVALHO, Luciano Gonçalves de. **Security requirements and tests for smart toys.** 2017. 71 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2017.

Toys are an essential part of our culture, and they have evolved over time. Currently, we can find in the market toys equipped with electronic circuits and sensors able to collect environmental and personal data from users. They are also able to automatically connect to communication networks through wireless network protocols in order to access mobile services aiming at customizing the gaming experience for each user. Known as smart toys, these are part of a so-called toy computing environment, consisting of a physical toy, a mobile device, which can be a tablet or smartphone, and a mobile app, which can control the physical toy and share information with mobile services. This new type of toy, which belongs to a new type of environment and also present characteristics of the Internet of Things, raises issues regarding information security which did not exist in conventional toys. Such issues, hence, should be treated in a way to avoid losses to the users of this technology. Accordingly, this research project presents twenty two (22) general security requirements identified following the Microsoft Security Development Lifecycle (SDL) and the threat modeling supported by the threat model STRIDE (Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service e Elevation of privilege). These requirements address security issues that smart toys should meet to avoid the main existent threats. All considered issues were extracted from the Children's Online Privacy Protection Act (COPPA), General Data Protection Regulation (GDPR) and Personal Information Protection and Electronic Documents Act (PIPEDA). Furthermore, we have also identified a general set of security tests based on the SDL process to check whether the identified security requirements have been met. A further analysis of the smart toys currently available in the market and their publicly related security flaws give evidence of the importance of meeting the proposed security requirements and executing the proposed security tests to avoid several security problems.

Keywords: Smart toy. Toy computing. Information security. Security Requirements. Security tests. Internet of things.

Lista de figuras

Figura 1. Arquitetura de aplicações móveis.	23
Figura 2. Arquitetura three-tier.....	24
Figura 3. Arquitetura do sistema Android.	25
Figura 4. Evolução do número de incidentes de segurança no Brasil.	26
Figura 5. Evolução do número de incidentes de segurança nos Estados Unidos.	26
Figura 6. Tipos de ataques acumulados em 2016.	29
Figura 7. Arquitetura do HconSTF.....	38
Figura 8. Ambiente de computação para brinquedos.	40
Figura 9. Exemplos de Amiibo.....	41
Figura 10. Sphero.	42
Figura 11. Tech Recon.....	42
Figura 12. Fisher-Price Smart Toy.	43
Figura 13. Hello Barbie.	44
Figura 14. I-QUE Intelligent Robot.....	45
Figura 15. My Friend Cayla.	45
Figura 16. CloudPets.	46
Figura 17. Arquitetura de Ameaça.	48
Figura 18. Diagrama de contexto.	53
Figura 19. DFD de nível 0.	54
Figura 20. DFD de nível 1.	54

Lista de tabelas

Tabela 1. Estrutura do BSIMM6.	34
Tabela 2. Comparação entre brinquedos.	40
Tabela 3. Requisitos de Segurança versus questões e ameaças.	58
Tabela 4. Requisitos versus classes de falhas de segurança.	61
Tabela 5. Requisito de segurança versus classe de testes.	61
Tabela 6. Testes de segurança.	62
Tabela 7. Verificação de atendimento de requisitos de segurança na fase de Implementação.	65
Tabela 8. Verificação de atendimento de requisitos de segurança na fase de Verificação.	65
Tabela 9. Verificação de atendimento de requisitos de segurança na fase de Liberação.	66

Lista de abreviaturas e siglas

AP	Access Point
API	Application Programming Interface
BSIMM	Building Security In Maturity Model
BYOD	Bring Your Own Device
CERT.br	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
COPPA	Children's Online Privacy Protection Act
DFD	Diagrama de Fluxo de Dados
DoS	Denial of Service
GDPR	General Data Protection Regulation
HconSTF	Hcon Security Testing Framework
IDC	International Data Corporation
IDE	Integrated Development Environment
IoT	Internet of Things
MMS	Multimedia Messaging Service
NDK	Native Development Kit
NFC	Near Field Communication
OWASP	Open Web Application Security Project
P2P	Peer-to-peer
PC	Personal Computer
PIPEDA	Personal Information Protection and Electronic Documents Act
PRS	Premium Rate Service
RS	Requisito de Segurança
QoS	Quality of Service
SamuraiSTFU	Samurai Security Testing Framework for Utilities
SamuraiWTF	Samurai Web Testing Framework
SAST	Static Application Security Testing
SCADA	Supervisory Control and Data Acquisition
SDK	Software Development Kit
SDL	Security Development Lifecycle
SGBD	Sistema de Gerenciamento de Banco de Dados
SMS	Short Messaging Service

SQL	Structure Query Language
SSDL	Secure Software Development Lifecycle
SSF	Software Security Framework
SSG	Software Security Group
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of Privilege
UML	Unified Modeling Language
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network
XSS	Cross-Site Scripting

Sumário

1	Introdução.....	15
1.1	<i>Motivação e justificativa</i>	<i>17</i>
1.2	<i>Objetivo</i>	<i>18</i>
1.3	<i>Método.....</i>	<i>18</i>
1.4	<i>Organização do documento</i>	<i>19</i>
2	Referencial Teórico	20
2.1	<i>Computação móvel.....</i>	<i>20</i>
2.2	<i>Segurança da informação</i>	<i>25</i>
2.2.1	SDL.....	31
2.2.2	OWASP	32
2.2.3	BSIMM.....	34
2.2.4	Hcon Security Testing Framework (HconSTF)	37
2.2.5	SamuraiWTF e SamuraiSTFU	38
2.3	<i>Brinquedos inteligentes.....</i>	<i>39</i>
2.3.1	Amiibo.....	41
2.3.2	Sphero.....	41
2.3.3	Tech Recon	42
2.3.4	Fisher-Price Smart Toys	43
2.3.5	Hello Barbie.....	43
2.3.6	My Friend Cayla e I-QUE Intelligent Robot	44
2.3.7	CloudPets.....	45
2.3.8	Problemas de segurança.....	46
2.4	<i>Trabalhos relacionados.....</i>	<i>47</i>
2.5	<i>Considerações finais</i>	<i>49</i>
3	Requisitos de segurança para brinquedos inteligentes	51
3.1	<i>Questões de segurança.....</i>	<i>51</i>
3.2	<i>Modelagem de ameaças</i>	<i>53</i>
3.3	<i>Requisitos de segurança.....</i>	<i>56</i>

4	Testes de segurança para brinquedos inteligentes	59
4.1	<i>Fase de Implementação.....</i>	<i>60</i>
4.2	<i>Fase de Verificação.....</i>	<i>61</i>
4.3	<i>Fase de Liberação.....</i>	<i>62</i>
4.4	<i>Testes de segurança para brinquedos inteligentes disponíveis no mercado</i>	<i>63</i>
5	Considerações finais.....	67
5.1	<i>Contribuições</i>	<i>67</i>
5.2	<i>Publicações</i>	<i>68</i>
5.3	<i>Trabalhos futuros.....</i>	<i>68</i>
	Referências	69

1 Introdução

Dispositivos móveis, tais como *smartphones* e *tablets*, são dispositivos computacionais portáteis utilizados para executar diversas tarefas do dia a dia das pessoas, como, por exemplo, conversar com outras pessoas, enviar mensagens de texto, trocar e-mails, fazer compras e pagamentos, acessar redes sociais etc. A adoção desses tipos de dispositivos, tanto como ferramentas de trabalho no ambiente corporativo quanto como brinquedo nas horas destinadas a lazer, tem crescido ao longo dos anos. Segundo uma pesquisa da Deloitte (2015), setenta por cento dos consumidores mundiais têm acesso a um *smartphone* e cinquenta por cento têm acesso a um *tablet*.

Para que a utilização dos dispositivos móveis seja plena, há a necessidade de recursos específicos de conectividade de rede e software, que propiciam o oferecimento de serviços úteis a seus usuários. As comunicações móveis dependem fortemente da conectividade de redes sem fio providas por diversas tecnologias (Wi-Fi, Bluetooth, *Near Field Communication* – NFC etc.), permitindo a utilização dos serviços móveis.

Já as aplicações móveis (apps) são aplicativos projetados e desenvolvidos para serem executados em dispositivos móveis, cujo desenvolvimento leva em consideração uma plataforma e sistema operacional específicos. Android e iOS são os dois sistemas operacionais mais consolidados para dispositivos móveis. Esses sistemas determinam a linguagem de programação, as APIs (*Application Programming Interfaces*) e os frameworks de desenvolvimento disponíveis para o desenvolvimento de aplicações móveis.

Muitas empresas atualmente dispõem de infraestrutura de hardware e software propícia para que seus colaboradores utilizem dispositivos móveis como ferramenta de trabalho. Como a grande maioria das pessoas possui um dispositivo móvel, é comum que os colaboradores utilizem seus próprios dispositivos no ambiente de trabalho. Essa nova tendência recebeu o nome de *Bring Your Own Device* (BYOD), o que tem motivado o desenvolvimento de políticas para regular essa nova condição no local de trabalho (Miller et. al., 2012).

De fato, uma pesquisa realizada por um grupo de empresas parceiras da Cisco Systems (CISCO, 2013) já havia apontado que, entre os trabalhadores americanos, noventa por cento usam seus próprios smartphones para trabalhar. A utilização de smartphones próprios representa, por exemplo, mais de oitenta por cento em instituições financeiras e de saúde. Ainda de acordo com a pesquisa, a segurança da informação é um dos maiores desafios relacionados ao BYOD.

O acesso a informações sensíveis e a serviços móveis das organizações devem ser cuidadosamente planejados, gerenciados e controlados, evitando acessos não autorizados e vazamento de informações sigilosas. Sendo assim, requisitos envolvendo confidencialidade, integridade e disponibilidade, conceitos chave em segurança da informação, precisam ser levados em conta desde a concepção dos sistemas que manipulam as informações, até a sua implantação e futura manutenção, evitando o surgimento de vulnerabilidades de segurança inerentes ao processo e métodos de desenvolvimento, já que elas podem estar presentes na própria tecnologia empregada, como por exemplo na linguagem de programação adotada, nos servidores de aplicação utilizados, protocolos de comunicação necessários etc.

As vulnerabilidades são fraquezas apresentadas pelos sistemas, que são exploradas por meio de softwares maliciosos (*exploits*), isto é, ameaças a sistemas computacionais que são desenvolvidas e utilizadas por atacantes com a finalidade de comprometer a segurança desses sistemas, culminando em um incidente de segurança. Este, por sua vez, pode ser definido como qualquer tentativa, bem ou malsucedida, de se comprometer a confidencialidade, integridade ou disponibilidade de um ativo.

Embora os maiores alvos de ataques sejam as empresas, qualquer cidadão que possui um dispositivo móvel também pode ser vítima dessas ações criminosas. Em virtude de tais ações, em trinta de novembro de 2012 foi sancionada no Brasil a Lei No. 12.737 (BRASIL, 2012), que dispõe sobre a tipificação criminal de delitos informáticos. Portanto, justificando também a necessidade de se garantir a segurança da informação para cada indivíduo.

Como já citado anteriormente, dispositivos móveis também são utilizados nas horas de lazer e recentemente passaram a ser utilizados como peças-chave em modernos brinquedos, conhecidos como brinquedos inteligentes (*smart toys*). Tais brinquedos, são fruto da combinação entre um brinquedo físico convencional (carros, bonecas etc.) equipado com sensores e componentes eletrônicos que propiciam comunicação em rede, um dispositivo móvel que provê serviços móveis, e um app que propicia a interação com o brinquedo.

Segundo Rafferty e Hung (2015), a combinação entre os componentes físicos convencionais de um brinquedo e as funcionalidades providas por dispositivos móveis, tais como capacidade de conexão de rede sem fio e sensores disponíveis, dão origem a uma nova área de pesquisa, apelada por eles de computação para brinquedos (*toy computing*).

Alguns brinquedos inteligentes são capazes de se conectar automaticamente a redes sem fio para enviar e receber informações de outros dispositivos ou servidores na Internet, isto é, objetos do mundo real que podem fazer parte de redes da Internet das Coisas (Internet of Things - IoT), que por sua vez criam um ambiente adequado que permite a integração de objetos físicos

a sistemas computacionais com o objetivo de proporcionar maior eficiência e precisão, além da redução da intervenção humana em vários domínios.

1.1 Motivação e justificativa

Neste novo cenário, onde não existe uma infraestrutura computacional que auxilie na proteção individual dos usuários contra ameaças virtuais, como no caso das organizações, e que, além disso, passa a incluir crianças como usuários dessa tecnologia, novas preocupações relacionadas à segurança da informação emergem, obrigando todos os responsáveis a refletir sobre essa nova condição.

Partindo do pressuposto de que as pessoas, em momentos de lazer, tendem a não priorizar análises de risco a ameaças virtuais, e que crianças, na grande maioria das vezes, não são capazes de identificar claramente situações de risco, principalmente os relacionados a aspectos tecnológicos, elas acabam sendo alvos fáceis de crimes virtuais e, caso o delito se concretize, podendo acarretar danos catastróficos.

De fato, uma simples pesquisa em motores de busca na Internet vai revelar o quanto eles são vulneráveis a ataques, pois há vários relatos de falhas de segurança em brinquedos inteligentes disponíveis no mercado. Falhas essas que incluem vazamento de informações confidenciais, brinquedos usados como espiões, estranhos interagindo com crianças por meio do brinquedo, dentre outras. Essas falhas podem ameaçar inclusive a segurança física de crianças pois em geral os brinquedos inteligentes tratam informações de localização.

O risco apresentado pelo uso de brinquedos eletrônicos fez com que a agência de investigação dos Estados Unidos FBI (Federal Bureau of Investigation), em julho de 2017, emitisse um alerta às famílias sobre os riscos de trazer brinquedos inteligentes para suas casas (FBI, 2017). De acordo com a agência, muitos brinquedos lidam com funções como reconhecimento facial e de fala, ou armazenam informações que podem ser acessadas pela Internet, e, portanto, podem colocar a privacidade e a segurança das crianças em risco em razão da grande quantidade de dados pessoais que podem ser divulgados por uma eventual falha. Uma das razões para isso é que mecanismos de segurança em brinquedos como estes podem ser negligenciados em razão das pressões para o rápido lançamento de novos produtos no mercado.

Portanto, além da utilização de contramedidas de segurança, estas devem ser exaustivamente testadas de forma a reduzir consideravelmente os riscos envolvidos. Os testes de software são um recurso importante para garantir que todas as funcionalidades providas por um sistema estão funcionando como planejado, isto é, de acordo com os requisitos

especificados. Diversos tipos de testes, segundo seus propósitos, podem ser empregados ao longo de um processo de desenvolvimento de software, como por exemplo os testes funcionais, confiabilidade, performance e segurança.

Embora seja possível encontrar na literatura diversos documentos que envolvem testes de segurança, bem como sua sistematização ao longo de todo ciclo de vida do desenvolvimento de software, trabalhos que abordam segurança em brinquedos inteligentes, ou mesmo para a computação para brinquedos, são escassos, já que, como mencionado anteriormente, essa área é nova e, portanto, carece de estudos em vários aspectos.

1.2 Objetivo

O presente trabalho tem por objetivo geral contribuir com a área de segurança de brinquedos eletrônicos por meio da identificação de um conjunto mínimo de requisitos de segurança, cujo atendimento evitará o surgimento de diversas falhas de segurança, e respectivos testes de segurança capazes de identificar falhas de segurança decorrentes do não atendimento a tais requisitos em brinquedos inteligentes. Os requisitos e testes de segurança identificados são baseados em um ambiente genérico de computação para brinquedos, portanto são abstratos e precisam ser concretizados para cada ambiente específico para que os testes possam ser de fato executados.

Os objetivos específicos deste trabalho incluem a identificação de questões de segurança para brinquedos inteligentes com base na Children's Online Privacy Protection Act (COPPA), General Data Protection Regulation (GDPR) e Personal Information Protection and Electronic Documents Act (PIPEDA); a identificação de ameaças por meio de técnica de modelagem de ameaças, ambos conduzidos segundo as fases do Microsoft Trustworthy Computing SDL (Security Development Lifecycle); identificação de requisitos de segurança alinhados às leis e regulações; identificação de testes de segurança para cada requisito de segurança identificado; mapeamento dos requisitos e testes de segurança possíveis em alguns brinquedos inteligentes disponíveis no mercado.

1.3 Método

A identificação dos requisitos de segurança e dos testes de segurança para brinquedos inteligentes foi realizada tendo como base as fases do processo de desenvolvimento de software SDL (Security Development Lifecycle) de computação confiável (MICROSOFT, 2010),

adotado pela Microsoft Corporation para desenvolvimento de software com segurança, complementado pela utilização de conceitos presentes no Open Web Application Security Project (OWASP, 2016). A escolha desse processo foi motivada por sua facilidade de adaptação para o contexto abordado neste trabalho, por abranger práticas consagradas para proteção de sistemas presentes em outras abordagens, tais como o OWASP e o Building Security In Maturity Model (BSIMM), e por suas diversas atualizações, fruto da efetiva utilização tanto pela Microsoft quanto por terceiros.

Embora o SDL seja utilizado para efetivamente desenvolver um novo software, neste trabalho suas cinco fases servem de guia para análise de segurança sobre um contexto genérico de computação para brinquedo, considerando-se seus principais usuários, crianças, e uma arquitetura típica para este domínio. As fases de Requisitos e Projeto são utilizadas para levantar questões de segurança de interesse do domínio, as ameaças possíveis e o conjunto de requisitos de segurança. As fases seguintes, Implementação, Verificação e Liberação, são utilizadas para identificar diferentes classes de teste para validar cada requisito de segurança identificado. Os detalhes sobre cada fase deste processo de desenvolvimento de software com segurança são apresentados no Capítulo 2.

1.4 Organização do documento

Esta dissertação de mestrado está organizada da seguinte forma. O capítulo 2 apresenta a fundamentação teórica necessária para compreensão do trabalho. O Capítulo 3 apresenta a execução das fases de Requisitos e Projeto do método SDL, portanto apresenta as questões de segurança levantadas para o contexto de brinquedos inteligentes, bem como os requisitos de segurança identificados. O Capítulo 4 apresenta a execução das fases de Implementação, Verificação e Liberação do processo SDL, portanto apresenta os testes de segurança identificados, agrupados por classes de teste. Além disso, neste capítulo é apresentado um mapeamento entre os requisitos e testes de segurança possíveis de serem aplicados em alguns produtos disponíveis no mercado. Por fim, as conclusões deste trabalho são apresentadas no Capítulo 5, juntamente com as contribuições pontuais, os trabalhos futuros e publicações realizadas até o momento.

2 Referencial Teórico

Para que seja possível compreender plenamente os resultados apresentados neste trabalho, considerando-se que este faz interseção com diversas áreas da computação, tais como computação móvel, segurança da informação, testes de software, brinquedos inteligentes e computação para brinquedos, o presente capítulo abordará os conceitos mínimos necessários que embasam a solução desenvolvida.

A seção 2.1 abordará conceitos relacionados à computação móvel. Os princípios fundamentais e complementares da segurança da informação, bem como as ameaças a esses princípios e tipos de teste de segurança serão apresentados na seção 2.2. Na seção 2.3 será detalhado o conceito de brinquedos inteligentes e computação para brinquedos, além de apresentar diversos exemplos de aplicações e problemas de segurança já documentados. A seção 2.4 apresenta as considerações finais deste capítulo.

2.1 Computação móvel

Fruto da necessidade de se ter acesso aos sistemas computacionais a qualquer momento e em qualquer lugar, uma nova geração de dispositivos computacionais e tecnologias de comunicação sem fio foram desenvolvidos, permitindo o estabelecimento da computação móvel.

Segundo Fischer e Smolnik (2013), a computação móvel além de estar relacionada à portabilidade da computação, isto é, o processo de se tornar um serviço computacional móvel, possibilita a localização e personalização das informações, tornando-a útil e tempestiva.

A utilização de notebooks e de tecnologias de rede sem fio locais (*Wireless Local Area Network* - WLAN) deram início à computação móvel, possibilitando o deslocamento físico dos dispositivos ao mesmo tempo em que mantém o acesso a todas às suas funcionalidades.

Ainda de forma a caracterizar a computação móvel, que segundo Satyanarayanan (2001) nasceu da junção entre sistemas distribuídos e clientes móveis, é necessário citar quatro restrições chaves que exigem técnicas de tratamento: variação imprevisível da qualidade da rede, baixa confiança e robustez dos elementos móveis, limitações dos recursos locais (em função de restrições de tamanho e peso dos dispositivos) e consumo de energia (relacionado ao uso de baterias).

B'Far (2005) argumenta que os sistemas computacionais móveis são um supergrupo que contém o grupo de sistemas não móveis, sendo diferentes destes em termos das seguintes dimensões da mobilidade:

- localização: em virtude dos frequentes deslocamentos, os dispositivos móveis e seus aplicativos devem ser capazes de obter a localização atual do dispositivo e tirar proveito dessa informação, fornecendo funcionalidades e recursos, possibilitando a seleção da lógica de negócio adequada, nível de fluxo de trabalho e interfaces;
- qualidade de serviço (*Quality of Service* - QoS): diversos fatores podem prejudicar a conectividade de rede sem fio, tais como clima ruim, erupções solares, locais com muitas barreiras físicas etc., devendo o software móvel ser capaz de lidar com perdas temporárias de conectividade sem, entretanto, deixar de operar;
- capacidades limitadas dos dispositivos: recursos de armazenamento e processamento costumam ser escassos quando comparados a outros sistemas não móveis, portanto, as aplicações devem ser otimizadas para o uso desses recursos;
- fonte de alimentação limitada: sistemas operacionais e aplicações devem encontrar formas de economizar energia, já que os dispositivos móveis utilizam essencialmente baterias como fonte primária de alimentação, sendo estas o maior e mais pesado componente do sistema, e que, portanto, possuem restrições de uso;
- variedade de interfaces de usuário: interface de usuário por voz, telas pequenas, canetas e outros dispositivos apontadores, telas sensíveis ao toque e teclados em miniatura são exemplos comuns de interfaces de usuário;
- proliferação de plataformas: vários fabricantes podem produzir dispositivos móveis, já que possuem menos hardware do que outros tipos de dispositivos, como por exemplo computadores pessoais (*Personal Computer* - PC), e consequentemente mais baratos, impactando o desenvolvimento de aplicações móveis que devem funcionar em várias plataformas de hardware;
- transações ativas: diferentemente dos sistemas não móveis em que, na grande maioria das vezes, as interações com o sistema são iniciadas pelo usuário (sistemas passivos), os sistemas móveis possuem diversas transações ativas, isto é, sequências de interações entre usuário e sistema, que são iniciadas por este, podendo ser síncronas (quando dependem do tempo), como por exemplo, quando exige resposta do usuário em um período de tempo específico, ou assíncronas, onde a resposta a uma requisição pode ser feita a qualquer momento.

As dimensões de mobilidade, como evidenciado anteriormente, impactam o desenvolvimento de aplicações móveis e de sistemas móveis.

Hoje, os dispositivos computacionais móveis mais comuns são os *smartphones* e *tablets*, ao passo que uma das tecnologias de WLAN mais utilizada é o Wi-Fi, embora as tecnologias de rede sem fio pessoais (*Wireless Personal Area Network* - WPAN), como por exemplo Bluetooth e NFC, e de rede sem fio de longa distância (*Wireless Wide Area Network* - WWAN), como por exemplo 3G e 4G, também sejam muito utilizadas.

A partir dos exemplos de dispositivos móveis citados, pode-se perceber que as características que também os distinguem dos dispositivos não móveis são o tamanho (menores que os desktops convencionais), conexões de rede essencialmente sem fio e fonte de energia limitada, além da natureza do uso e funcionalidades adaptadas ao usuário, ambos móveis (B'FAR, 2005).

Os dispositivos móveis, por meio de apps, proporcionam acesso a diversos tipos de serviços móveis (m-services), tais como serviços de voz e de dados. Os principais tipos de serviços móveis são: Peer-to-Peer Short Messaging Service (P2P SMS), Peer-to-Peer Multimedia Messaging Service (P2P MMS), Mobile Internet e Premium Rate Service (PRS).

Como mostrado na Figura 1, normalmente as aplicações móveis são estruturadas em três camadas principais (apresentação, negócio e dados), cuja localização dá origem a dois tipos de software, os clientes baseados em web (*web-based clients*) e os clientes ricos (*rich clients*). Nos clientes baseados em web, as camadas de negócio e dados estão localizadas em um servidor, ao passo que nos clientes ricos essas camadas estão no próprio dispositivo móvel (MEIER et al., 2008).

Na camada de apresentação, os componentes da interface do usuário são responsáveis pela forma de interação do usuário com a aplicação (apresentação, formatação, aquisição e validação de dados), e os componentes de processo sincronizam e orquestram essas interações. Já na camada de negócio, enquanto a fachada de aplicação propiciará o acesso a partir da camada de apresentação por meio de diferentes tecnologias de comunicação, os componentes de negócio, entidades de negócio e fluxos de trabalho de negócio proveem, respectivamente, serviços de negócio, passagem de dados entre componentes e definição e coordenação de processos de negócio com muitas etapas e de longa execução. Por fim, na camada de dados estão presentes os componentes de acesso a dados, que centraliza a funcionalidade de acesso a dados, *data helpers* / utilidades, responsáveis principalmente por maximizar a performance de

acesso a dados, e agentes de serviço, que evitam que a aplicação trate das várias formas de se invocar serviços diversos.

Figura 1. Arquitetura de aplicações móveis.

Fonte: Luciano Gonçalves de Carvalho, 2016.

Quando se opta por desenvolver um cliente baseado em web, embora favoreça a reutilização de código proporcionada pela quebra deste em camadas de acordo com funcionalidades, a escalabilidade só é alcançada por meio da infraestrutura física, onde o código é distribuído entre vários servidores, também de acordo com as funcionalidades.

Para que os requisitos operacionais da aplicação sejam atendidos eficientemente, deve-se estruturar os servidores e efetuar a distribuição de funcionalidades considerando uma arquitetura *n-tier*, onde um *tier* é um grupo de servidores que compartilham características tais como requisitos operacionais (escalabilidade, segurança, performance etc.), mesmo perfil de consumo de recursos (memória, sockets, canais de entrada e saída etc.) e restrições de projeto (segurança, confiabilidade, escalabilidade etc.). A Figura 2 apresenta uma arquitetura *three-*

tier, com servidores de apresentação, aplicação e banco de dados, propiciando segurança, escalabilidade e especialização do banco de dados.

Figura 2. Arquitetura three-tier.

Fonte: Luciano Gonçalves de Carvalho, 2016.

Segundo o International Data Corporation (IDC, 2015), os sistemas operacionais para dispositivos móveis mais utilizados são o Android (82,8%) e o iOS (13,9%). Como o Android atualmente domina o mercado, este trabalho, por questões de redução de escopo, não abordará outros sistemas operacionais.

O Android, atualmente desenvolvido pelo Google Inc., é baseado no núcleo (*kernel*) do sistema operacional Linux, tem *middleware*, bibliotecas e APIs desenvolvidas em linguagem C, e as aplicações executam sobre um framework de aplicação com bibliotecas compatíveis com JAVA. A Figura 3 apresenta a arquitetura do sistema.

As aplicações para Android são desenvolvidas em linguagem JAVA por meio do kit de desenvolvimento de software (*Software Development Kit - SDK*) Android, utilizando o ambiente integrado de desenvolvimento (*Integrated Development Environment - IDE*) Android Studio. Cada aplicação é executada dentro de sua própria máquina virtual Java. Há a possibilidade de se desenvolver aplicações com linguagens de código nativo como C e C++, principalmente por questões de uso intensivo de processamento, *engines* de jogos, processamento de sinais e simulação de fenômenos físicos (DEVELOPERS, 2016), sendo necessário a utilização do kit de desenvolvimento nativo (*Native Development Kit - NDK*).

Figura 3. Arquitetura do sistema Android.

Fonte: <https://source.android.com/devices/index.html>

2.2 Segurança da informação

A maioria dos sistemas de software manipulam e armazenam informações vitais, seja para os negócios a que dão suporte ou para o cidadão comum, sendo consideradas verdadeiros ativos, que neste caso específico recebem a denominação de ativos de informação.

O comprometimento da segurança de um ativo de informação, isto é, a ocorrência de um incidente de segurança, em muitos casos, pode determinar o fim dos negócios de uma empresa ou gerar muitos problemas às pessoas que dele dependem. As Figuras 4 e 5

apresentam, respectivamente, a evolução do número de incidentes de segurança registrados no Brasil e nos Estados Unidos.

Figura 4. Evolução do número de incidentes de segurança no Brasil.

Fonte: CERT.br.

Figura 5. Evolução do número de incidentes de segurança nos Estados Unidos.

Fonte: United States Government Accountability Office (GAO, 2017).

A segurança da informação envolve principalmente a preservação dos seguintes conceitos:

- confidencialidade: a informação poderá ser acessada apenas por quem tem o direito de o fazer, isto é, tem a devida autorização;
- integridade: a informação deverá permanecer exata e completa, isto é, da mesma forma em que foi concebida, bem como seus métodos de processamento, não podendo ser alterada indevidamente;
- disponibilidade: a informação e ativos correspondentes deverão estar disponíveis para acesso sempre que requerido, desde que se tenha a devida autorização.

Embora muitas das questões que envolvem a segurança da informação sejam contemplados pelos conceitos citados anteriormente, fato que faz com que a maior parte da literatura sobre o assunto aborde apenas eles, tais conceitos não são suficientes para tratar todos os problemas e necessidades existentes de forma direta, sendo necessário, portanto, o estabelecimento de conceitos complementares ou, como proposto por Sêmola (2003), aspectos de segurança, tais como:

- autenticidade: garantia de que as partes envolvidas em uma comunicação sejam realmente quem alegam ser;
- irretratabilidade (não-repúdio): impossibilidade de se negar a autoria de uma ação executada;
- legalidade: a informação, bem como sua produção, manipulação e armazenamento, deve estar em conformidade com a legislação vigente;
- privacidade: garantia da não associação entre autor e suas respectivas ações, quando essa situação é necessária e permitida;
- auditabilidade: possibilidade de se auditar tudo o que foi feito em determinado sistema, possibilitando a identificação de possíveis fraudes ou tentativas de ataques.

Normalmente, um conceito de segurança é comprometido por meio de uma ameaça, que se caracteriza pela combinação de uma vulnerabilidade, atacante e ataque.

Vulnerabilidades são fraquezas ou falhas de segurança presentes em sistemas, que podem ser exploradas de forma a permitir sua invasão ou impossibilitar seu correto funcionamento. Uma vulnerabilidade pode ser inserida, propositalmente ou não, pelo programador durante o desenvolvimento do sistema ou durante sua posterior correção, pode surgir devido a uma integração entre sistemas ou ser fruto de vulnerabilidades pertencentes às ferramentas ou tecnologias empregadas na construção ou no suporte dos sistemas.

O atacante é a pessoa interessada em comprometer a segurança da informação, podendo ser classificado, de acordo principalmente com seu objetivo e nível de conhecimento, em (Carvalho, 2005):

- *script kiddies*: com baixo conhecimento técnico, utilizam ferramentas disponíveis na Internet com o único intuito de ser bem-sucedido no ataque, independentemente do alvo;
- *crackers*: dotados de elevado conhecimento técnico, quebram proteções de segurança visando roubar informações e destruir sistemas;
- *insiders*: atacantes que podem ter nível de conhecimentos variados, valem-se de sua posição estratégica na empresa alvo para facilitar sua invasão;
- *coders*: responsáveis pela criação de ferramentas de ataque e consequente utilização ou disponibilização à comunidade de atacantes;
- *white hats*: profissionais experientes contratados por grandes empresas para encontrar vulnerabilidades em sistemas;
- *phreakers*: atacantes com conhecimento técnico avançado que visam o comprometimento de sistemas de telefonia;

Embora o conhecimento sobre os atacantes contribua para se compreender o objetivo dos ataques, estes são a principal informação utilizada pelos provedores de soluções de segurança para evitar o comprometimento dos sistemas. Um ataque é composto pela ferramenta utilizada e consequente tecnologia e método empregados por ela para atacar o sistema.

Atualmente não é possível quantificar o número de ataques diferentes existentes, pois a cada dia esse número cresce e variações de um mesmo ataque podem receber uma nomenclatura diferente, dependendo da empresa que fez a divulgação do problema. Portanto, agrupá-los por tipo ou categorias é uma abordagem interessante. Um exemplo dessa abordagem é a categorização utilizada pelo Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br), que leva em consideração:

- *worm*: propagação automatizada de código malicioso em redes;
- DoS (*Denial of Service* – Negação de Serviço): ataque que visa deixar inoperante um sistema ou impedi-lo de atender a requisições de usuários válidos;
- invasão: comprometimento da confidencialidade, integridade ou ambas, por meio de acesso não autorizado a computadores ou redes de computadores;
- web: desfiguração de páginas web (*defacement*) ou comprometimento de servidores web;

- *scan*: varreduras de sistemas com o objetivo de identificar serviços ativos, que eventualmente apresentem vulnerabilidades que possam ser exploradas;
- fraude: ataques que tentam ludibriar os usuários objetivando obter algum tipo de vantagem.

A Figura 6 apresenta a quantidade acumulada de ataques segundo cada tipo ao longo do ano de 2016. Os ataques que não se enquadram nos tipos apresentados anteriormente são somados à categoria “outros”.

Figura 6. Tipos de ataques acumulados em 2016.

Considerando o tipo de ataque web, que envolve essencialmente comunicação de dados, pode-se dividi-lo nas seguintes subcategorias de ataques:

- interceptação: dados são acessados indevidamente durante uma comunicação;
- modificação: dados são acessados e modificados indevidamente e enviados ao destino original da comunicação;
- fabricação: o atacante insere dados falsos na comunicação original ou cria uma comunicação falsa;
- interrupção: o atacante inviabiliza a comunicação entre origem e destino.

Apesar da existência de diversos ataques, divididos pelos tipos e categorias mencionados anteriormente, pode-se perceber que alguns deles, devido, entre outros fatores, à

sua facilidade de execução, potencial de dano e quantidade de sistemas que apresentam a vulnerabilidade explorada por eles, são amplamente documentados e difundidos, e que merecem especial atenção. Por esse motivo, alguns desses ataques são apresentados a seguir:

- ataque físico: roubo de equipamentos (principalmente *laptops*, *tablets* e *smartphones*), mídias removíveis (CDs, DVDs, pen drives, HDs externos) e documentos impressos;
- engenharia social: ludibriar usuários de sistemas computacionais ou detentores de informações privilegiadas, tais como executivos, de forma a extrair deles informações confidenciais ou acessos normalmente restritos;
- *sniffing*: captura de pacotes de comunicação de dados que trafegam em redes de computadores;
- varredura de porta (*port scanning*): detecção de portas abertas, que estão relacionadas a serviços disponíveis, em servidores;
- força bruta: processo automatizado de tentativa e erro, usado para adivinhar nomes de usuários, senhas, chaves criptográficas etc.;
- injeção de SQL (*SQL Injection*): inserção de blocos de código SQL (Structure Query Language) em campos de entrada da interface de sistemas com o intuito de obter informações confidenciais armazenadas em banco de dados, ou para burlar sistemas de autenticação;
- *buffer overflow*: extrapolação do espaçamento de memória destinado ao armazenamento de dados e consequente sobrescrita de dados adjacentes, podendo fazer com que os sistemas fiquem em um estado imprevisível que pode ser inseguro, isto é, passível de exploração;
- *cross-site scripting* (XSS): inserção de blocos de scripts em campos de entrada de aplicações web de forma a serem incorporados à página web e posteriormente executados nas visualizações seguintes, permitindo desde a desfiguração até o redirecionamento de páginas, burlando alguns mecanismos de autenticação.

A susceptibilidade de sistemas a diversos tipos de ataques pode ser apontada por meio de análises e testes de segurança, que podem ocorrer durante todo o processo de desenvolvimento de software e após a entrada do mesmo em produção.

Para que seja possível organizar e sequenciar o emprego das análises e testes de segurança durante o desenvolvimento e posterior entrada em produção de um software, optou-se neste trabalho por utilizar como referência o processo de desenvolvimento de software SDL (Security Development Lifecycle) de computação confiável (LIPNER, 2004)(MICROSOFT,

2010), adotado pela Microsoft Corporation para desenvolvimento de software com segurança, descrito na seção 2.2.1.

Além do SDL, para organizar especificamente a aplicação de testes de segurança em aplicações, pode-se citar algumas abordagens: OWASP, BSIMM Framework, Hcon Security Testing Framework, SamuraiWTF e SamuraiSTFU.

2.2.1 SDL

O SDL é um processo criado pela Microsoft Corporation que visa garantir a segurança de sistemas a partir de seu desenvolvimento, combinando para isso uma abordagem holística e prática para a redução da quantidade de vulnerabilidades e respectivo grau de severidade.

Para que seja possível introduzir segurança ao software em desenvolvimento, as seguintes fases e respectivas atividades de segurança obrigatórias são necessárias (MICROSOFT, 2010):

- Requisitos: estabelecimento de requisitos de segurança, criação de portões de qualidade e barras de erros (estabelecem níveis mínimos de segurança e privacidade) e avaliação de riscos de segurança e privacidade;
- Design: estabelecimento de requisitos de design, análise de superfície de ataque e modelagem de ameaças;
- Implementação: utilização de ferramentas aprovadas, desaprovação de funções não seguras e execução de análise estática;
- Verificação: execução de análise dinâmica e de testes *fuzzing*, e revisão da superfície de ataque;
- Liberação: elaboração do plano de resposta de incidentes, execução da revisão final de segurança e liberação do software.

As análises de segurança são divididas entre estáticas e dinâmicas, como mencionado no SDL, ao passo que os testes de segurança para softwares podem ser divididos em testes de segurança funcionais, relacionados a requisitos de segurança funcionais, e testes de segurança baseados em risco, relacionados a riscos identificados na análise de risco (MICHAEL et al., 2013).

Na fase de implementação do SDL, encontram-se as análises de segurança estática, que são análises geralmente automatizadas, feitas no código fonte do software (teste de caixa

branca), isto é, sem que o software seja executado, de forma a encontrar defeitos de segurança e prevenir que erros conhecidos passem despercebidos.

As principais técnicas de análise estática de código fonte, segundo o Open Web Application Security Project (OWASP, 2016), são:

- fluxo de dados: coleta de informações sobre como as variáveis de um software são utilizadas;
- grafo de fluxo de controle: representação por meio de grafo, onde cada nó representa um bloco básico de código e as arestas caminhos entre eles, de todos os possíveis caminhos a serem percorridos durante a execução do software;
- análise de contaminação (*taint analysis*): identificação de todas as variáveis que podem ser contaminadas a partir de uma variável modificada por um usuário externo, isto é, valores atribuídos a variáveis a partir da entrada de dados feita pelo usuário;
- análise léxica: conversão da sintaxe da linguagem de programação em *tokens* de informação, possibilitando a comparação eficiente destes com construções vulneráveis, normalmente encontradas em bibliotecas específicas.

Durante a fase de verificação do SDL, as análises dinâmicas e testes *fuzzing* são executados. As análises dinâmicas visam identificar vulnerabilidades que podem levar a problemas tais como de corrupção de memória e de privilégio de usuários durante a execução do software (teste de caixa preta), por meio da simulação de ataques e monitoramento do comportamento do software frente a eles. Já os testes *fuzzing*, que introduzem dados aleatórios ou defeituosos no software, objetivam induzir falhas no sistema.

Após o desenvolvimento do software e posterior entrada em produção, pode-se ainda realizar testes de penetração (*pen test*), cujo objetivo é procurar por vulnerabilidades, tanto para quando não se tem nenhuma informação interna sobre o programa (caixa preta) quanto para quando se tem muitas informações internas relevantes (caixa branca).

Informações detalhadas e atualizadas sobre o SDL podem ser obtidos na página web do processo: <http://www.microsoft.com/sdl>.

2.2.2 OWASP

O guia de testes da OWASP (2013), que atualmente está em sua quarta versão, apresenta um framework de teste de segurança que auxilia no teste de aplicações web, levando em

consideração a inclusão de segurança em cada uma das fases do processo de desenvolvimento de software e utilização das seguintes técnicas:

- inspeção manual e revisão: inspeção manual (humana) sobre as implicações de segurança nas pessoas, políticas e processos;
- modelagem de ameaça: abordagem estruturada que permite a identificação de ameaças de segurança, bem como quantificar e endereçar os respectivos riscos, a partir da análise da arquitetura do sistema e consequente implementação;
- revisão de código: verificação manual do código fonte da aplicação em busca de questões de segurança;
- teste de penetração: teste da aplicação em execução de forma remota e sem conhecimento prévio do funcionamento interno da mesma, de forma a encontrar vulnerabilidades.

Como há diversos processos de desenvolvimento de software, o framework da OWASP considera um modelo de desenvolvimento genérico, tornando possível sua fácil adaptação, e que consiste das seguintes atividades:

- antes do início de desenvolvimento: definição de um processo de desenvolvimento de software, revisão de políticas e padrões, desenvolvimento de medidas e métricas, e garantia de rastreabilidade;
- durante a definição e projeto: revisão de requisitos de segurança, revisão de projeto e arquitetura, criação e revisão de modelos em linguagem de modelagem unificada (Unified Modeling Language - UML), e criação e revisão de modelos de ameaças;
- durante o desenvolvimento: promoção da explicação da lógica e fluxo de código por parte dos desenvolvedores à equipe de segurança (*code walk through*), e revisão de código;
- durante a implantação: execução de testes de penetração e testes de gerenciamento de configuração;
- manutenção e operação: condução de revisões de gerenciamento operacional, condução de checagens de saúde (*health checks*) periódicos e garantir a verificação de alterações.

2.2.3 BSIMM

Resultado do estudo de diversas iniciativas de segurança de software empregadas por grandes empresas, o Building Security In Maturity Model (BSIMM), atualmente em sua sexta versão, tem por objetivo quantificar as atividades conduzidas em iniciativas reais de segurança de software. Ele é composto de cento e doze (112) atividades, organizadas por meio de seu framework de segurança de software (Software Security Framework - SSF), que estão distribuídas em doze (12) tipos de práticas, que por sua vez pertencem a quatro (4) domínios (MCGRAW et al., 2015), como mostra a Tabela 1.

O domínio Governança agrupa atividades para organização, gerenciamento e medição de iniciativas de segurança de software, ao passo que o domínio de Inteligência reúne práticas cujos resultados são conhecimentos utilizados na condução de atividades de segurança de software. A partir desses resultados, análises e avaliações de artefatos e processos de software são feitos por meio das práticas apresentadas no domínio Secure Software Development Lifecycle (SSDL) Touchpoints. Por fim, entendendo que a configuração e manutenção de software, bem como questões relacionadas ao ambiente computacional interferem diretamente na segurança do software, as práticas do domínio de Implantação visam endereçar esses problemas.

Tabela 1. Estrutura do BSIMM6.

Domínios	Práticas
Governança	Estratégias e Métricas
	Conformidade e Política
	Treinamento
Inteligência	Modelo de Ataque
	Recursos de Segurança e Projeto
	Padrões e Requisitos
Secure Software Development Lifecycle (SSDL) Touchpoints	Análise de Arquitetura
	Revisão de Código
	Testes de Segurança
Implantação	Testes de Penetração
	Ambiente de Software
	Gerenciamento de Configuração e Vulnerabilidade

Fonte: Luciano Gonçalves de Carvalho, 2016.

Como evidenciado pela estrutura do BSIMM6, as análises e testes de segurança estão restritos às práticas do domínio SSDL Touchpoints e à prática Testes de Penetração, pertencente

ao domínio de Implantação, e que, portanto, serão as únicas que terão suas atividades mencionadas neste trabalho.

A prática Análise de Arquitetura, onde se documenta a arquitetura de software, possui as seguintes atividades:

- executar a revisão de recursos de segurança: identificação dos recursos de segurança e análise dos mesmos em busca de problemas que afetam seu correto funcionamento ou de sua insuficiência em prover a segurança requerida;
- executar a revisão de projeto para aplicações de alto risco: quebra da arquitetura e revisão detalhada do projeto;
- ter um Grupo de Segurança de Software (Software Security Group - SSG) para conduzir os esforços de revisão de projeto: executar a revisão de projeto e construir a habilidade de identificar falhas;
- usar um questionário de risco para ranquear aplicações: coletar informações básicas sobre a aplicação para determinar o risco e esquemas de priorização;
- definir e usar processos de análise de arquitetura: definição e documentação de um processo de análise de arquitetura e sua aplicação na revisão do projeto pelo SSG;
- padronizar descrições arquiteturais: utilização de formato padrão para descrição de arquitetura;
- fazer com que o SSG esteja disponível como recurso ou mentor na análise de arquitetura: resposta a questionamentos relacionados à análise de arquitetura;
- ter arquitetos de software liderando esforços de revisão de projeto: processos de análise da arquitetura liderados por arquitetos de software;
- guiar resultados das análises para padrões de arquitetura: prevenção de erros futuros por meio do aprendizado com os erros identificados na análise de arquitetura e consequente melhoria nos padrões de projeto.

Caracterizada pela utilização de ferramentas de revisão de código, análise manual e resultados rastreáveis e mensuráveis, a prática Revisão de Código possui as atividades a seguir:

- usar lista de principais bugs: guiar a correção de bugs;
- executar revisão *ad-hoc*: execução de revisão *ad-hoc* de código pela SSG para aplicações de alto risco;
- utilizar ferramentas automatizadas junto com revisões manuais: executar análise estática de código com supervisão humana;

- tornar a revisão de código mandatória para todos os projetos: liberação de software somente após a revisão de código;
- utilizar relatórios centralizados para fechar o ciclo de conhecimento e guiar treinamentos: criação de um repositório centralizado de bugs com o objetivo criar relatório de resumo e de tendências;
- impor padrões de codificação: evitar violações dos padrões de codificação seguros;
- designar mentores de ferramentas: mostrar aos desenvolvedores como extrair o máximo das ferramentas de revisão de código;
- utilizar ferramentas automatizadas com regras adaptadas: customizar a análise estática para reduzir os falsos positivos e aumentar a eficiência;
- construir uma fábrica: escrita de scripts para invocar automaticamente múltiplas técnicas de detecção de vulnerabilidades e combinação dos resultados em um formato consumível por soluções de geração de relatórios;
- criar a capacidade de erradicar bugs específicos de toda a base de código: possibilitar a criação de regras para identificação de um tipo de bug e consequente identificação do mesmo em toda a base de código;
- automatizar a detecção de código malicioso: automatização da detecção de código malicioso, tais como *backdoors*, bombas lógicas, injeção de código dinâmico etc.

A prática Testes de Segurança, cujo foco é na descoberta de vulnerabilidades durante a construção do software, considera as seguintes atividades:

- garantir que a equipe de garantia da qualidade dê suporte a testes com condições de valores de borda e fronteira: execução de testes simples que consideram casos de fronteira e condições de borda;
- conduzir testes com requisitos de segurança e recursos de segurança: execução de testes dos mecanismos de segurança derivados de requisitos e recursos de segurança;
- integrar ferramentas de teste de segurança caixa preta nos processos de garantia da qualidade: utilização de ferramentas de teste de segurança como parte do processo de garantia da qualidade;
- compartilhar os resultados de segurança com a equipe de garantia da qualidade: a equipe de segurança deve compartilhar os resultados das análises com a equipe de garantia da qualidade;
- incluir testes de segurança na automação da garantia da qualidade: testes de segurança e funcionais andam juntos como parte dos testes de regressão;

- executar testes *fuzzing* adaptados a APIs: customização de frameworks de testes *fuzzing* para APIs que serão utilizadas;
- conduzir testes a partir dos resultados da análise de risco: utilização dos resultados da prática de Análise da Arquitetura para direcionar os testes;
- alavancar as análises de cobertura: medição da cobertura dos testes visando identificar partes de código não testadas;
- iniciar a construção e aplicação de testes baseados em casos de abuso: incorporação de casos de teste baseados em casos de abuso.

A prática de Testes de Penetração (domínio de Implantação), que envolve a execução de testes de penetração por especialistas, possui as seguintes atividades:

- utilizar profissionais que realizam testes de penetração que sejam externos à organização: profissionais externos além de proporcionarem outros pontos de vista, podem demonstrar que o código da aplicação tem problemas;
- alimentar os sistemas de mitigação e gerenciamento de falhas com os resultados dos testes de penetração: realimentação do processo de desenvolvimento;
- utilizar internamente ferramentas para testes de penetração: criação de competência interna para realização de testes de penetração utilizando ferramentas;
- prover toda informação disponível aos profissionais que executam testes de penetração: proporciona análises mais profundas;
- agendar testes de penetração periódicos para cobertura da aplicação: realização de testes periódicos para garantir que o software não está vulnerável a novos ataques;
- usar profissionais externos à organização para fazer análises profundas: realização de análises profundas para projetos críticos e introdução de novas formas de pensamento sobre segurança;
- adaptar ferramentas e criar scripts: criar ou adaptar ferramentas para testes de penetração que sejam mais efetivas e que ataque exaustivamente os sistemas da organização.

2.2.4 Hcon Security Testing Framework (HconSTF)

Com o objetivo de auxiliar na realização de testes de penetração e varredura de vulnerabilidades, o HconSTF é um framework de código aberto que contém ferramentas para

realização de ataques tais como SQL Injection, XSS, Cross-Site Request Forgery (Falsificação de solicitação entre sites) etc.

As tarefas realizadas pelo HconSTF, que são semi-automatizadas, estão relacionadas aos testes de penetração, desenvolvimento de *exploits* e análise de código malicioso, todos para a web, e sua arquitetura, de modo a ser mais flexível em termos de desenvolvimento e apresentar um menor número de problemas de compatibilidade, foi projetada em camadas, como mostra a Figura 7.

Duas versões da ferramenta estão disponíveis para utilização, uma construída com tecnologia Mozilla, desenvolvida pela Mozilla Foundation, e a outra construída com tecnologia Chromium, desenvolvida pelo Google Inc.

2.2.5 SamuraiWTF e SamuraiSTFU

O Samurai Web Testing Framework (SamuraiWTF) é um ambiente Linux contido em um CD (Live CD), que foi configurado para realização de testes de penetração web. As ferramentas disponíveis nesse framework são capazes de realizar atividades de reconhecimento, mapeamento, descoberta e exploração do alvo.

Figura 7. Arquitetura do HconSTF.

Fonte: Adaptado de Mystry (2014).

Com o objetivo de atender à demanda de testes de segurança em sistemas SCADA (Supervisory Control and Data Acquisition) e de rede inteligente (Smart Grid), a empresa UtiliSec criou o Samurai Security Testing Framework for Utilities (SamuraiSTFU), uma distribuição Linux de código aberto que reúne diversas ferramentas de segurança para testes de

penetração em redes e web, além de ferramentas especializadas para aplicações utilizadas no setor de energia. Segundo Searle (2013), há ferramentas para realização de testes de penetração em redes de computadores, redes de computadores sem fio, web e hardware.

2.3 Brinquedos inteligentes

Com o objetivo fundamental de divertir as pessoas, os brinquedos fazem parte de nossa cultura e, segundo especialistas, são essenciais no desenvolvimento cognitivo, motor e social das crianças. Além do simples brincar, brinquedos também são utilizados como recurso educativo e ferramenta no auxílio à reabilitação (brinquedo terapêutico).

A rápida evolução tecnológica, a consequente redução dos custos de fabricação de dispositivos eletrônicos, o aumento do poder aquisitivo das famílias e o enorme interesse das novas gerações por bens de alta tecnologia, tais como computadores, *tablets* e *smartphones*, fomentou o desenvolvimento de brinquedos de alta tecnologia, isto é, que trazem em sua composição circuitos eletrônicos, sensores e softwares, e que, muitas vezes, são capazes de se comunicar com outros dispositivos por meio de tecnologias de rede sem fio.

Com o surgimento dos brinquedos inteligentes (*smart toys*), que são a união entre um brinquedo convencional (carro, avião, barco, boneca etc.) dotado de elementos eletrônicos e sensores que permite sua conexão a sistemas computacionais, por meio de redes de comunicação de dados, de forma a utilizar serviços que ampliam suas funcionalidades, uma nova área de pesquisa é inaugurada, denominada por Rafferty e Hung (2015) de computação para brinquedos (*toy computing*). A Tabela 2 apresenta uma comparação entre brinquedos tradicionais, brinquedos eletrônicos e brinquedos inteligentes.

Segundo Rafferty e Hung (2015), a computação para brinquedos associa a computação física, cujo objetivo é prover ao brinquedo tradicional sistemas embarcados e sensores que lhe possibilitem maior interação com o ambiente e comunicação com outros dispositivos, aos serviços móveis, que são serviços acessados por dispositivos móveis por meio de redes sem fio.

Os sensores utilizados nos brinquedos podem ser de movimento (acelerômetro, giroscópio, magnetômetro, barômetro, gravidade, aceleração linear e vetor de rotação), posicionamento (campo geomagnético, proximidade e GPS) e ambientais (humidade relativa, luminosidade, pressão e temperatura).

Tabela 2. Comparação entre brinquedos.

	Tradicional	Eletrônico	Inteligente
Interação	Mecânica	Mecânica Sensor	Mecânica Sensor Visual Auditiva <i>Wireless</i>
Coleta de Dados	Não	Sim Limitada	Sim
Compartilhamento de Dados	Não	Sim Limitada	Sim
Armazenamento de Dados	Não	Sim Interna	Sim Externa
Capacidade de Processamento	Não	Sim Limitado	Sim
Capacidade de Comunicação em Rede	Não	Sim Limitado	Sim
Controlado por Dispositivo Móvel	Não	Não	Sim

Fonte: Adaptado de Rafferty e Hung (2015).

A computação para brinquedos necessita de um ambiente que propicie ao brinquedo físico a interação, por meio de uma tecnologia de comunicação sem fio, com o respectivo dispositivo móvel, que será capaz de acessar diversos serviços móveis, para que os usuários tenham a experiência prometida pelos fabricantes. A Figura 8 apresenta os elementos de tal ambiente, onde o usuário interage com o brinquedo e este, por meio da comunicação sem fio entre brinquedo físico e dispositivo móvel promoverão a devida resposta.

Figura 8. Ambiente de computação para brinquedos.

Fonte: Adaptado de Carvalho e Eler (2017).

Muitos serviços móveis, com o intuito de oferecer ao usuário o melhor, mais adequado e tempestivo serviço, fazem uso de dados pessoais e de contexto, como por exemplo idade, sexo e dados de localização, coletados e armazenados nos dispositivos móveis.

Os dados de contexto podem ser coletados de forma voluntária (o usuário voluntariamente informa os dados), observada (identificada pelo próprio dispositivo por meio

de seus sensores) e inferida (depreendida a partir dos dados coletados de forma voluntária e observada).

Os brinquedos inteligentes mais conhecidos são os Amiibo, Sphero, Tech Recon, Fisher-Price Smart Toys, Hello Barbie, My Friend Cayla, I-QUE Intelligent Robot e CloudPets, descritos nas subseções seguintes.

2.3.1 Amiibo

Desenvolvidos pela Nintendo Corporation, os Amiibo são bonecos em miniatura dos personagens mais famosos dos jogos criados pela empresa, como por exemplo, Mario Bros. e The Legend of Zelda, que por meio da utilização da tecnologia NFC, comunicam-se com os principais consoles criados pela própria organização. A Figura 9 apresenta exemplos de Amiibo e consoles.

Dentre os recursos proporcionados pelos Amiibo, pode-se destacar a incorporação do personagem a um jogo, liberação de bônus ou itens especiais, mudança de nível ou personalização do personagem, e desbloqueio de fases.

2.3.2 Sphero

A empresa Sphero criou uma bola robótica, também chamada Sphero, que é controlada por meio de app instalado em um *tablet* ou *smartphone*, utilizando tecnologia bluetooth para a comunicação. A Figura 10 apresenta o modelo mais comum do brinquedo.

Figura 9. Exemplos de Amiibo.

Fonte: Adaptado de <http://www.nintendo.com/amiibo>

Figura 10. Sphero.

Fonte: Adaptado de <http://www.sphero.com/sphero>

2.3.3 Tech Recon

As armas de brinquedo Tech Recon HammerHead e Tech Recon Havoc, desenvolvidas pela empresa Tech 4 Kids, proporcionam uma experiência mais realista de tiros em campos de batalha.

Com a utilização de um app, disponível tanto para o sistema operacional Android quanto para o iOS, permite o acoplamento de um smartphone às armas de brinquedo. A Figura 11 apresenta os dois modelos do brinquedo.

Figura 11. Tech Recon.

Fonte: Adaptado de <http://www.tekrecon.com>

Ao instalar o app e executá-lo no smartphone, ele utilizará a tecnologia de GPS do aparelho para refletir o deslocamento do brinquedo e proporcionará um sistema de troca de mensagens de voz em tempo real e radar de rastreamento. A comunicação é feita por meio da tecnologia Wi-Fi ou 3G.

2.3.4 Fisher-Price Smart Toys

A Mattel Fisher-Price desenvolveu uma linha de brinquedos inteligentes para crianças na faixa etária entre 3 e 8 anos de idade, composta por um macaco, panda e urso de pelúcia, ambos exibidos na Figura 12, que propicia aprendizagem interativa por meio de recursos de reconhecimento de voz e imagem, sendo capazes de coletar dados com o intuito de se adaptar ao contexto, criando uma experiência personalizada.

As atividades que serão realizadas pelo brinquedo são selecionadas por meio de cartões inteligentes (*smart cards*) que os acompanha.

O fabricante disponibiliza um app que possibilita que o brinquedo se conecte a uma rede sem fio para atualizações (atividades bônus) e armazenamento de informações fora do dispositivo.

Figura 12. Fisher-Price Smart Toy.

Fonte: Adaptado de <http://fisher-price.mattel.com>

2.3.5 Hello Barbie

Hello Barbie é uma boneca desenvolvida pela Mattel que está equipada com um microfone, um alto falante e um dispositivo de reconhecimento de voz, permitindo uma conversação em duas vias quando o brinquedo inteligente está conectado a uma rede sem fio. A Figura 13 exibe as três versões disponíveis.

Figura 13. Hello Barbie.

Fonte: Adaptado de <http://helloworldbarbiefaq.mattel.com/>

A tecnologia de reconhecimento de voz foi desenvolvida pela empresa Toy Talk, responsável por receber as gravações de áudio coletadas pelo brinquedo, para que seja possível propiciar a conversação.

Com a utilização de um app disponibilizado pelo fabricante, os pais podem configurar uma conta que permite que eles ouçam as conversas das crianças com a boneca.

2.3.6 My Friend Cayla e I-QUE Intelligent Robot

A Genesis Toys é a empresa responsável pelos brinquedos My Friend Cayla e I-QUE Intelligent Robot. Cayla é uma boneca interativa capaz de entender e responder a perguntas em tempo real, ler histórias, jogar e compartilhar fotos. Já o I-QUE é um robô interativo que conta piadas, responde perguntas, joga com o usuário e reproduz sons. As Figuras 14 e 15 exibem os referidos brinquedos.

Os brinquedos inteligentes utilizam uma conexão Bluetooth para se conectar a um dispositivo móvel, para que possam acessar a Internet. No caso da boneca Cayla, seu app Violet permite a proteção de seus usuários de palavras e imagens sensíveis e ofensivas, além de possibilitar a inclusão de palavras e frases adicionais que não devem ser ouvidas.

Figura 15. My Friend Cayla.

Fonte: Adaptado de <https://www.genesis-toys.com/my-friend-cayla>

Figura 14. I-QUE Intelligent Robot.

Fonte: Adaptado de <http://ique-robot.co.uk/>

2.3.7 CloudPets

A Spiral Toys é a empresa responsável pelos CloudPets, que são animais de estimação de pelúcia dotados de dispositivos eletrônicos com serviços móveis associados, que permite que pais ou responsáveis gravem e enviem mensagens por meio de um app, sendo posteriormente recebidas pelo brinquedo e reproduzidas quando o mesmo é apertado.

As crianças também podem gravar mensagens por meio do brinquedo físico, que serão transmitidas por rede sem fio a um dispositivo móvel próximo. A Figura 16 apresenta as diferentes versões do brinquedo.

Figura 16. CloudPets.

Fonte: Adaptado de <http://cloudpets.com/>

2.3.8 Problemas de segurança

Diversos serviços móveis, tais como comércio eletrônico, Internet banking móvel e serviços baseados em localização (BROLL et al., 2007), utilizam dados de contexto, tais como faixa etária, sexo, localização etc., para prover experiências personalizadas aos usuários. Essas informações são normalmente armazenadas no dispositivo móvel e, como citado anteriormente, podem ser coletadas voluntariamente, observadas e inferidas.

Como muitos brinquedos inteligentes fazem uso de serviços móveis, consequentemente, informações pessoais dos usuários são coletadas, observadas e inferidas de forma a personalizar a experiência de brincar. As informações são coletadas pelo brinquedo físico e enviadas ao app instalado no dispositivo móvel que o controla por meio da rede sem fio. De posse dessas informações, pode-se utilizar serviços móveis providos por servidores web.

A coleta de informações pode se tornar um problema caso controles de segurança sejam mal implementados ou estejam ausentes, pois a confidencialidade e privacidade, por exemplo, podem ser comprometidas devido à vazamentos. Este problema é evidenciado pela facilidade de se encontrar relatos de problemas de segurança em brinquedos inteligentes, tais como vazamento de informações confidenciais e, até mesmo, viabilização da interação entre invasores do sistema e seus usuários, que podem ser crianças.

A Genesis Toys (fabricante dos produtos Cayla e I-QUE) foi acusada por um grupo de consumidores nos Estados Unidos, entre outras coisas, de coletar informações pessoais de crianças (BARANIUK, 2016). Esses brinquedos também possibilitavam que qualquer dispositivo móvel com a tecnologia Bluetooth ativada pudessem se conectar a eles e também

era possível interceptar facilmente a troca de dados entre os brinquedos e os dispositivos móveis.

O app da boneca Hello Barbie podia se conectar automaticamente em redes Wi-Fi inseguras e revelar informações confidenciais e privadas (NEWMAN, 2015). Já o Smart Toy Bear, um dos Fisher-Price Smart Toys, possuía uma vulnerabilidade em seus sistemas de *backend* que propiciava que atacantes tivessem acesso a informações privadas. Alguns servidores web podem falhar ao autenticar usuários, expondo informações e perfis de usuários.

Informações pessoais, que incluíam endereços de e-mail, senhas, gravações de áudio, de mais de meio milhão de pessoas que utilizaram os CloudPets foram comprometidas, pois tais informações estavam armazenadas em um banco de dados desprotegido (HERN, 2017).

Conforme o mercado de brinquedos inteligentes cresce, a preocupação com segurança aumenta, especialmente devido ao uso massivo desses brinquedos por crianças, que são consideradas vulneráveis e, muitas vezes, incapazes de tomar decisões racionais. Em muitos países, as crianças são protegidas por leis e estatutos específicos, como a Children's Online Privacy Protection Act (COPPA) nos Estados Unidos (United States Federal Trade Commission, 1998), em outros países a proteção se estende a todos de modo geral, como no caso da Personal Information Protection and Electronic Documents Act (PIPEDA) no Canadá (CANADIAN PUBLIC WORKS AND GOVERNMENT SERVICES, 2000) e da General Data Protection Regulation (GDPR) na União Europeia (THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION, 2016), justificando a devida investigação de possíveis ameaças e definição de requisitos de segurança que devem ser atendidos pelos brinquedos inteligentes de forma a garantir um nível adequado de segurança a seus usuários.

2.4 Trabalhos relacionados

Conscientes sobre questões relacionadas à segurança e privacidade na união entre brinquedos e apps, Ng e seus colaboradores (NG et al., 2015) apresentaram em seu trabalho duas preocupações principais acerca dessa questão: histórico de localização e dados de rastreamento, e encriptação e segurança de dados. Dispositivos móveis conectados à Internet por meio de um plano de dados móvel ou por meio de uma rede sem fio são suscetíveis a diferentes ataques, como, por exemplo, DoS, *man-in-the-middle*, *spoofing*, etc. Tais dispositivos normalmente armazenam o histórico de localização do usuário que, quando transmitido pela Internet, pode ser interceptado, expondo padrões de movimentação e a localização do indivíduo em tempo real.

Também com o intuito de prover segurança e privacidade em um ambiente de computação para brinquedos, Rafferty e seus colaboradores (RAFFERTY et al., 2015), por meio de um modelo formal de ameaça à privacidade inspirado por técnicas amplamente conhecidas de modelagem de ameaças, investigaram requisitos de privacidade para tal ambiente. A análise feita por eles, que considerou a arquitetura de ameaça ilustrada na Figura 17, foi efetuada em cinco (5) etapas:

1. Visão Geral da Arquitetura: perspectiva arquitetural da aplicação para computação para brinquedos;
2. Ativos e Fluxo de Dados: identificação de ativos e decomposição da aplicação;
3. Ameaças à Privacidade: identificação e mapeamento de ameaças à privacidade;
4. Métodos de Ataque: determinação de árvores de ameaça à privacidade e criação de cenários de casos de mau uso;
5. Requisitos de Privacidade e Controles: requisitos de privacidade e proposta de controle.

Figura 17. Arquitetura de Ameaça.

Fonte: Adaptado de Rafferty et al. (2015).

Ainda segundo os autores, devido à natureza da computação para brinquedos e de acordo com a arquitetura de ameaça, a privacidade é afetada pelas seguintes questões:

- identidade da criança: identidade está associada com os dados coletados;
- dados de localização: coleta de dados de localização e provável associação com identidades.
- capacidades de comunicação de rede: compartilhamento de dados coletados por meio da Internet.

Como resultado do trabalho, Rafferty e seus colaboradores compilaram seis (6) requisitos de privacidade ou direitos de privacidade:

- o direito dos pais ou do guardião legal de requisitar restrições de uso e divulgação de informação privada de suas crianças.
- o direito dos pais ou do guardião legal de acessar, copiar e inspecionar registros coletados de suas crianças.
- o direito dos pais ou guardião legal de requisitar a eliminação dos registros privados de suas crianças ou respectiva correção, caso estejam incorretos.
- o direito dos pais ou guardião legal de requisitar confirmação por meio de um canal de comunicação quando informações privadas de suas crianças são coletadas.
- o direito de registrar reclamações sobre o fabricante do brinquedo.
- o direito de saber onde os dados privados de suas crianças foram compartilhados alheios aos propósitos do brinquedo

Em um outro trabalho recente, Rafferty e seus colaboradores (RAFFERTY et al., 2017) propuseram um modelo conceitual de regra de privacidade onde os pais ou responsáveis legais são os donos dos dados de suas crianças, devendo consentir o compartilhamento dos dados coletados segundo regras de acesso.

Apesar dos trabalhos apresentados nesta seção abordarem questões de segurança para brinquedos inteligentes, eles são restritos a problemas de privacidade e confidencialidade que, embora importantes, não contemplam todas as dimensões da segurança da informação.

2.5 *Considerações finais*

Embora os brinquedos inteligentes já estejam disponíveis, como apresentado nas subseções anteriores, essa nova área de pesquisa apresenta diversos desafios, que abrangem desde a construção de brinquedos físicos que sejam atrativos e altamente interativos, até a criação de aplicações móveis que além de serem capazes de se comunicar e controlar o hardware

do brinquedo físico, devem possibilitar a utilização de serviços móveis de forma a auxiliar e personalizar as experiências dos usuários.

Para que seja possível proporcionar uma experiência rica a qualquer usuário, de forma a não restringir o uso do brinquedo a uma classe particular de clientes, faz-se necessário a coleta, manipulação e armazenamento de informações pessoais, fato que enseja um cuidado maior em relação à segurança dessas informações.

A preocupação com segurança tem aumentado significativamente ao longo dos anos em função da diversidade de problemas existentes, motivando o desenvolvimento de diversas soluções, que normalmente são eficazes em situações particulares, devendo ser adaptadas para cada nova situação, como no caso da computação para brinquedos.

As soluções de segurança devem atender a requisitos de segurança específicos e sua eficácia e abrangência devem ser verificadas por meios de testes de segurança, também específicos.

Devido à inexistência de requisitos e testes de segurança específicos para brinquedos inteligentes e, conseqüentemente, de métodos para aplicação desses testes, associado à importância do tema, justifica-se o estudo e endereçamento dessas questões.

3 Requisitos de segurança para brinquedos inteligentes

Para a identificação de requisitos de segurança gerais, que possam ser utilizados para desenvolver qualquer brinquedo inteligente, foi considerado um ambiente típico de computação para brinquedos: um brinquedo físico controlado por uma aplicação móvel que é executada em um dispositivo móvel, com possibilidade de acesso a serviços móveis.

Como citado anteriormente neste trabalho, o processo SDL da Microsoft foi escolhido para auxiliar na identificação dos requisitos de segurança. As fases de Requisitos e Projeto do SDL são as únicas aplicáveis neste estágio. A primeira levou em conta, como principais fontes de informação sobre questões de segurança, a COPPA, por tratar especificamente de informações pessoais relacionadas a crianças, e a PIPEDA e GDPR, já que informações pessoais e confidenciais são coletadas, manipuladas e armazenadas.

Ao se utilizar a COPPA, PIPEDA e GDPR, respectivamente, de origem americana, canadense e europeia, abrange-se grande parte do mercado consumidor de brinquedos inteligentes e que, geralmente, é mais exigente em relação ao tratamento de informações.

Na fase de Projeto, que objetiva identificar requisitos de segurança relacionados à arquitetura do sistema, levou-se em consideração o ambiente já apresentado na Figura 8 e a arquitetura de ameaça exibida na Figura 17 para a execução da modelagem de ameaças, que abrangeu a identificação de ativos, criação de diagramas de fluxo de dados (DFD) por meio da ferramenta SDL Threat Modeling da Microsoft e identificação de ameaças considerando a classificação de ameaças STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of Privilege), cuja única função é servir como referência para agrupamento de ameaças por categorias.

As questões de segurança identificadas, fruto da fase de Requisitos do SDL, são apresentadas na seção 3.1. As ameaças relativas aos brinquedos inteligentes, resultado da fase de Projeto do SDL, são apresentadas na seção 3.2. Por fim, os requisitos de segurança identificados com base nas questões e ameaças modeladas são apresentados na seção 3.3.

3.1 *Questões de segurança*

A partir da fase de Requisitos do SDL, buscou-se identificar as principais fontes de informação para o levantamento de questões relacionadas à segurança da informação, embasando a definição dos requisitos de segurança. Como os brinquedos inteligentes podem ser utilizados por crianças, a COPPA foi considerada uma importante fonte de informação, pois

define, do ponto de vista da segurança da informação, as seguintes questões a serem endereçadas relacionadas às informações sobre crianças:

- Questão 01 (Q01): fornecer notificação sobre práticas de coleta, uso e divulgação de informações.
- Questão 02 (Q02): obter consentimento dos pais para coleta, uso e divulgação de informações pessoais.
- Questão 03 (Q03): não promover divulgação desnecessária de informações pessoais.
- Questão 04 (Q04): proteger a confidencialidade, integridade e disponibilidade das informações pessoais.

Partindo do pressuposto de que informações pessoais e confidenciais, por padrão, devem ser protegidas, a PIPEDA e a GDPR também foram consideradas.

A PIPEDA define as seguintes questões a serem endereçadas:

- Questão 05 (Q05): prover o mesmo nível de proteção para o processamento de informações de terceiros.
- Questão 06 (Q06): implementar procedimentos para proteger informações pessoais.
- Questão 07 (Q07): documentar os propósitos de se coletar informação pessoal.
- Questão 08 (Q08): obter consentimento individual para a coleta, uso e divulgação de informações pessoais.
- Questão 09 (Q09): especificar o tipo de informação pessoal a ser coletada.
- Questão 10 (Q10): reter informação pessoal somente pelo tempo necessário.
- Questão 11 (Q11): manter as informações pessoais precisas, completas e atualizadas conforme necessário.
- Questão 12 (Q12): proteger as informações pessoais contra perda ou furto, acesso não autorizado, divulgação, cópia, uso e modificação.

As questões a serem endereçadas segundo a GDPR, não abordadas pela PIPEDA, são:

- Questão 13 (Q13): a requisição de consentimento deve ser feita em um formulário inteligível e de fácil acesso.
- Questão 14 (Q14): a notificação de violação é mandatória e o responsável pelo processamento de dados deve notificar seus clientes.
- Questão 15 (Q15): direito de ter os dados pessoais apagados, cessar futuras divulgações de dados e interromper o processamento de dados por terceiros.

- Questão 16 (Q16): inclusão de proteção de dados desde o início da concepção do sistema.

3.2 Modelagem de ameaças

Após o levantamento das questões relacionadas à segurança da informação, procedeu-se a modelagem de ameaças, técnica também presente no OWASP, que corresponde à fase de Projeto do SDL. Segundo esta fase, efetuou-se a decomposição do sistema com auxílio de DFD, gerando o diagrama de contexto, apresentado na Figura 18, o DFD de nível zero (0) e DFD de nível um (1), apresentados nas Figuras 19 e 20, respectivamente.

Embora se pudesse tentar gerar mais DFDs com níveis maiores de detalhamento, tal tarefa dependeria de detalhes específicos de implementação, particulares de cada brinquedo, portanto, não contribuindo para o objetivo deste trabalho.

A partir da decomposição do sistema, que permitiu uma melhor compreensão de seus componentes, os seguintes ativos foram considerados:

- brinquedo físico.
- dispositivo móvel.
- aplicação móvel (app).

Figura 18. Diagrama de contexto.

Fonte: Luciano Gonçalves de Carvalho, 2017.

Figura 19. DFD de nível 0.

Fonte: Luciano Gonçalves de Carvalho, 2017.

Figura 20. DFD de nível 1.

Fonte: Luciano Gonçalves de Carvalho, 2017.

Em virtude da coleta, armazenamento e compartilhamento de dados realizados pelo dispositivo móvel por meio do app, os seguintes ativos de informação foram considerados:

- banco de dados do app.
- arquivos de configuração.
- arquivos de dados (localização, imagens, vídeo, áudio, etc.).

A interação entre usuário, dispositivo móvel e brinquedo físico, segundo uma análise de superfície de ataque, expõe os seguintes pontos de entrada do sistema:

- interface do app: o usuário insere dados na interface móvel.
- sensores do dispositivo móvel: o dispositivo móvel obtém dados de seus sensores.
- sensores do brinquedo físico: o brinquedo físico obtém dados de seus sensores.
- comunicação entre brinquedo físico e dispositivo móvel: comunicação por meio de redes sem fio utilizando protocolos de WLAN ou WPAN tais como Wi-Fi, Bluetooth, NFC, etc.
- comunicação entre dispositivo móvel e provedores de serviço móvel: comunicação por meio de redes sem fio utilizando protocolos de WLAN e WWAN tais como Wi-Fi, 3G, 4G, etc.

A partir das informações obtidas durante a modelagem de ameaças e com o suporte do modelo STRIDE, foi possível identificar as seguintes ameaças:

- A1: Spoofing
 - A1.1: as crianças não estão brincando, mas sim um atacante (*insider*), que deseja descobrir informações confidenciais.
 - A1.2: um atacante está usando outro dispositivo móvel para controlar o brinquedo por meio do protocolo Bluetooth.
 - A1.3: o provedor de serviços móveis é falso.
- A2: Tampering
 - A2.1: modificação do arquivo de configuração do dispositivo móvel (carrega um arquivo de configuração inadequado para o usuário).
 - A2.2: modificação da informação trocada entre os componentes do sistema (brinquedo físico, dispositivo móvel e ponto de acesso (Access Point - AP)/roteador).
 - A2.3: modificação do banco de dados no dispositivo móvel (mudanças nos pontos do jogo, histórico de ações do usuário, etc.).

- A3: Repudiation
 - A3.1: o usuário nega a compra de serviços, acessórios, etc.
- A4: Information disclosure
 - A4.1: divulgação de informações pessoais armazenadas no banco de dados.
 - A4.2: divulgação de informações utilizadas para requisição de serviços móveis (dados de localização, contexto, etc.).
 - A4.3: divulgação de informações armazenadas no dispositivo móvel (fotos, vídeos, mensagens de texto, etc.)
- A5: Denial of Service
 - A5.1: um serviço insere um volume de informações no banco de dados suficiente para alcançar a capacidade total do sistema de armazenamento do dispositivo móvel.
 - A5.2: mais de um dispositivo móvel envia comandos para o brinquedo físico fazendo com que ele não seja capaz de responder corretamente.
 - A5.3: um atacante inviabiliza o acesso a serviços móveis por meio de ataque ao AP/roteador.
- A6: Elevation of privilege
 - A6.1: um atacante observa os dados trocados entre o dispositivo móvel e o brinquedo físico por meio da rede de comunicação, e os utiliza para ter acesso ao brinquedo.
 - A6.2: um atacante observa os dados trocados entre o dispositivo móvel e os serviços móveis por meio da rede de comunicação, e os utiliza para ter acesso aos serviços móveis.

3.3 Requisitos de segurança

Com base nas dezesseis (16) questões relacionadas à informação e nas quinze (15) ameaças identificadas, foi possível especificar os seguintes requisitos de segurança gerais:

- RS01: O app do brinquedo inteligente deverá informar o usuário sobre quais informações são coletadas e seu uso futuro e práticas de divulgação.
- RS02: O app do brinquedo inteligente deverá prover uma interface específica para identificar a idade do usuário e obter seu consentimento antes da coleta e manipulação

de informações pessoais; no caso de usuários menores de idade, obter consentimento dos responsáveis e permitir sua revisão, ambos de forma verificável.

- RS03: O app do brinquedo inteligente não deverá solicitar mais informações pessoais para que ele continue sua operação.
- RS04: O app do brinquedo inteligente deverá autenticar os usuários.
- RS05: A comunicação entre o brinquedo físico e o dispositivo móvel deverá ser feita por meio de protocolos que possuam mecanismos de autenticação e autorização.
- RS06: Provedores de serviços móveis deverão possuir certificados digitais, permitindo a verificação de identidade.
- RS07: A integridade de arquivos de configuração deverá ser mantida e verificada a cada sessão de jogo.
- RS08: Cada comunicação no ambiente de computação para brinquedos deverá utilizar mecanismos de criptografia.
- RS09: O Sistema de Gerenciamento de Banco de Dados (SGBD) deverá prover autenticação de usuários.
- RS10: O SGBD deverá prover mecanismos de segurança contra modificação externa de dados armazenados.
- RS11: O app do brinquedo inteligente deverá solicitar renovação de autenticação antes de cada transação financeira.
- RS12: O SGBD deverá prover mecanismo próprio de encriptação de dados ou permitir que ferramentas de terceiros cumpram essa função.
- RS13: O app do brinquedo inteligente deverá criptografar informações pessoais acessadas de outros apps instalados no mesmo dispositivo móvel.
- RS14: O app do brinquedo inteligente não deverá acessar arquivos desnecessários de outros apps instalados no mesmo dispositivo móvel.
- RS15: O app do brinquedo inteligente deverá monitorar e limitar o crescimento da base de dados.
- RS16: O brinquedo físico não deverá aceitar comandos de outros dispositivos móveis não pertencentes à sessão atual de jogo.
- RS17: Toda comunicação deverá utilizar protocolos seguros com mecanismos criptográficos.
- RS18: O app do brinquedo inteligente deverá exibir a política de privacidade quando requerido pelo usuário.

- RS19: O brinquedo inteligente deverá apagar toda a informação pessoal desnecessária que foi coletada.
- RS20: O brinquedo inteligente deverá manter as informações pessoais precisas, completas e atualizadas conforme for necessário.
- RS21: O app do brinquedo inteligente deverá notificar os usuários sobre brechas de segurança e atualizações disponíveis.
- RS22: O desenvolvimento e atualizações do brinquedo inteligente deverão considerar princípios de segurança amplamente conhecidos.

A Tabela 3 apresenta as questões e ameaças que são endereçadas por cada um dos requisitos de segurança especificados.

Tabela 3. Requisitos de Segurança versus questões e ameaças.

Requisitos de Segurança	Questões e/ou Ameaças
RS01	Q01 e Q08
RS02	Q02, Q08 e Q13
RS03	Q03
RS04	A1.1
RS05	Q04, Q12 e A1.2
RS06	Q04, Q05, Q12 e A1.3
RS07	Q04, Q12 e A2.1
RS08	Q04, Q05, Q06, Q12 e A2.2
RS09	Q04, Q12 e A2.3
RS10	Q04, Q12 e A2.3
RS11	A3.1
RS12	Q04, Q06, Q12 e A4.1
RS13	Q04, Q06, Q12 e A4.2
RS14	Q04, Q12 e A4.3
RS15	Q04 e A5.1
RS16	Q04 e A5.2
RS17	Q04, Q06, Q12, A6.1 e A6.2
RS18	Q07 e Q09
RS19	Q10 e Q15
RS20	Q11
RS21	Q14
RS22	Q16

Fonte: Luciano Gonçalves de Carvalho, 2017.

4 Testes de segurança para brinquedos inteligentes

Após a execução das primeiras fases do SDL, Requisitos e Projeto, e consequente especificação dos requisitos de segurança gerais para brinquedos inteligentes em um ambiente de computação para brinquedos, as três últimas fases, Implementação, Verificação e Liberação, foram fundamentais para mapear os testes de segurança que demonstram o atendimento a esses requisitos. A identificação de testes de segurança que possam evidenciar o adequado atendimento aos requisitos de segurança é fundamental para garantir que vulnerabilidades conhecidas não estejam presentes no brinquedo inteligente.

Durante a fase de Implementação, procede-se à detecção de falhas de segurança no código fonte por meio da utilização de ferramentas de análise de código fonte, também conhecidas como *Static Application Security Testing* (SAST), prática também presente no BSIMM (prática Revisão de Código).

Ferramentas SAST são utilizadas para identificar falhas de segurança, que podem ser provenientes da não utilização das melhores práticas para codificação de controles de segurança, implementação inadequada dessas soluções e eventual ausência de tais mecanismos, amplamente conhecidas em nível de código fonte.

Na fase de Verificação, há a necessidade de se utilizar testes de segurança mais especializados de forma a garantir que o código da aplicação atenda adequadamente aos requisitos de segurança previamente estabelecidos. Portanto, cada requisito de segurança ainda não verificado foi classificado, quando aplicável, de acordo com uma das seguintes classes de testes de segurança (OWASP, 2013):

- Gerenciamento de Configuração e Instalação;
- Gerenciamento de Identidade;
- Autenticação;
- Autorização;
- Gerenciamento de Sessão;
- Validação de Entrada;
- Tratamento de Erro;
- Criptografia fraca;
- Lógica de Negócio;
- Lado do Cliente.

Uma vez identificado a classe de teste a qual o requisito de segurança se enquadra, torna-se possível estabelecer o teste adequado à sua verificação. Embora as classes de testes de segurança apresentadas não tenham sido originalmente criadas para aplicações móveis, elas são gerais o suficiente para serem utilizadas neste contexto.

Já na fase de Liberação, foi possível verificar, por meio dos testes alfa e beta, todos os requisitos de segurança que dependiam da existência de uma versão funcional da aplicação.

Embora a utilização de testes alfa e beta para verificação do atendimento aos requisitos de segurança seja a melhor estratégia, deve-se, na sua ausência desses tipos de testes, executar testes de penetração, como os utilizados no HconSTF, SamuraiWTF e SamuraiSTFU, adaptados ao ambiente de computação para brinquedos.

A seguir, são apresentados os resultados da execução das três últimas fases do processo SDL. Os testes de segurança capazes de verificar o devido atendimento aos requisitos de segurança previamente estabelecidos, organizados segundo as fases de Implementação, Verificação e Liberação do SDL, são apresentados na seção 4.1, 4.2 e 4.3, respectivamente. Por fim, na Seção 4.4 são apresentados testes possíveis de serem aplicados a alguns brinquedos inteligentes disponíveis no mercado.

4.1 Fase de Implementação

Durante a fase de Implementação, utiliza-se ferramentas SAST para identificar falhas de segurança no código fonte do app do brinquedo inteligente. Para o sistema operacional Android, as ferramentas QARK, Androbugs e JAADAS costumam ser utilizadas para execução de análises estáticas. Tais ferramentas normalmente serão capazes de identificar falhas de segurança das seguintes classes ou áreas (OWASP, 2017):

- validação de dados.
- autenticação.
- gerenciamento de sessão.
- autorização.
- criptografia.
- tratamento de erros.
- procedimento de *login*.
- configuração de segurança.
- arquitetura de rede.

A Tabela 4 apresenta os requisitos de segurança cujo atendimento poderá ser verificado por meio da utilização de ferramentas de análise estática de código fonte, bem como as respectivas classes de falhas de segurança relacionadas.

Tabela 4. Requisitos versus classes de falhas de segurança.

Requisitos de Segurança	Classes de Falhas de Segurança
RS04	Autenticação
RS05	Autenticação e Autorização
RS08	Criptografia
RS09	Autenticação
RS13	Criptografia
RS17	Criptografia
RS22	todas as classes

Fonte: Luciano Gonçalves de Carvalho, 2017.

4.2 Fase de Verificação

Na fase de Verificação, onde testes de segurança mais especializados são necessários para garantir o atendimento de requisitos de segurança, identificou-se para cada requisito de segurança a ser verificado nesta fase a classe de testes capaz de atestar seu adequado atendimento, como mostra a Tabela 5.

Tabela 5. Requisito de segurança versus classe de testes.

Requisito de Segurança	Classe de Testes
RS06	Criptografia fraca
RS07	Lógica de Negócio
RS10	Lógica de Negócio
RS12	Criptografia fraca
RS14	Gerenciamento de Configuração e Instalação
RS15	Gerenciamento de Configuração e Instalação
RS19	Gerenciamento de Configuração e Instalação

Fonte: Luciano Gonçalves de Carvalho, 2017.

Após a identificação das classes de testes, procedeu-se a identificação dos testes específicos pertencentes a essas classes, de forma a verificar o requisito de segurança associado a ela, como mostra a Tabela 6.

Tabela 6. Testes de segurança.

RS	Teste	Descrição
RS06	Teste de falhas em protocolos com criptografia	Checar se a autoridade certificadora é confiável Checar se o certificado é válido Checar se o nome do site é o mesmo do certificado
RS07	Teste de integridade de verificações	Verificação de que a aplicação não permite aos usuários destruir a integridade de qualquer parte do sistema ou seus dados
RS10	Teste de integridade de verificações	Verificação de que a aplicação não permite aos usuários destruir a integridade de qualquer parte do sistema ou seus dados
RS12	Teste de criptografia	Verificação do arquivo gerado pelo banco de dados
RS14	Teste de configuração da plataforma da aplicação	Verificação da configuração de cada elemento da arquitetura da aplicação
RS15	Teste de configuração da plataforma da aplicação	Verificação da configuração do banco de dados
RS19	Teste de configuração da plataforma da aplicação	Verificação da configuração de cada elemento da arquitetura da aplicação

Fonte: Luciano Gonçalves de Carvalho, 2017.

4.3 Fase de Liberação

Já na fase de Liberação, foi possível executar testes para a verificação dos demais requisitos de segurança propostos neste trabalho, já que os mesmos demandam que haja uma versão funcional da aplicação. Portanto, testes operacionais alfa e beta, que neste caso são direcionados a verificar aspectos relacionados à segurança da informação, cumprem adequadamente esse papel.

Os testes alfa são capazes de verificar os seguintes requisitos de segurança:

- RS01: um usuário humano deve usar a aplicação para encontrar a notificação requerida.
- RS02: um usuário humano deve usar a aplicação para encontrar a interface requerida.

- RS03: um usuário humano deve usar todos os recursos disponibilizados pelo brinquedo inteligente sem informar nenhuma informação pessoal adicional.
- RS11: um usuário humano deve se deparar com uma interface de autenticação antes de acessar qualquer recurso pago.
- RS16: um usuário humano deve enviar comandos ao brinquedo físico durante uma sessão de jogo de outro usuário.
- RS18: um usuário humano deve requerer acesso à política de privacidade e a obter em seguida em vários momentos distintos.

Os testes beta serviram para verificar os seguintes requisitos de segurança:

- RS20: um usuário final deve checar a acurácia, completeza e atualidade de seus dados.
- RS21: um usuário final deve checar se a aplicação o alerta sobre brechas de segurança e atualizações disponíveis.

4.4 Testes de segurança para brinquedos inteligentes disponíveis no mercado

Os vinte e dois (22) requisitos de segurança identificados neste trabalho endereçam muitas das questões gerais de segurança associados a brinquedos inteligentes em um ambiente de computação para brinquedos. Ao se analisar os problemas reais de segurança apresentados pelos brinquedos inteligentes Smart Toy Bear, Hello Barbie, My Friend Cayla, I-QUE Intelligent Robot e CloudPets, todos documentados na seção 2.3.8 desta dissertação, pode-se verificar que o atendimento dos requisitos de segurança propostos seria suficiente para evitar seu surgimento.

As falhas de segurança no Smart Toy Bear que foram causadas pelo uso de APIs inseguras, permitindo que atacantes acessassem informações pessoais e enviassem comandos para o brinquedo físico, seriam evitadas pelo atendimento aos requisitos de segurança RS04, RS05, RS06, RS07, RS08, RS12, RS16 e RS17.

Diversas vulnerabilidades, que envolvem questões de interceptação de comunicações, divulgação de informações pessoais e conexões a redes Wi-Fi inseguras, foram descobertas na boneca Hello Barbie. Tais problemas poderiam ser evitados por meio do atendimento aos requisitos de segurança RS05, RS06, RS07, RS08, RS12 e RS17.

A boneca My Friend Cayla e o I-QUE Intelligent Robot, entre outros problemas, permitiam que atacantes requisitassem informações pessoais aos usuários (crianças) e

possibilitavam conexões não autorizadas via protocolo Bluetooth de qualquer dispositivo móvel próximo ao brinquedo. Os problemas relatados poderiam ter sido evitados se tais brinquedos atendessem aos requisitos de segurança RS03, RS04 e RS05.

No caso dos CloudPets, cujas informações pessoais dos usuários foram acessadas por atacantes devido a falhas de segurança relacionadas ao acesso ao banco de dados que as armazenava, poderiam ser evitadas com o atendimento dos requisitos de segurança RS09 e RS12.

É importante destacar que menos da metade dos requisitos de segurança propostos já foi suficiente para endereçar os problemas de segurança atualmente conhecidos, indicando que o atendimento de mais requisitos poderia conferir um grau de segurança maior a um brinquedo inteligente. Entretanto, o atendimento a todos os requisitos levantados não garante segurança total a ataques, pois, embora a COPPA, PIPEDA e GDPR abordem diversas questões relativas à segurança da informação, servindo adequadamente para se atingir os objetivos gerais deste trabalho, elas podem não atender plenamente às questões particulares de alguns países. Neste caso, uma análise personalizada, levando-se em consideração leis e regulamentações locais, deverá ser feita de forma a propiciar uma identificação mais precisa de requisitos de segurança.

Com o conhecimento de quais requisitos de segurança cada um dos brinquedos inteligentes deveria atender de forma a evitar os problemas de segurança já documentados, pode-se estabelecer, de forma geral, quais testes seriam necessários para se verificar o adequado atendimento a tais requisitos.

Considerando a ordem de aplicação de testes adotada neste trabalho, que segue as três últimas fases do SDL, e os requisitos de segurança sugeridos para cada um dos exemplos de brinquedos inteligentes que tiveram problemas de segurança documentados, foi possível estabelecer quais testes deveriam ser aplicados caso os respectivos requisitos tivessem sido atendidos. Durante a fase de Implementação do SDL, ferramentas SAST seriam suficientes para verificar parte dos requisitos de segurança para cada um dos brinquedos, como mostra a Tabela 7.

Na fase de Verificação, exige-se testes de segurança mais específicos, de acordo com as particularidades de cada brinquedo inteligente, isto é, a arquitetura, linguagem de programação, bibliotecas, etc., dificultando a execução de tais testes quando não se conhece os detalhes e, portanto, sendo possível especificar apenas os tipos de testes a serem executados, como mostra a Tabela 8.

Tabela 7. Verificação de atendimento de requisitos de segurança na fase de Implementação.

Brinquedo Inteligente	Requisitos de Segurança	Testes de Segurança
Smart Toy Bear	RS04, RS05, RS08 e RS17	Ferramentas SAST
Hello Barbie	RS05 e RS17	
My Friend Cayla	RS04 e RS05	
I-QUE Intelligent Robot	RS04 e RS05	
CloudPets	RS09	

Fonte: Luciano Gonçalves de Carvalho, 2017.

Tabela 8. Verificação de atendimento de requisitos de segurança na fase de Verificação.

Brinquedo Inteligente	RS	Testes de Segurança
Smart Toy Bear	RS06	Teste de falhas em protocolos com criptografia
	RS07	Teste de integridade de verificações
	RS12	Teste de criptografia
Hello Barbie	RS06	Teste de falhas em protocolos com criptografia
	RS07	Teste de integridade de verificações
	RS12	Teste de criptografia
My Friend Cayla	Não possui	Não aplicável
I-QUE Intelligent Robot	Não possui	Não aplicável
CloudPets	RS12	Teste de criptografia

Fonte: Luciano Gonçalves de Carvalho, 2017.

Por fim, na fase de Liberação, os requisitos de segurança restantes seriam verificados por meio dos testes alfa e beta, como mostra a Tabela 9.

É relevante relatar que será difícil medir a eficácia dos testes de segurança propostos neste trabalho devido a seu caráter mais geral, pois testes mais detalhados do ponto de vista de sua execução exigiria a obtenção de acesso aos exemplos de brinquedos inteligentes citados, o que é inviável devido ao custo dos mesmos.

Uma alternativa seria a criação de um protótipo para possibilitar um detalhamento maior dos testes de segurança, que, para ser efetivo, deveria ser criado sem a intervenção do autor deste trabalho, de forma a não influenciar na segurança do mesmo. Outra característica importante de tal protótipo seria o oferecimento de inúmeros recursos ao usuário, de forma a simular todas as possibilidades de interação e execução de procedimento que poderiam ser suscetíveis a ataques, o que, do ponto de vista do desenvolvimento, seria custoso e demorado. Portanto, tal alternativa não fez parte do escopo desta dissertação.

Tabela 9. Verificação de atendimento de requisitos de segurança na fase de Liberação.

Brinquedo Inteligente	RS	Teste	Descrição
Smart Toy Bear	RS16	alfa	Um usuário humano deve enviar comandos ao brinquedo físico durante uma sessão de jogo de outro usuário.
Hello Barbie	Não possui	Não aplicável	Não aplicável
My Friend Cayla	RS03	alfa	Um usuário humano deve usar todos os recursos disponibilizados pelo brinquedo inteligente sem informar nenhuma informação pessoal adicional.
I-QUE Intelligent Robot	RS03	alfa	Um usuário humano deve usar todos os recursos disponibilizados pelo brinquedo inteligente sem informar nenhuma informação pessoal adicional.
CloudPets	Não possui	Não aplicável	Não aplicável

Fonte: Luciano Gonçalves de Carvalho, 2017.

5 Considerações finais

Os brinquedos fazem parte de nossa cultura há muito tempo e sua utilização pode ter objetivos diversos, tais como o mero entretenimento, servir de ferramenta de ensino e até mesmo auxiliar em tratamento médico de crianças, exemplos que evidenciam a diversidade de público e importância para a humanidade.

Com o passar do tempo, os brinquedos foram se tornando mais sofisticados, e hoje se pode encontrar no mercado exemplares denominados brinquedos inteligentes (*smart toys*), que incluem diversos componentes eletrônicos e sensores, e que, muitas vezes, permitem que sejam controlados e personalizados por meio de apps instalados em dispositivos móveis tais como *tablets* e *smartphones*, utilizando para isso protocolos de comunicação sem fio.

A combinação entre os componentes físicos convencionais de um brinquedo e as funcionalidades providas pelos dispositivos móveis utilizados para a interação dão origem à área de pesquisa conhecida como computação para brinquedos (*toy computing*), que prevê um ambiente onde o brinquedo inteligente acessa serviços móveis e compartilha informações pessoais com o objetivo de prover um alto nível de personalização da experiência.

Como alguns brinquedos inteligentes são capazes de se conectar automaticamente a redes sem fio, permitindo a troca de informações com outros dispositivos sem intervenção humana, tais artefatos passam a fazer parte da Internet das Coisas (IoT), ampliando ainda mais as possibilidades de interação.

Devido ao alto grau de desenvolvimento dos brinquedos inteligentes e tecnologias associadas, constata-se que, diferentemente dos brinquedos tradicionais, esses brinquedos passam a ser um alvo potencial de diversas ameaças virtuais, e, portanto, necessitando de proteção para garantir a segurança da informação.

5.1 Contribuições

A seguir são listadas as contribuições pontuais deste projeto de pesquisa cujo objetivo é aumentar a segurança dos brinquedos inteligentes.

Identificação de vinte e dois (22) requisitos gerais de segurança da informação para brinquedos inteligentes, de forma a evitar que um conjunto considerável de ameaças obtenha êxito em comprometer a segurança desses brinquedos e consequentemente prejudicar seus usuários, que podem ser extremamente vulneráveis, como no caso de crianças. Esses requisitos

foram identificados com base em regulações e leis como a COPPA, a PIPEDA e GDPR, que são importantes fontes de informações relacionadas a questões e segurança.

Identificação de um conjunto geral de testes de segurança visando a atender aos requisitos de segurança identificados. Os testes identificados são um guia geral cujas implementações dependem de detalhes específicos da arquitetura e implementação dos brinquedos inteligentes a serem avaliados.

Mapeamento dos requisitos e testes necessários para garantir a segurança de um conjunto de brinquedos inteligentes disponíveis no mercado, cujos problemas de segurança já foram reportados publicamente. Entende-se que os requisitos de segurança especificados, quando atendidos, poderiam ter evitado os problemas de segurança apresentados por esses brinquedos. Isso demonstra a importância de se levar em consideração requisitos de segurança específicos durante o processo de desenvolvimento desses tipos de brinquedo.

5.2 *Publicações*

Até o término da escrita desta dissertação, as contribuições científicas originadas diretamente desta pesquisa resultaram na seguinte publicação:

- CARVALHO, L.; ELER, M. M..Security Requirements for Smart Toys. In: 19th International Conference on Enterprise Information Systems, 2017, Porto - Portugal.

5.3 *Trabalhos futuros*

Como trabalhos futuros pretende-se implementar um protótipo sofisticado de brinquedo inteligente utilizando um dispositivo móvel como brinquedo físico, já que este pode ser dotado de diversos recursos de comunicação e interação. Este protótipo propiciará um ambiente adequado para a implementação dos requisitos de segurança bem como a concretização dos testes de segurança gerais identificados nesta dissertação. Este ambiente também propiciará oportunidades para a criação de ferramentas de apoio para a execução e avaliação dos testes de segurança. Além disso, pretende-se identificar padrões de segurança específicos para aplicações utilizadas em brinquedos inteligentes com base na arquitetura dos brinquedos e aplicações, bem como dos requisitos e testes de segurança aqui especificados. Por fim, pretende-se aplicar testes de segurança em brinquedos inteligentes disponíveis no mercado com o objetivo de identificar falhas de segurança relacionadas às ameaças e requisitos identificados neste trabalho.

Referências

B'FAR, R. Mobile Computing Principles, Designing and Developing Mobile Applications with UML and XML. Cambridge University Press, 2005.

BARANIUK, C. Call for Privacy Probes over Cayla Doll and I-QUE Toys. BBC News, Technology, 6 dec 2016. Disponível em < <http://www.bbc.co.uk/NEWS/TECHNOLOGY-38222472>>. Acessado em 12 dez. 2016.

BRASIL, Lei No. 12.737, de 30 de Novembro de 2012. Disponível em <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12737.htm>.

BROLL, G. et al. Modeling Context Information for Realizing Simple Mobile Services. In Mobile and Wireless Communications Summit, 2017, 16th IST, pp.1-5.

CANADIAN PUBLIC WORKS AND GOVERNMENT SERVICES. Personal Information Protection and Electronic Documents Act, 2000.

CARVALHO, L. G. Segurança de Redes. Editora Ciência Moderna. Rio de Janeiro, 2005.

CARVALHO, L. G.; ELER, M. M. Security Requirements for Smart Toys. In Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017), 2017, Volume 2, pages 144-154.

CISCO. BYOD Insights 2013: A Cisco Partner Network Study, 2013. Disponível em: <<http://www.ciscomcon.com/sw/swchannel/registration/internet/registration.cfm?SWAPPID=91&RegPageID=350200&SWTHEMEID=12949>>. Acessado em 22 mai. 2016.

DELOITTE. 2015 Global Mobile Consumer Survey: US Edition - The rise of the always-connected consumer, 2015. Disponível em: <<http://www.deloitte.com/us/mobileconsumer>>. Acessado em 22 mai. 2016.

DEVELOPERS. Android NDK, 2016. Disponível em: <<https://developer.android.com/ndk/index.html?hl=pt-br>>. Acessado em 25 de mai. 2016.

FBI, Federal Bureau of Investigation, 2017. Disponível em: <<https://www.ic3.gov/media/2017/170717.aspx>>. Acessado em 15 de nov. 2017.

FISCHER N.; SMOLNIK S. The Impact of Mobile Computing on Individuals, Organizations, and Society – Synthesis of Existing Literature and Directions for Future Research. 46th Hawaii International Conference on System Sciences, 2013.

GAO, United States Government Accountability Office, 2017. Disponível em: <<https://www.gao.gov/products/GAO-17-440T>>. Acessado em 17 de out. 2017.

HERN, A. CloudPets Stuffed Toys Leak Details of Half a Million Users. Technology, The Guardian, 28 Feb. 2017. Disponível em: <<https://www.theguardian.com/technology/2017/feb/28/cloudpets-data-breach-leaks-details-of-500000-children-and-adults>>. Acessado em 15 de nov. 2017.

IDC. Smartphone OS Market Share, 2015 Q2. 2015. Disponível em <<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>>. Acessado em 25 de mai. 2016.

LIPNER, S. The Trustworthy Computing Security Development Lifecycle. Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04), IEEE, 2004.

MCGRAW et al. BSIMM6. 2015. Disponível em <<https://www.bsimm.com/>>. Acessado em 26 de mai. 2016.

MEIER et al. Mobile Application Architecture Guide. Application Architecture Pocket Guide. Microsoft patterns & practices. 2008.

MICHAEL et al. Risk-Based and Functional Security Testing. 2013. Disponível em: <<https://buildsecurityin.uscert.gov/articles/bestpractices/securitytesting/riskbasedandfunctionalsecuritytesting#sectest>>. Acessado em 26 de mai. 2016.

MICROSOFT. Implementação Simplificada do Microsoft SDL. 2010. Disponível em <<https://www.microsoft.com/pt-br/download/details.aspx?id=12379>>. Acessado em 26 de mai. 2016.

MILLER, K. W. et al. BYOD: Security and Privacy Considerations, IT Professional, IEEE, v. 14, i. 5, p. 53-55, 2012.

MISTRY, A. Hcon Security Testing Framework Manual. 2014. Disponível em <https://sourceforge.net/projects/hconframework/files/HconFramework-Fire/HconSTF_0.5_Prime/HconSTF_Manual.pdf/download>. Acessado em 26 de mai. 2016.

NEWMAN, J. Internet-connected Hello Barbie Doll Can Be Hacked. PC World, Security, News, 7 Dec 2015. Disponível em <<http://www.pcworld.com/article/3012220/security/internet-connected-hello-barbie-doll-can-be-hacked.html>>. Acessado em 12 dez. 2016.

NG et al. Toys and Mobile Applications: Current Trends and Related Privacy Issues. Mobile Services for Toy Computing, International Series on Computer Entertainment and Media Technology, Springer, 2015.

OWASP. Static Code Analysis. 2016. Disponível em <https://www.owasp.org/index.php/Static_Code_Analysis>. Acessado em 25 de mai. 2016.

OWASP. Testing Guide 4.0. 2013. Disponível em <https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf>. Acessado em 25 de mai. 2016.

OWASP. Mobile AppSec Verification 0.9. Disponível em <https://www.owasp.org/index.php/File:OWASP_Mobile_AppSec_Verification_Standard_v0.9.3.pdf>. Acessado em 15 de nov. 2017.

RAFFERTY, L.; HUNG, P. C. K. Introduction to Toy Computing. Mobile Services for Toy Computing. International Series on Computer Entertainment and Media Technology, Springer, 2015, p. 1-7. ISSN 2364-947X.

RAFFERTY et al. Privacy Requirements in Toy Computing. Mobile Services for Toy Computing. International Series on Computer Entertainment and Media Technology, Springer, 2015, p. 141-173. ISSN 2364-947X.

RAFFERTY et al. Towards a Privacy Rule Conceptual Model for Smart Toys. In Proceedings of the 50th Hawaii International Conference on System Sciences, HICSS, 2017.

SATYANARAYANAN, M. Pervasive Computing: Vision and Challenges. IEEE Personal Communications, August 2001.

SEMOLA, M. Gestão da Segurança da Informação: Uma Visão Executiva. Elsevier Editora, 2003;

THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. Regulations, Official Journal of the European Union, 2016.