



II Workshop Regional de Segurança da Informação
e de Sistemas Computacionais
25 setembro de 2017
Santa Maria, RS

Anais

Sociedade Brasileira de Computação (SBC)

Organizador
Raul Ceretta Nunes (UFSM)

Realização
Departamento de Computação Aplicada
Universidade Federal de Santa Maria (UFSM)

Copyright © 2017 da Sociedade Brasileira de Computação
Todos os direitos reservados

Dados Internacionais de Catalogação na Publicação (CIP)

S612 Workshop Regional de Segurança da Informação e de Sistemas Computacionais (II. 2017: Santa Maria, RS).

Anais [do] WRSeg 2017, II Workshop Regional de Segurança da Informação e de Sistemas Computacionais, 25 de setembro de 2017, Santa Maria / Sociedade Brasileira de Computação ; Organizador, Raul Ceretta Nunes – Santa Maria, RS: Sociedade Brasileira de Computação, 2017.

78 p.
Internet (pdf)

ISBN: 978-85-7669-412-0

1. Ciência da computação. 2. Informática. 3. Segurança da informação. 4. Segurança de sistemas. 5. Redes de Computadores I. Nunes, Raul Ceretta (org.). II. Universidade Federal de Santa Maria. III. Sociedade Brasileira de Computação. VI. Título.

CDD – 004.060981

Sociedade Brasileira de Computação – SBC

Presidência

Lisandro Zambenedetti Granville (UFRGS), Presidente
Thais Vasconcelos Batista (UFRN), Vice-Presidente

Diretorias

Renata de Matos Galante (UFRGS), Diretora Administrativa
Carlos André Guimarães Ferraz (UFPE), Diretor de Finanças
Antônio Jorge Gomes Abelém (UFPA), Diretor de Eventos e Comissões Especiais
Avelino Francisco Zorzo (PUC-RS), Diretor de Educação
José Viterbo Filho (UFF), Diretor de Publicações
Claudia Lage da Motta (UFRJ), Diretora de Planejamento e Programas Especiais
Marcelo Duduchi Feitosa (CEETEPS), Diretor de Secretarias Regionais
Eliana Silva de Almeida (UFAL), Diretora de Divulgação e Marketing

Diretorias Extraordinárias

Ricardo de Oliveira Anido (UNICAMP), Diretor de Relações Profissionais
Esther Colombini, Diretora de Competições Científicas
Raimundo José de A. Macêdo (UFBA), Diretor de Cooperação com Sociedades Científicas
Cláudia Cappelli (UNIRIO), Diretora de Articulação com Empresas

Conselho

Mandato 2015-2019

Altigran Soares da Silva (UFAM)
Ana Carolina Salgado (UFPE)
Fabio Kon (USP)
Rodolfo Azevedo (UNICAMP)
Paulo Roberto Freire Cunha (UFPE)

Mandato 2017-2021

José Carlos Maldonado (USP)
Roberto da Silva Bigonha (UFMG)
Rosa Maria Vicari (UFRGS)
Cristiano Maciel (UFMT)
Itana Maria De Souza Gimenes (UEM)

Suplentes - Mandato 2017-2019

Alex Sandro Gomes (UFPE)
Ismar Frango Silveira (UNICSUL, Mackenzie)
Sérgio Lifschitz (PUC-RIO)
Hermano Perrelli de Moura (UFPE)
André Luís Alice Raabe (UNIVALI)

Contato

Av. Bento Gonçalves, 9500
Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia
91.509-900 – Porto Alegre RS

CNPJ: 29.532.264/0001-78

<http://www.sbc.org.br>

Comitês de Organização da 15^a ERRC

Coordenação Geral

Carlos Raniery Paula dos Santos (UFSM)

Coordenação do Comitê de Programa

Érico Hoff Amaral (Unipampa)
Gleizer Voss (IF Farroupilha/São Vicente)

Coordenação do WRSeg

Raul Ceretta Nunes (UFSM)

Organização de Minicursos, Palestras e Oficinas

Simone Ceolin (CTISM- UFSM)
Tiago Coelho Ferreto (PUC-RS)

Organização de Caravanas

Roben Lunardi (IFRS)

Organização Local

Roseclea Duarte Medina (UFSM)
Marcia Pasin (UFSM)

Organização de Divulgação e Patrocínios

Márcia Henke (CTISM- UFSM)
Marcelo da Silva Conterato (SENAC-RS)

Webmaster

Thales Nicolai Tavares (UFSM)

Comitê Consultivo

Jeferson Campos Nobre (UNISINOS)
Weverton Cordeiro (UFRGS)
Luis Augusto Dias Knob (IMED)

Organização Local

Anderson Monteiro (IFFar/UFSM)
Brenda Salenave (CT - UFSM)
Caroline Chagas (CT - UFSM)
Daniel Matheus (CT - UFSM)
Dênes Vargas (CT - UFSM)
Filipe Simões (CT - UFSM)
Gabriel Cardoso (CT - UFSM)
Ivana Fischer (CTISM - UFSM)
Jhillian Bianchi (CT - UFSM)
Larissa Souto (CTISM - UFSM)

Leonardo Marcuzzo (UFSM)
Luísa Perin Lucca (CT - UFSM)
Maurício Matter Donato (CT - UFSM)
Nilton Camargo (UFSM)
Otávio Prestes (CTISM - UFSM)
Paulo Vinicius Cardoso (CT - UFSM)
Rafael Guimarães (CT - UFSM)
Thales Nicolai Tavares (UFSM)
Vinícius Fülber Garcia (UFSM)
Yagor Duarte (CTISM - UFSM)

Mensagem do Coordenador Geral

É um prazer e uma distinção organizar um evento de tamanha relevância para a computação no estado, mais ainda por completar-se 10 anos desde a primeira vez que Santa Maria teve o privilégio de sediá-lo. Agradecemos à comunidade pela confiança em contribuir com uma das escolas regionais mais tradicionais e relevantes nacionalmente. No contexto desafiador em que o país se encontra nesse ano de 2017, é ainda mais importante reforçar o papel primordial do ERRC como fórum de discussão, encontros e divulgação dos melhores trabalhos produzidos pela comunidade de pesquisa regional.

A programação do ERRC 2017 está diversificada, abrangente e tem excelente qualidade; nesse sentido, a contribuição da comunidade foi fundamental para a valorização do evento e o fortalecimento da Ciência e Tecnologia no estado. A programação engloba 6 sessões técnicas com apresentação de 22 artigos científicos completos, selecionados por meio de um rigoroso trabalho de revisão, 5 palestras proferidas por pesquisadores internacionalmente renomados. São oferecidos ainda 5 minicursos, voltados à formação e atualização dos participantes em temas de ponta.

A excelência das atividades programadas nesta edição é reflexo da competência e empenho dos seus respectivos coordenadores. Um agradecimento muito especial a Marcia Pasin (UFSM), Roseclea Duarte Medina (UFSM), Simone Ceolin (CTISM-UFSM), Raul Ceretta Nunes (UFSM), Érico Hoff do Amaral (Unipampa), Gleizer Voss (IF-Farroupilha), Tiago Coelho Ferreto (PUC-RS), Roben Lunardi (IFRS), Márcia Henke (CTISM- UFSM) e Marcelo da Silva Conterato (SENAC-RS). Exaltamos ainda o trabalho voluntário, intenso, atencioso e contínuo, realizado pelos colegas do Comitê de Organização Local.

Agradecemos aos integrantes do Comitê Consultivo do ERRC, pelos aconselhamentos prestados à organização do ERRC 2017. Gostaríamos ainda de agradecer aos patrocinadores do simpósio: aos órgãos de fomento, CAPES e FAPERGS, aos nossos apoiadores e às empresas patrocinadoras por valorizarem e reconhecerem o ERRC como um evento importante para o fomento à pesquisa e inovação.

Nosso agradecimento especial às nossas instituições, especialmente, ao Departamento de Computação Aplicada (DCOM) e ao Departamento de Linguagens e Sistemas de Computação (DLSC), ao Colégio Técnico Industrial de Santa Maria (CTISM), ao Centro de Tecnologia e à Reitoria da UFSM, pelo indispensável suporte para a realização do ERRC.

Em nome do Comitê Organizador do ERRC 2017, desejamos a todos uma semana agradável em Santa Maria, rica em discussões e encontros.

Carlos Raniery Paula dos Santos
Coordenador Geral da ERRC 2017

Mensagem do Coordenador do WRSeg

O Workshop Regional de Segurança da Informação e de Sistemas Computacionais (WRSeg), evento integrante da Escola Regional de Redes de Computadores (ERRC), é um fórum que visa incentivar a participação de alunos de graduação e pós-graduação na produção e divulgação de trabalhos científicos e técnicos nas áreas de segurança da informação e de sistemas computacionais. Este fórum visa promover a reunião de pesquisadores e profissionais interessados no desenvolvimento da área, fomentando o debate e a troca de experiências, estimulando assim o engajamento em pesquisas na área de segurança computacional.

Nesta segunda edição o WRSeg contou com 15 submissões de artigos completos, dos quais 10 foram selecionados para publicação e apresentação. O número de 15 submissões revela uma demanda real por este espaço de discussão junto à ERRC. Esta demanda, aliada a sempre presente necessidade de fomento a atividades de iniciação científica com qualidade, aponta para a consolidação do WRSeg.

Para a realização da segunda edição, o Comitê de Programa foi constituído por 17 pesquisadores, que juntos realizaram mais de 50 revisões. Deste modo, meu agradecimento especial à imensurável contribuição de todos os membros do Comitê de Programa, pela competência e dedicação na condução do processo de avaliação dos artigos. Gostaria de aproveitar a oportunidade para agradecer, também, aos coordenadores e organizadores da ERRC 2017, pelo constante apoio dedicado ao WRSeg, o qual foi essencial para sua viabilização.

A programação do evento está composta por 2 palestras e 2 sessões técnicas, cada uma com 5 artigos. As palestras nos brindam com dois temas de fundamental importância: *i*) os desafios em desenvolvimento seguro; e *ii*) gerenciamento de identidades e monitoramento de federações. Nas seções técnicas, além de seus temas básicos, também serão discutidos temas como: segurança em nuvem, privacidade, segurança em IoT, testes de penetração, WAFs, dentre outros. Enfim, a diversidade de temas retrata a riqueza do programa desta segunda edição do WRSeg.

Saúdo a todos os participantes do II Workshop Regional de Segurança da Informação e de Sistemas Computacionais e desejo um ótimo workshop a todos!

Raul Ceretta Nunes
Coordenador do WRSeg

Comitê de Programa do WRSeg

Adenauer Yamin (UCPEL e UFPEL)
Avelino F. Zorzo (PUCRS)
Bruno Dalmazo (Univ. of Coimbra)
Bruno Mozzaquattro (UFSM)
Cristiano Bonato Both (UFCSPA)
Érico Hoff do Amaral (UNIPAMPA)
Jéferson Campos Nobre (UNISINOS)
Luciano Ignaczak (UNISINOS)
Luis Augusto Dias Knob (IFRS - Sertão)
Márcia Henke (UFSM)
Marco Antônio Sandini Trentin (UPF)
Raul Ceretta Nunes (UFSM) - Coordenador
Raul Fernando Weber (UFRGS)
Roben Castagna Lunardi (IFRS - Restinga)
Simone Ceolin (UFSM)
Tiago Antônio Rizzetti (UFSM)
Weverton Cordeiro (UFRGS)

Sumário

Comitês de Organização da 15^a ERRC	iv
Mensagem do Coordenador Geral	v
Mensagem do Coordenador do WRSeg	vi
Comitê de Programa do WRSeg	vii

Palestra 1

<i>Desafios em Desenvolvimento Seguro – uma visão do Front da Batalha</i> Sean Michael Wykes (Nascent Secure Technologies)	1
---	---

Palestra 2

<i>Gestão de Identidades e Monitoramento de Federações</i> Ricardo Tombesi Macedo (UFSM/FW)	2
--	---

Sessão Técnica 1

Chair: Raul Ceretta Nunes (UFSM)

<i>Segurança na Computação em Nuvem: Um Estudo de Caso Sobre a Viabilidade de sua Implantação</i> Gabriel Pozzebon	3
---	---

<i>Elaboração de Modelo para Proteção de Dados em Discos Virtuais usando os Esquemas IBE e ICP</i> Paulo Renato de Moraes Vieira, Luciano Ignaczak	11
---	----

<i>Randomização de Endereço MAC como Técnica para Prover Privacidade a Usuários de Redes WiFi</i> Bruno S. Alves, Yagor S. Duarte, Bolívar M. Silva	17
--	----

<i>Mecanismo de Autenticação de Dispositivos para Internet das Coisas</i> Jonathan Monteiro Araujo, André Peres	23
--	----

<i>Proposição de um Sistema de Autenticação Simplificado e Interativo com Dispositivo IoT</i> Fabio Lopes Brezolin, Erciles Andrei Bellei, Jucélia Giacomelli Beux, Marco A. Sandini Trentin, Angelo Elias Dalzotto, Joao Mário L. Brezolin	29
--	----

Sessão Técnica 2

Chair: Luis Knob (IFRS/Sertão)

<i>Protocolo de Busca a Testes de Penetração em Dispositivos Móveis</i> Guilherme Leal Kaiser, Daniel Dalalana Bertoglio	35
<i>Uma Proposta de Arquitetura para Identificação de Anomalias em Redes IoT utilizando Registros de Logs</i> Jonathan Ortiz Preuss, Bolívar M. Silva, Raul Ceretta Nunes	44
<i>Anomaly-based Web Application Firewall using HTTP-specific features and One-Class SVM</i> Nico Epp, Ralf Funk, Cristian Cappo	50
<i>Detecção de Botnets através da Análise do tráfego DNS e Engenharia Reversa</i> Juliano Stolpe	56
<i>Propondo uma Análise de Risco focada na Singularidade da Internet das Coisas</i> Silvio Antonio Beskow, Érico S. Rocha	63

PALESTRA 1

Desafios em Desenvolvimento Seguro – uma visão do Front da Batalha

25 setembro 2017, 14:00 - 15:30, Auditório INPE

Resumo: Em um mundo de ataques digitais maciços e automatizados, onde os inimigos não são virtuais, aumentar a segurança dos sistemas se tornou prioridade máxima. É preciso melhorar significativamente a forma pela qual projetamos, desenvolvemos e validamos software, e essa melhoria contempla necessariamente o “tripé” de pessoas, processos e tecnologia. Quais os desafios práticos encontrados em projetos de hardware e software, o que podemos fazer para torná-los mais seguros, e quais as questões que ainda nos impedem de usar a frase “desenvolvimento seguro” com plena convicção?



Bio: Britânico, Sean Michael Wykes é mestre em Engenharia de Informação pela Universidade de Southampton. Radicado no Brasil, ele possui mais de 20 anos de experiência prática em design e desenvolvimento de projetos de software e sistemas seguros, com base em tecnologias como cartões inteligentes e elementos seguros. É CTO da empresa brasileira Nascent Secure Technologies, onde se dedica à aplicação prática das tecnologias de criptografia, consultoria especializada, treinamentos e workshops teórico-práticos, além de mentoria tecnológica de equipes no Brasil e no exterior. Ele é autor do livro "Criptografia Essencial - A Jornada do Criptógrafo" pela Editora Elsevier, 2016.

PALESTRA 2

Gestão de Identidades e Monitoramento de Federações

25 setembro 2017, 17:30 - 19:00, Auditório INPE

Resumo: Os Sistemas de Gerenciamento de Identidades (SGI) vêm recebendo atenção da academia e da indústria devido ao seu potencial em integrar diferentes domínios administrativos, preservando tecnologias e políticas locais. A principal vantagem destes sistemas consiste em empregar autoridades de autenticação como guardiões das informações críticas dos usuários, separando o provimento de recursos do gerenciamento dos dados críticos dos usuários. Através dos SGIs, a autenticação de usuários em um único domínio possibilita o acesso a múltiplos domínios, reduzindo a complexidade do gerenciamento e incrementando a experiência dos usuários. Esta palestra aborda os desafios relacionados com o monitoramento e a disponibilidade das autoridades de autenticação, as principais soluções desenvolvidas nas últimas décadas para contornar estes desafios e as oportunidades de desenvolvimento de pesquisa para tornar os SGIs mais seguros. Além disto, a palestra apresenta experiências práticas quanto ao desenvolvimento de soluções de segurança para federações de identidades.



Bio: Ricardo Tombesi Macedo é professor adjunto do Departamento de Tecnologia da Informação da Universidade Federal de Santa Maria (UFSM), campus Frederico Westphalen. É Doutor em Ciência da Computação pela Universidade Federal do Paraná (com período sanduíche na Universidade de *La Rochelle* - França), Mestre em Engenharia da Produção pela UFSM e Bacharel em Ciência da Computação pela Universidade de Cruz Alta (Unicruz). Atualmente está realizando o Pós-Doutorado junto ao CCSC (*Center for Computational Security sCience* - Centro de Ciência de Segurança Computacional) da Universidade Federal do Paraná (UFPR). Seus interesses de pesquisa consistem em redes sem fio, redes veiculares, gerenciamento de identidades e redes definidas por software.

Segurança na Computação em Nuvem: Um Estudo de Caso Sobre a Viabilidade de sua Implantação

Gabriel Pozzebon¹

¹Curso Superior de Tecnologia em Redes de Computadores – Colégio Técnico Industrial de Santa Maria (CTISM) – Universidade Federal de Santa Maria (UFSM)
Caixa Postal 97015-900 – Santa Maria – RS – Brasil

gabriel.pozzebon@redes.ufsm.br

Resumo. O presente artigo tem por finalidade realizar um estudo da segurança em Cloud Computing. Devido ao aumento do uso da computação em nuvem, os problemas de segurança foram se agravando, com isso, teve-se a ideia de realizar um estudo bibliográfico sobre o assunto de computação em nuvem e os problemas de segurança existentes e como mitigá-los a ponto de tentar resolver possíveis falhas existentes. Como resultado, esperamos responder as questões de segurança encontradas na nuvem, a fim de viabilizar uma possível implementação de ativos físicos em um ambiente de nuvem seguro.

Palavras-chave: Cloud Computing. Computação em Nuvem. Segurança. Implementação.

Abstract. This article aims to conduct a study of security in Cloud Computing. Due to the increased use of cloud computing, security issues have worsened, with the idea of conducting a bibliographic study on the subject of cloud computing and existing security problems and how to mitigate them. try to resolve existing flaws. As a result, we expect to address the security issues found in the cloud in order to enable a possible deployment of physical assets in a secure cloud environment

Keywords: Cloud Computing. Cloud computing. Safety. Deployment.

1. Conceito de Computação em Nuvem

A Computação em nuvem possibilita acessar recursos computacionais, como por exemplo, servidores e sistemas de armazenamento de maneira prática e a qualquer momento. Para pequenas empresas, por exemplo, adquirir equipamentos e expandir a infraestrutura, bem como adquirir licenças de *software* pode gerar uma grande economia com um ganho no futuro (VERAS, 2012).

2. Principais Modelos de Serviços na Nuvem

Serviços na computação em nuvem é o conceito de poder reutilizar componentes, isto é conhecido como “*as a service*” (como um serviço) (VELTE, 2012). Estes modelos de serviços são muito importantes pois definem um padrão de arquitetura para as soluções

em computação em nuvem.

Nos últimos anos, a computação em nuvem tem sido responsável por grandes mudanças na área de TI (Tecnologia da Informação), uma vez que essas mudanças têm impactado no crescimento e desenvolvimento de empresas, na medida em que a nuvem oferece cada vez mais serviços, recursos, segurança, facilidades e com custos cada vez mais atraentes para tamanhos diferentes de empresas (OPUS SOFTWARE, 2015).

2.1. IaaS – Infrastructure as a Service

IaaS (Infraestrutura como um Serviço) é a capacidade que um provedor de serviços tem de oferecer a clientes uma infraestrutura de armazenamento e processamento de forma transparente. Basicamente, o cliente deixa de adquirir *hardwares* e *softwares* e passa a gerir máquinas virtuais através de virtualização (VERAS, 2012).

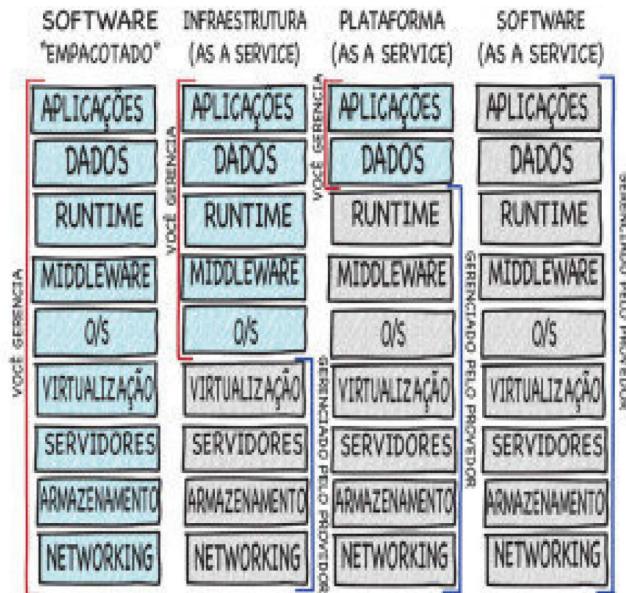
2.2. PaaS – Platform as a Service

O ambiente PaaS (Plataforma como um Serviço) fornece uma infraestrutura completa de *hardware*, armazenamento, sistemas operacionais e internet para implantação de aplicativos. Nesse tipo de serviço, o cliente apenas tem que criar e executar suas ações, não precisando se preocupar com detalhes de baixo nível da plataforma (REESE, 2009). O PaaS oferece vários serviços voltados para o desenvolvimento de aplicativos que interagem com dispositivos móveis. (OPUS SOFTWARE, 2015).

2.3. SaaS – Software as a Service

O SaaS (Software como um Serviço) é o modelo em que um aplicativo é distribuído como um serviço aos clientes empresariais que o acessam através da internet (VELTE e ELSENPETER, 2012). Exemplos dessa aplicação mais familiares estão as aplicações de gerenciamento de relacionamento com clientes como o *Salesforce*, pacotes de produtividade como o *Google Apps* e o *Dropbox*.

Na figura abaixo, um demonstrativo dos três principais modelos de nuvem, indicando quem pode utilizar tal modelo, os serviços disponíveis e para qual função mais adequada para cada modelo de nuvem.

**Figura 1. Modelos de Nuvem SaaS, PaaS, IaaS.**

Fonte: Opus Software

3. Segurança na Computação em Nuvem

A segurança na computação em nuvem não é diferente da TI convencional, no entanto, com os diferentes modelos de implantação apresentam riscos diferentes (GURKOK, 2014). Muitas das ameaças podem ser combatidas com práticas de segurança simples, enquanto outras exigem soluções mais específicas da nuvem, uma vez que cada arquitetura pode ter diferentes níveis de vulnerabilidade o que afeta diretamente a sua segurança frente os aspectos de gerenciamento da nuvem (NIST, 2011). O acesso a recursos podem ser garantidos com práticas que podem aumentar a segurança, como o uso de senhas fortes, *tokens*, cartões de segurança e biometria. A identidade digital um método eficiente de se prover segurança, podendo ser usada não só na nuvem, mas também na rede local o que torna a segurança de dados flexível em um ambiente de TI (VACCA, 2016).

3.1. Problemas Importantes de Segurança e Privacidade

Devido ao aumento da utilização da computação em nuvem, muitos problemas foram surgindo em decorrência desse aumento. A computação em nuvem pública representa uma mudança no paradigma das normas convencionais de uma infraestrutura, pois move tudo que era concentrado fisicamente em um ambiente conhecido, para uma infraestrutura de uma organização distante, onde aplicações de diferentes clientes também podem operar.

Governança

A governança implica no controle e supervisão pela organização em relação a políticas, procedimentos e padrões. Com a ampla disponibilidade de serviços de computação em nuvem, a falta de controle sobre funcionários pode ser um problema, embora a computação em nuvem simplifique a aquisição da plataforma, não deixa de exigir

menos governança sobre os sistemas. Se ações de governança não forem tomadas, as políticas e procedimentos para privacidade, segurança e supervisão podem não ter a devida atenção, podendo assim, sistemas vulneráveis serem implantados sem consentimento, gerando um custo acima do normal, sublocando recursos para fins diversos que não são autorizados, o que pode prejudicar ou criar problemas.

Segurança dos Dados

A criptografia é o ponto chave da segurança dos dados, usando de métodos matemáticos, a criptografia mascara as informações e só são reveladas às partes interessadas com o uso de uma chave de descriptografia possuída pela parte autorizada. A criptografia impede que uma pessoa ou parte que não esteja autorizada a ver o conteúdo, tanto quando os dados estão em trânsito ou armazenados na nuvem (VACCA, 2016).

O controle de acesso pode se tornar um problema devido ao modo que a nuvem oferece acesso aos dados nela contida (BUYYA, BROBERG e GOSCINSKI, 2011), com a perda da confidencialidade, integridade ou disponibilidade os clientes devem entender a extensão da proteção que a nuvem oferece para que possam compreender o risco racional, dependendo do nível de impacto, baixo, médio ou alto a nuvem pode oferecer uma determinada proteção aos dados e sistemas de acordo com estes níveis.

As normas da *FIPS (Federal Information Processing Standard)* ditam a adequação da nuvem para armazenar e processar os dados dos clientes variando no nível de impacto e as garantias de proteção para estes dados. Essa capacidade de proteger os dados varia de acordo como são acessados os dados, podendo ser em trânsito para um provedor ou de algum provedor onde os dados do cliente devem estar protegidos, da mesma forma se um cliente quiser baixar estes dados, os mesmos devem estar protegidos na transferência (*NIST*, 2010).

Hashing

Hashing é utilizado quando necessita de proteção unilateral e não-reversível, isto é, quando um hash é gerado, fica disponível apenas para uma chave, e nenhuma outra chave é gerada para realizar o desbloqueio. Um exemplo de utilização é para a armazenagem de senhas. Além disso pode ser utilizado para proteger os dados na nuvem, incluindo usuários não autorizados e maliciosos e não-autorização de acesso (ERL, MAHMOOD E PUTINI, 2013).

Segurança de Identidades

A segurança de identidades é a chave para a segurança flexível dos dados em um ambiente de nuvem (BUYYA, BROBERG e GOSCINSKI, 2011). Uma assinatura digital deve assegurar a autenticidade de uma mensagem e o não repúdio, ou seja, deve dificultar uma parte a forjar uma assinatura digital válida e utilizá-la em mensagens diferentes (VACCA, 2016).

A criptografia assimétrica é comumente utilizada para esse tipo de segurança em ambientes de nuvem, então se possuir a chave pública gerada e quiser a assinatura digital confiável enviada pela rede, é preciso solicitar a outra ponta que assine ou criptografe estes dados, assim, se for possível descriptografar os dados, é possível saber que estes dados foram criptografados pela parte autorizada, pois só ela conhece a chave privada (VACCA, 2016).

3.2. Princípios da Segurança

Em um ambiente de TI convencional é comum observar o detimento da segurança visando o bom funcionamento dos sistemas. Tal fato, cria um ambiente desprotegido e suscetível a ataques externos. Porém, quando os ativos se encontram na mão de outras pessoas, a segurança deve ser tratada com criticidade (KANDUKURI, 2009).

Os sistemas computacionais nunca estarão livres de terem vulnerabilidades, pois foram projetados e testados por humanos, e estes sempre cometem erros, por exemplo, a construção de um *hardware*, a criação de um *software* sempre haverá uma falha, mais conhecida por vulnerabilidade, ou fraqueza. Essas fraquezas podem ser exploradas por invasores, geralmente tentando obter informações e dados sobre sistemas de empresas (VACCA, 2016).

De modo geral, a computação em nuvem oferece um maior ganho com a sua implementação, reduzindo os custos e aumentando a produtividade das empresas, com um acesso amplo a rede, agrupamento de recursos e serviços tornou a computação em nuvem popular na TI. Além desses pontos, a segurança é crucial para a computação em nuvem. A segurança na computação em nuvem não é diferente da TI convencional, no entanto, com os diferentes modelos de implantação apresentam riscos diferentes (GURKOK, 2014).

Seguindo o contexto de comprometer os princípios de segurança na nuvem, alguns pontos críticos encontrados na nuvem não são encontrados na computação convencional, fazendo com que a computação em nuvem possa ser vista como um obstáculo ou dificultador na TI.

Complexidade dos Sistemas

Um ambiente de computação em nuvem pública é uma plataforma mais complexa que uma TI convencional, devido a seus recursos e componentes que resulta em uma grande superfície de ataque. A segurança não depende apenas de correções e eficácia dos componentes, já que a inclusão de novos e mais recursos a nuvem, aumentam a sua complexidade, o que aumenta uma demanda de manutenções e gerenciamento, o que geralmente se relaciona inversamente com a segurança, dando uma maior vulnerabilidade.

Ambiente Compartilhado

O compartilhamento de nuvem entre multiusuários pode ser usada como uma brecha de

segurança, uma vez que a nuvem coloca uma dependência maior da separação lógica em várias camadas da pilha de aplicativos em vez de uma separação física dos recursos. Um cliente pode explorar as vulnerabilidades dentro da nuvem, ultrapassar as barreiras de separação e obter acesso a dados restritos, o que também pode ser expor os dados a outros clientes da nuvem ou bloquear o acesso a clientes que antes eram autorizados.

Serviços Orientados para a Internet

O serviço em nuvem pública são distribuídos pela internet, o que expõe as interfaces utilizadas pra realizar os acesso a nuvem, agora, os serviços que antes estavam protegidos dentro do *firewall* da empresa, agora estão em uma nuvem pública enfrentando o risco de ameaças de rede que visam roubar os dados. O desempenho e a qualidade via internet também podem ser afetados.

Perda de controle

Os serviços e dados que antes estava sob domínio da TI local, agora estão migrados para a nuvem, o que acarreta numa transferência de responsabilidades para o provedor e uma certa perda de controle sobre os recursos na nuvem contidos, a falta de um contato direto com o provedor de serviços, aumentam essa perda de controle. Esta situação torna a organização dependente da cooperação entre o provedor da nuvem em realizar atividades que abrangem as responsabilidades, como reportar o monitoramento e incidentes. A perda de controle implica em diminuição da capacidade da empresa em manter a consciência sobre a situação operacional, diante disso, a manutenção de responsabilidades pode ser mais desafiadora frente a TI local (JANSEN, GRANCE, 2011).

Princípios da Segurança	Risco	Questionamentos
Autenticidade	Verificação da autenticidade das entidades ativas.	Que ações são tomadas para autenticação e controle de acesso dos usuários?
Confidencialidade	Ativos de diversos usuários que dividem o mesmo sistema.	Como é realizada a segregação dos dados?
Disponibilidade	Recuperação de dados.	Como é garantida a recuperação dos dados?
Integridade	Violação dos dados e leis de proteção.	Quais as garantidas de preservação dos dados?
Não-repúdio	Análise das ações executadas por usuários dos ativos.	Os usuários são capazes de negarem suas ações?

Tabela 1. Princípios da Segurança

4. Questões de Segurança Relevantes

Para a realização de uma implantação em computação em nuvem ou migração de ativos físicos para um ambiente virtual, foram levantadas questões sobre a segurança e de que forma poderiam ser mitigadas e amortecidas no seu uso, métodos eficientes e seguros de acesso, recuperação de desastres, entre outras medidas. A tabela 2 abaixo, mostra questões levantadas pelo autor com base em pesquisas bibliográficas, dúvidas surgidas, o risco impactado na adoção e uma forma de mitigar as dúvidas recorrentes.

Questões	Dúvidas	Risco	Medida esperada
Formas de acesso	Como prover acesso seguro a nuvem?	Alto	Através de uma <i>VPN</i> (<i>Virtual Private Network</i>).
Recuperação de Dados	Como recuperar eventuais dados perdidos?	Alto	A nuvem deve prover plano de recuperação.
Controle de acesso	De que forma é realizado o controle de acesso?	Alto	Geração de <i>tokens</i> e identidades digitais para os clientes autorizados.
Backup dos ativos	Qual a frequência dos backups dos ativos?	Médio	A nuvem deve possuir <i>backups</i> das Máquinas Virtuais pelo menos uma vez ao mês.
Registro de controle	É possível ter registros sobre acessos e alterações?	Médio	Através de <i>LOGs</i> gerados, podendo ser solicitados esporadicamente pelo cliente ao provedor de nuvem.

Tabela 2. Requisitos de Segurança

5. Conclusão

Hoje a internet é usada para praticamente tudo, e como consequência disso, a Computação em Nuvem cresceu nos últimos anos e isso implicou em medidas de segurança mais robustas e eficientes para o meio. Os sistemas computacionais nunca estarão livres de terem vulnerabilidades, pois foram projetados e testados por humanos, e estes sempre cometem erros, por exemplo, a criação de um software sempre haverá uma falha ou fraqueza. Essas fraquezas podem ser exploradas e podem causar danos ou roubo de dados importantes de empresas que utilizam a nuvem. Por isso, não basta apenas ter um controle rigoroso de acesso, se durante o processo de transferência de dados ou armazenagem ocorrem falhas de segurança. É preciso pensar desde o acesso

até a forma que é feito a armazenagem e controle do provedor de nuvem. Para a empresa, não faz sentido ter uma forte segurança no acesso, com tokens, por exemplo e no provedor, os dados ficarem expostos a acesso de funcionários não autorizados do provedor de nuvem.

Referências

- Buyya, R., Broberg, J. e Goscinski, A. M. (2010) “Cloud Computing: Principles and Paradigms” 1. Ed. John Wiley & Sons, 660 páginas.
- NIST – National Institute os Standards and Technology(2011) “Chapter 14: Cloud Computing Securitu Essentials and Architeture”. Disponível em: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=919233
- Castro, R., C., C. e Sousa, V., L., P., “Segurança em Cloud Computing: Governança e Gerenciamento de Riscos de Segurança”, (2010), 7 páginas, Disponível em: [http://infobrasil.inf.br/userfiles/26-05-S5-1-68740-Seguranca%20em%20Cloud\(1\).pdf](http://infobrasil.inf.br/userfiles/26-05-S5-1-68740-Seguranca%20em%20Cloud(1).pdf)
- Erl, T., Mahmood, Z. e Puttini, R. “Coud Computing: Concepts, Technology & Architecture” (2013), Prentice Hall, Upper Saddle River, New Jersey, 528 páginas.
- Gurkok C., “Network and System Security. Chapter 4. Securing Cloud Computing Systems”. 2. Ed. Elsevier Inc.,(2014), Waltham, 58 páginas.
- Jansen, W.; Grance, T., “Guidelines on Security and Privacy in Public Cloud Computing”, (2011), NIST – National Institute os Standards and Technology, 80 páginas, Disponível em: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-144.pdf>
- Kandukuri, Balachandra Reddy, V, Ramakrishna Paturi, RAKSHIT, Dr. Atanu. (2009) “Cloud Security Issues. IEEE International Conference On Services Computing”, Pune, India, n., 520 páginas.
- Opus Software Comércio e Representações Ltda, “Computação em Nuvem”, (2015), 1. Ed. Opus Software, São Paulo.
- Rao, M. N., “Cloud Computing”, (2015) 1. Ed. PHI Learning Provate Limited, Delhi, 204 páginas.
- Reese, G. “Cloud Application Architetures”, (2009), 1. Ed. O'Reilly Media, 206 páginas.
- Vacca, J. R., “Cloud Computing Security Foundations and Challenges”, (2016), 1. Ed. CRC Press, Taylor & Francis Groupe, Boca Raton, 496 páginas.
- Velte, A. T., Velte, T. J., Elsenpeter, R., “Computação em Nuvem: Uma abordagem Prática”, (2012), Alta Books Editora, Rio de Janeiro, 352 páginas.
- Veras, M., “Computação em Nuvem: Nova Arquitetura de TI”, (2012), 1. Ed. Editora Brasport, São Paulo, 215 páginas.

Elaboração de Modelo para Proteção de Dados em Discos Virtuais usando os Esquemas IBE e ICP

Paulo Renato de Moraes Vieira, Luciano Ignaczak

Escola Politécnica – Universidade do Vale do Rio dos Sinos (UNISINOS)

São Leopoldo – RS – Brasil

paulo.renato@terra.com.br, lignaczak@unisinos.br

Abstract. *The new data storage' structure provided by virtual disk services has allowed users to develop different methods for sharing files over the internet. However, the data security level available on these services is limited by definition, requiring an option to ensure the protection of the files on the cloud, like data cryptography. This paper introduces a new model of cryptography on such scenario, converging the asymmetric schemas IBE and ICP, hence establishing the data confidentiality based on users' identity and the authentication using digital certificates.*

Resumo. *Os serviços de discos virtuais proporcionaram novas estruturas de armazenamento de dados, alterando a forma como as pessoas passaram a compartilhar arquivos por meio da internet. Entretanto, por definição, esses serviços apresentam limitações em relação à capacidade de garantir a segurança da informação, exigindo métodos de controle que assegurem a proteção dos arquivos na nuvem, como o uso de criptografia de dados. Este artigo propõe um modelo de criptografia voltado a esse cenário, associando a estrutura dos esquemas assimétricos IBE e ICP de proteção, garantindo assim, a confidencialidade dos dados com base na identidade dos usuários e a autenticação através de certificados digitais.*

1. Introdução

A estrutura da sociedade em rede, termo que caracteriza o comportamento atual das pessoas está lastreada em definições como as publicadas pelo [NIST (2011)] e em citações como a de [Newcombe (2012)], que apresentam a tecnologia de computação em nuvem (*cloud computing*) disponibilizando um alto nível de computação com reduzido custo de investimento e novas estruturas de produtos e serviços de Tecnologia da Informação (TI) no mercado. Dentre as soluções, os discos virtuais (*cloud storages*) são aqueles que permitem aos usuários fazerem *download*, *upload* e compartilhamento de arquivos pela *internet*, sendo uma opção enxuta para armazenamento de dados, devido ao custo de investimento reduzido frente à estrutura proporcionada pelos Provedores de Serviços na Nuvem (PSN). Alguns exemplos de mercado são o Google Drive, Amazon S3, Microsoft OneDrive e o Dropbox, sendo que o último conta com uma carteira de 500 milhões de clientes em todo o mundo, refletindo aproximadamente em 3.3 bilhões de conexões de compartilhamento de arquivos, segundo matéria do site [UOL (2016)].

No entanto, conforme texto publicado na revista [ZDNET (2015)], utilizando técnicas particulares de exploração das vulnerabilidades de sincronia desses serviços, uma pessoa mal intencionada é capaz de acessar os arquivos armazenados na nuvem mesmo sem ter conhecimento das credenciais do usuário, ratificando a afirmação de

[Newcombe (2012)], que aponta fatores de comprometimento e de riscos à informação devido a essa flexibilidade dos serviços sob demanda, requerendo uma atenção maior ao processo de segurança da informação às soluções dispostas na rede.

Nesse contexto, a criptografia é identificada como a solução ideal na busca por métodos para promover a confidencialidade dos dados, podendo passar a percepção de segurança à informação, situação capaz de justificar a alta demanda, nos últimos anos, de usuários em busca de opções de criptografia, segundo [Martin (2008)]. Das opções disponíveis no mercado, a Infraestrutura de Chaves Públicas (ICP) é o esquema criptográfico que promove métodos avançados de autenticação da identidade dos atores envolvidos na transação. A base do processo está na troca de chaves-públicas autenticadas previamente por uma entidade denominada Autoridade de Certificação (AC), resultando em estrutura de dados conhecida como certificados digitais. A utilização de certificados digitais permite a replicação da estrutura física de autenticação no ambiente eletrônico, conforme cita [Adams e Lloyd (2003)].

Uma das principais características da estrutura ICP de proteção é o alto custo computacional e de processos que esse esquema apresenta. Nesse contexto, segundo [Martin (2008)], o modelo criptográfico com base na identidade é aquele com maior simplicidade e praticidade de implementação, sendo essa talvez a principal característica, que justifica a rápida aceitação e adoção do referido padrão criptográfico. O *Identity-Based Encryption* (IBE) é um esquema assimétrico de proteção, que permite o cálculo da chave pública com base em um fator pseudorrandômico utilizado para identificar os atores da comunicação, fator que torna desnecessária a negociação prévia das chaves públicas ou utilização de um diretório de segurança para publicação das mesmas, como apresenta o padrão ICP. Dessa forma, é garantido aos atores um modo mais simples e prático de assegurar a confidencialidade da informação, mesmo ressaltando que é fundamental a integração com as bases de autenticação presente em outra solução para garantir que a aplicação sustente todos requisitos de segurança exigidos no cenário atual.

Das soluções de proteção presentes no mercado, aquelas voltadas para a segurança dos arquivos dispostos na nuvem, em sua grande maioria, estão apoiadas no uso de senhas ou estruturas complexas com base no esquema ICP de criptografia. Diante desse cenário, este trabalho busca responder a seguinte questão: é possível apresentar uma solução alternativa de proteção, capaz de garantir a confidencialidade dos arquivos na nuvem, associando a praticidade do modelo IBE e o esquema ICP de proteção?

Para responder a essa pergunta, o objetivo deste artigo é propor um modelo criptográfico capaz de sincronizar a capacidade de garantir a confidencialidade dos dados do esquema IBE ao padrão de autenticação presente no esquema ICP de proteção. O modelo permitirá aos autores avaliar a viabilidade de implementação em um ambiente real, garantindo a proteção de arquivos armazenados em discos virtuais e possibilitando o compartilhamento de dados de forma segura.

A organização deste artigo está distribuída em quatro seções, a segunda apresenta os trabalhos de diferentes autores que se relacionam ao objeto deste artigo, buscando pontuar os conceitos, percepções e conclusões. A terceira seção descreve o modelo criptográfico proposto, fornecendo informações gerais acerca dos componentes presentes e as respectivas relações. Por fim, a seção quatro apresenta as considerações finais, indicando os pontos de relevância ao projeto e proposta de segurança apresentada neste documento.

2. Trabalhos Relacionados

São apresentadas, nesta seção, visões de diferentes autores quanto aos principais tópicos relacionados e trabalhados ao longo desse projeto, buscando pontuar os conceitos, percepções e conclusões que facilitarão na identificação dos diferenciais presentes na solução projetada. Para tanto, o trabalho de pesquisa focou na busca por artigos em bases de dados acadêmicas, como a ACM, IEEE e Springer Link.

Dos desafios de segurança da computação em nuvem, a grande maioria das vulnerabilidades e ameaças elencadas pelos autores estão ligadas às características intrínsecas da tecnologia ou à implementação, em especial aos fatores da terceirização do controle e compartilhamento de recursos (estrutura “multi-inquilinos”). Para [Bouayad et al. (2012)] é preciso que haja uma visão holística do processo de proteção, garantindo que cada camada do serviço (da física à lógica) esteja devidamente avaliada e controlada, sob padrões de segurança que melhor representam os diferentes modelos presentes no ambiente, os quais contemplam diferentes atores envolvidos no cenário. Com o estudo dirigido à estrutura dos PSNs, [GroBauer et al. (2010)] propõem uma estrutura de referência dos principais componentes que orbitam os serviços na nuvem, permitindo assim uma visão mais ampla das vulnerabilidades e ameaças do ambiente. No trabalho de [Prakash e Dasgupta (2016)] estão descritos 9 dos principais pontos de vulnerabilidade do ambiente em nuvem considerados pela Aliança de Segurança da Computação em Nuvem (CSA), bem buscam apontar métodos de atender a essa demanda. Dentre os pontos citados, é possível constatar que o nível de segurança dos métodos de autenticação que é usualmente baixo, do tipo usuário e senha, demandando que haja maior atenção nesse quesito.

Quanto aos trabalhos que referenciam a estrutura de segurança do esquema IBE de proteção, [Lee (2010)] descreve uma solução híbrida que faz uso das tecnologias ICP e IBE, denominada Unified Public Key Infrastructure (UPKI). Na descrição desta solução, o autor propõe a criação de uma entidade que contemple os papéis de uma CA e uma PKG no ambiente, denominada Key Generation and Certification Authority (KGCA), além de agentes responsáveis pela recuperação da respectiva chave privada do esquema IBE, denominados Key Privacy Agents (KPAs). O sistema proposto [Sharma e Joshi (2016)] trata dos paradigmas de revogação das chaves utilizadas para criptografia de arquivos, sincronizando as características do esquema IBE com aquele que tem a base em atributos aleatórios, conhecido como ABE. Além da PKG, nesse ambiente há uma entidade denominada KU-CSP, que pode ser gerenciada pelo provedor da nuvem e é responsável por parte importante do processo de revogação das chaves. Propondo um método avançado de troca de chaves, [Júnior et al (2004)] apresentam solução alternativa à associação disjuntiva descrita por [Boneh e Franklin (2003)] na descrição original do esquema IBE.

Para proteção de arquivos na nuvem, desde acesso, armazenamento e troca, [Kumar e Singh (2015)] propõem um protocolo de autenticação dos usuários com base em criptografia utilizando uma estrutura de cálculo baseada em curvas elípticas, além de uma senha, do tipo One Time Password (OTP), gerado por um webserver na nuvem. Já no trabalho publicado por [Wei, Liu e Hu (2013)], os autores utilizam a estrutura IBE de proteção para proteger arquivos na nuvem, porém contam com a presença de uma terceira entidade responsável pela revogação da chave, denominada proxy re-encryption.

O modelo proposto neste artigo garante a confidencialidade da informação e autenticação dos usuários na troca e armazenamento de dados na nuvem, integrando o uso de certificados digitais ao processo de proteção com base no esquema IBE. Há similaridade com a proposta e estudos descritos em [Lee (2010)], porém contando com entidades distintas pelas ações de AC e PKG. A ideia central é utilizar as chaves IBE somente para garantir a confidencialidade dos dados e se apoiar na estrutura ICP presente no mercado para garantir os requisitos de autenticação e comprometimento legal das ações presentes nesse serviço, sendo um modelo avançado frente a processos como o apresentado por [Kumar e Singh (2015)].

3. Modelo Proposto

O objetivo central do modelo proposto neste artigo é apresentar um método de garantir a confidencialidade dos arquivos armazenados na nuvem. Conforme descrito na Figura 1, o modelo está estruturado na presença de cinco elementos, segmentados em locais e remotos, sendo os locais aqueles que iniciam e finalizam os processos, o Remetente e o Destinatário da informação. Fica a cargo dos elementos remotos sustentar as estruturas de armazenamento de dados na nuvem (Disco Virtual), recuperar as chaves de segurança utilizadas nos processos de criptografia e decriptografia (PKG), além de garantir a confiabilidade no processo de autenticação da identidade (AC).

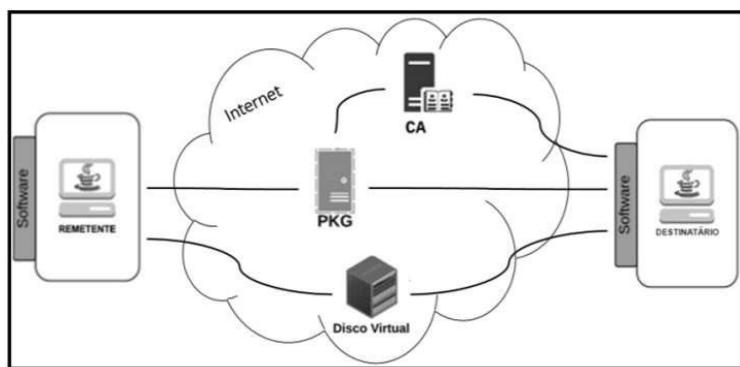


Figura 1 - Componentes e relacionamentos presentes no modelo criptográfico

Dos elementos ilustrados na Figura 1, o usuário “Remetente” é aquele que dá início ao processo de criptografia, buscando por um método de compartilhamento seguro de arquivos na nuvem, com controle de acesso aos dados mesmo que nunca tenha havido contato entre os envolvidos, independente do serviço de disco virtual utilizado. Serviço web de armazenamento de dados, o “Disco Virtual” é utilizado como ponto de distribuição online de arquivos, promovendo a disponibilidade da informação aos métodos de compartilhamento de dados utilizado pelos usuários. Os usuários podem enviar e baixar arquivos de qualquer lugar desde que haja conexão com a internet, através de páginas ou por aplicativos, *desktop* ou de dispositivos móveis. Entidade responsável pela emissão do certificado digital utilizado pelo Destinatário para autenticação, além de responder às requisições online de verificação do status do certificado, a “AC” pode fazer parte de uma estrutura ampla de confiança, e estar disponível na internet, ou de perímetro reduzido, de acesso limitado à rede local, conforme descrito na Figura 1. A “PKG” é responsável pela custódia das chaves de segurança dos usuários e validação da identidade do destinatário, atuando como ponto crucial tanto no processo de criptografia quanto de decriptografia dos arquivos. Para funcionamento pleno do sistema, é primordial que esse

componente esteja disponível na internet. Elemento final do fluxo do modelo proposto, o “Destinatário” é o usuário com o qual se deseja estabelecer a troca segura de arquivos na nuvem, sendo utilizado o respectivo email como informação chave para proteção dos arquivos e controle de acesso aos dados.

O método de criptografia escolhido para este modelo contempla a utilização de uma informação de domínio público, de fácil recuperação e diretamente ligado à identidade dos usuários. A informação elencada nesta proposta foi o endereço de correio eletrônico, permitindo que os usuários possam trocar dados de forma confidencial mesmo que nunca tenham estabelecido um canal seguro de compartilhamento anteriormente. O Remetente envia uma solicitação online à PKG, indicando no conteúdo da requisição o email daquele com quem se quer compartilhar a informação na nuvem. O processamento desses dados pela PKG irá gerar um código enviado na resposta ao Remetente, permitindo que o *software* local do usuário criptografe o conteúdo do arquivo utilizado, pronto para envio e armazenamento na nuvem através dos serviços de discos virtuais. Não há comunicação com a AC nas operações de criptografia de arquivos, uma vez que o usuário que inicia o processo é a origem, bem como não é foco neste modelo garantir a autenticidade e o não repúdio da informação.

Aspecto muito importante presente neste modelo é o bloqueio do acesso à informação original por usuários não autorizados. Isso ocorre no processo de decriptografia do arquivo, no qual o usuário Destinatário precisa autenticar-se junto à PKG, utilizando certificado digital válido e emitido junto a uma AC presente na raiz de confiança da PKG, para obter acesso à chave privada correspondente ao email utilizado no processo de criptografia. Somente após essa validação a PKG irá retornar os valores necessários para que o software local recupere a chave de proteção utilizada para criptografar o arquivo, permitindo assim acesso ao conteúdo original da informação.

Importante ressaltar que é pré-requisito para o sucesso deste modelo, a presença de certificado digital válido para autenticação da identidade do Destinatário. O esquema proposto fornece um amplo suporte para customizações acerca da área de atuação das entidades AC e PKG, de modo que estas atendam da melhor forma possível às necessidades do ambiente, exigindo apenas que haja um alinhamento prévio entre esses componentes.

4. Considerações Finais

A estrutura apresentada neste documento torna viável a convergência do que há de melhor nos esquemas IBE e ICP de criptografia, assegurando o sigilo da informação a um custo reduzido de complexidade operacional. Indo além, o projeto fornece suporte para resposta aos desafios relacionados aos fatores de retenção e gerenciamento do número extensivo de chaves criptográficas, demonstrando uma capacidade atemporal de decriptografia, não importando a versão das chaves assimétricas associadas aos certificados dos usuários.

O padrão criptográfico gerado após a associação de dois esquemas assimétrico de proteção, permite afirmar que a proposta atende aos requisitos de confidencialidade da informação, conforme descrito nos padrões internacionais de segurança ISO 27001 e 27002, fornecendo suporte necessário para elevar o nível de proteção dos dados armazenados em discos virtuais, enquanto garante o controle de acesso à informação somente para o usuário autorizado.

5. Referências

- Adams, C. e Lloyd, S. "Understanding pki. Boston", Massachusetts - EUA: Pearson Education, Inc, 2003.
- Boneh, D. e Franklin, M. "Identity-based encryption from the weil pairing". SIAM J. of Computing., [S.l.], v. 32, n. 3, p. 586–615, 2003.
- Bouayad, A. et al. "Cloud computing: security challenges". 2012 Colloquium in Information Science and Technology., [S.l.], p. =26–31, Out 2012.
- Grobauer, B. e Wallosche, T. e Stöcker, E. "Understanding cloud computing vulnerabilities". IEEE Security & Privacy., [S.l.], v. 2, n. 9, p. =50–57, Jun 2010.
- Júnior, W. D. B. e Terada, R. et al. "An ibe scheme to exchange authenticated secret keys". IACR Cryptology ePrint Archive 2004., [S.l.], p. =71, 2004.
- Kumar, V. e Singh, S. "Secured user's authentication and private data storage- access scheme in cloud computing using elliptic curve cryptography". 2th International Conference on Computing for Sustainable Global Development (INDIACoM)., [S.l.], p. 791–795, Mar 2015.
- Lee, B. "Unified public key infrastructure supporting both certificate-based and id-based cryptography". ARES'10 International Conference on Availability, Reliability and Security. [S.l.], p. 54–61, Fev 2010.
- Martin, L. "Introduction to identity-based encryption". Norwood, Massachusetts - EUA: Artech HouseIT, 2008.
- Newcombe, L. "Securing cloud services". Cambridgeshire, UK: IT Governance Publishing, 2012.
- NIST. "The nist definition of cloud computing (september 2011)". Disponível em: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf> . Acesso em: 1 maio 2016.
- Prakash, C. e Dasgupta, S. "Cloud computing secuity analysis: Challenges and possible solutions." Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on. IEEE, 2016.
- Sharma, R. e Joshi, B. "H-IBE: Hybrid-identity based encryption approach for cloud security with outsourced revocation." Signal Processing, Communication, Power and Embedded System (SCOPES), 2016 International Conference on. IEEE, 2016.
- UOL. "Dropbox comemora meio bilhão de usuários". Disponível em: <http://codigofonte.uol.com.br/noticias/dropbox-comemora-meio-bilhao-de-usuarios> . Acesso em: 30 maio 2016.
- Wei, J. e Liu, W. e Hu, X. "Secure data sharing in cloud computing using revocable-storage identity-based encryption". IEEE Transactions on Cloud Computing., [S.l.], p. 1–6, Ago 2013.
- ZDNET. "Attackers can access dropbox, google drive, onedrive files without a user's password". Disponível em: <http://www.zdnet.com/article/dropbox-google-drive-onedrive-files-man-cloud-attack> . Acesso em: 23 Junho 2016.

Randomização de endereço MAC como técnica para prover Privacidade a Usuários de redes WiFi

Bruno S. Alves¹, Yagor S. Duarte², Bolívar M. Silva¹

¹Centro de Tecnologia – Universidade Federal de Santa Maria (UFSM)
Caixa Postal 1000 – 97105-900 – Santa Maria – RS – Brazil

²Colégio Técnico Industrial – Universidade Federal de Santa Maria (UFSM)

bdalves@inf.ufsm.br, {bolivar, yagor}@redes.ufsm.br,

Abstract. *MAC (Media Access Control) address randomization is a technique commonly used to raise the level of privacy of devices, implemented in the vast majority of current operating systems. This hardware address remodeling acts to restrain third parties from analyzing the information traffic and linking them to a device, and even using them for tracking, that is, obtaining the physical location of the device and consequently the user. The way in which such randomization will be performed is exclusively at the discretion of the device manufacturers. The main idea of this article is to expose reasons why privacy measures are still necessary and why only MAC address randomization is necessary but not enough to guarantee the expected protection and anonymity.*

Resumo. *MAC (Media Access Control) address randomization é uma técnica comumente utilizada para elevar o nível de privacidade dos dispositivos, implementada na grande maioria dos sistemas operacionais atuais. Essa remodelação do endereço de hardware atua com intuito de coibir terceiros de analisar o tráfego de informações e vinculá-las a um dispositivo, e de até mesmo utilizá-los para rastreio, ou seja, obter a localização física do dispositivo e consequentemente do usuário. A maneira como tal randomização será realizada é de decisão exclusiva dos fabricantes dos dispositivos. A ideia principal deste artigo é expor motivos pelos quais ainda se fazem necessárias medidas de privacidade e por que somente à randomização do endereço MAC é necessária, porém não suficiente para garantir a proteção e anonimato esperados.*

1. Introdução

Com o crescimento do uso de smartphones, cada vez mais capazes de realizar com eficiência tarefas diárias, e até mesmo substituir os computadores pessoais em alguns casos, surgem novas possibilidades de inovações tecnológicas nas mais diversas áreas. De acordo com uma pesquisa realizada pela Folha de São Paulo [de Sao Paulo 2017] e levantamentos do IBGE (Instituto Brasileiro de Geografia e Estatística), o número de *Smartphones* no Brasil supera o número de habitantes. Praticamente todas as pessoas carregam consigo no mínimo um desses dispositivos. Tais fatos demonstram que a sociedade está cada vez mais ligada a dispositivos conectados em rede. Como consequência, o nível de privacidade do usuário está cada vez menor. Cada um desses dispositivos está equipado com diversos sensores (GPS, acelerômetros, sensor de proximidade, microfone, sensor de luminosidade, giroscópio, etc), que capturam e compartilham informações do usuário e

do ambiente, para diversos fins e aplicações [Ali and Khusro 2016]. Entre esses, está a interface *Wireless* que possibilita conexões WiFi (padrão IEEE 802.11). Essa interface é identificada por um endereço denominado MAC (*Media Access Control*).

O endereço MAC é um endereço físico composto por 12 dígitos hexadecimais (48 bits) utilizado para identificar as interfaces de rede de um dispositivo [Eastlake 3rd and Abley 2013]. Por se tratar de um endereço globalmente único, para rastreá-lo, basta que o interessado capture informações de rastros digitais deixados por esse dispositivo, para saber sua localização. Uma vez conhecida a localização do dispositivo, consequentemente, a localização do portador do mesmo também será conhecida. Essa técnica, passou a ser referida na literatura como *tracking* [Ali and Khusro 2016] [Patil and Kokil 2015].

O rastreamento de pessoas em ambientes públicos e privados representa uma ameaça a segurança e privacidade dos usuários. Uma vez que essas informações podem ser utilizadas para diversos fins, desde a identificação de mobilidade urbana em cidades, até a identificação de clientes em centros comerciais. De posse das informações a respeito das preferências dos clientes, o *marketing* das lojas, podem ser direcionados para estes usuários. Outro problema de privacidade que pode vir a surgir, está relacionado com o anonimato incorreto dos dados capturados. Por esses e outros motivos, empresas de tecnologia passaram a implementar a randomização do endereço MAC¹, com objetivo de dificultar o rastreamento de um dispositivo. Essa técnica consiste em gerar endereços MAC para as interfaces do dispositivo, de forma aleatória e periódica. Infelizmente, essas técnicas apresentam falhas, e na maioria das vezes, não são efetivas, conforme demonstrado em [Jeremy Martin 2017][Vanhoeft et al. 2016] [Martin et al. 2017].

Nesse contexto, o objetivo desse artigo é realizar um estudo sobre as técnicas de randomização adotadas pelos principais sistemas operacionais utilizados em dispositivos que oferecem suporte a conexão WiFi, focando principalmente nos sistemas utilizados em *Smartphones*. Para isso, será montado um cenário, com um *Access Point* e posteriormente testado, sondagem e conexão de alguns dispositivos de sistemas operacionais e fabricantes diferentes. Além disso, será apresentado alguns trabalhos relacionados, a fim de apresentar as pesquisas que estão sendo realizadas nessa área.

2. Trabalhos Relacionados

Diversos pesquisadores têm realizado estudos a respeito das vulnerabilidades em técnicas de randomização existentes. Além disso, algumas das pesquisas realizadas são voltadas para a exploração e utilização desses dados para uso nas mais variadas áreas de aplicação.

No trabalho de [Mathy Vanhoef 2016], são apresentadas formas de contornar a randomização dos endereços MAC, explorando dispositivos com suporte a função WPS (*Wi-Fi Protected Setup*). No trabalho é observado que durante a utilização da função WPS², algumas das informações fornecidas, representam uma falha no processo de mascara-

¹A randomização do endereço MAC é uma técnica que visa impossibilitar o rastreamento do dispositivo, através dos rastros deixados pelas requisições de sondagem. Uma vez que o dispositivo utiliza endereços randômicos durante o processo de sondagem, em teoria, o mesmo não pode ser rastreado através do endereço MAC anunciado em *broadcast* [Jeremy Martin 2017].

²O WPS permite que um dispositivo realize uma conexão com um ponto de acesso, sem a necessidade de inserção das credenciais necessárias.

mento de endereço. O uso do WPS, se baseia na utilização de um endereço UUID-E (*Universally Unique IDentifier-Enrollee*), que é derivado do endereço MAC do dispositivo. Dessa forma, é possível obter o endereço real, a partir do identificador UUID-E. Uma vez feito isso, a randomização de endereços MAC torna-se ineficaz.

Em [Jeremy Martin 2017], os autores apresentam as técnicas de randomização utilizadas por alguns dos Sistemas Operacionais mais populares da atualidade, bem como as falhas encontradas em cada técnica. Nesse trabalho, são apresentadas duas contribuições principais. A primeira contribuição, está relacionada a identificação de uma nova vulnerabilidade no processo de randomização, onde é explorada uma falha em alguns *chipsets wireless*. A segunda contribuição consiste na utilização conjunta de algumas técnicas já conhecidas, para possibilitar o rastreio de dispositivos apesar da randomização. Os autores concluem que, apesar dos dispositivos utilizarem técnicas de randomização, muitas vezes, não são totalmente efetivas, no que diz respeito a impedir o rastreamento dos dispositivos.

[Mathieu Cunche 2016], explora as vulnerabilidades deixadas com base nos outros campos do *frame* de *probe request*, tanto no cabeçalho, quanto no conteúdo do mesmo. Uma está relacionada ao número de sequência dos *probes request*³ emitidos, que não são reinicializados a cada randomização. Outra forma de rastreamento, seria através da obtenção de impressões digitais dos dispositivos, utilizando algumas informações como o IEs. Estes são blocos de dados que identificam os recursos suportados por uma estação. Além da análise de informações dos *frames*, é apresentado uma análise com base na frequência e no tempo com que as requisições são enviadas. Por fim, os autores concluem que apenas a randomização de endereços MAC não é capaz de garantir totalmente o anonimato pretendido. Além disso, citam a possibilidade da adoção futura de uma nova camada para a pilha de protocolos.

Tanto os trabalhos relacionados apresentados, como os analisados, apesar de apresentarem um foco ligeiramente diferente, têm em comum a opinião dos autores de que a randomização de endereços MAC não é efetiva em diversos casos, por motivos diversos. Nesse trabalho, pretende-se identificar, através de alguns testes, se os dispositivos Wi-Fi, cujos os SOs oferecem suporte, estão de fato implementando a randomização de endereços.

2.1. Métodos de Descoberta do endereço MAC verdadeiro

Utilizando determinados métodos, é possível identificar o dispositivo que se deseja coletar informações, descobrir seu endereço MAC original, e rastreá-lo. Um dos métodos, consiste em observar o *handoff* entre APs. Outro fator que pode ser levado em consideração, é a frequência de envio dos *probe requests*. Observados os padrões de requisições, é possível traçar perfis de comunicação dos dispositivos, e isolar o dispositivo alvo [Célestin Matte 2016].

Muitos dos APs atuais, possuem um protocolo chamado WPS (*WiFi Protected*

³*Probes request* são requisições enviadas quando um dispositivo encontra-se com a interface de rede Wi-Fi ativa, o mesmo envia mensagens em *broadcast* para qualquer outro dispositivo que esteja escutando. Estas mensagens de *probes request* têm como objetivo principal situar o dispositivo requisitante sobre os pontos de acessos ao seu alcance. Estas solicitações contém, entre outras coisas, o endereço MAC do mesmo.

Setup), que permite que dispositivos passem do estado de dissociado para associado sem que o usuário tenha que informar as credenciais. Para permitir isso, alguns campos extras são adicionados aos *probe requests*. Esses campos, além de conter o fabricante e o modelo do dispositivo, contém um identificador único utilizado para realizar uma conexão WPS. Esse identificador é derivado do endereço MAC, e a partir do conhecimento do identificador é possível reaver o MAC [Martin et al. 2017].

Outro método conhecido para a obtenção do endereço MAC real, é um ataque denominado *Karma* [Vanhoeff et al. 2016] [Martin et al. 2017]. Ele baseia-se em configurar um roteador com o SSID (*Service Set Identifier*) idêntico a um já conhecido pelo dispositivo. É comum que os dispositivos se conectem automaticamente em redes nas quais já se conectaram anteriormente. Isso faz com que o dispositivo se conecte no AP, e anuncie seu endereço MAC verdadeiro.

O algoritmo utilizado em [Célestin Matte 2016] se aproveita do fato de que os dispositivos costumam enviar *frames* de mensagens com a mesma frequência de tempo. Dessa maneira, é possível agrupar *frames* de mensagens de um dispositivo, através da análise da diferença de tempo entre mensagens. Assim, mesmo se ao longo do tempo, o dispositivo tenha seu MAC randomizado, ainda assim é possível descobrir sua identidade. Esse algoritmo obteve, em média, 77,2% de sucesso, na identificação de *frames* provenientes do mesmo dispositivo.

3. Metodologia

Com intuito de testar se os métodos de randomização de endereço MAC efetivamente omitem o endereço MAC real das interfaces WiFi, foram realizados testes capturando os *probes request* emitidos por alguns dispositivos WiFi de Sistemas Operacionais diferentes. Os testes realizados foram pensados partindo do princípio de que os dispositivos, estando com sua interface de rede WiFi ativa, enviam requisições em busca de pontos de acesso próximos para uma possível conexão. Essas requisições não são destinadas a dispositivos específicos, mas sim em *broadcast*. Dessa forma, qualquer dispositivo ao alcance, pode interceptá-las. Ao interceptar os *probes requests*, foram comparados os endereços MAC obtidos, com os endereços reais dos dispositivo.

Os Sistemas Operacionais (SO) abordados nos testes apresentam randomização do endereço MAC. Cada um utiliza uma abordagem própria, conforme descrito a seguir.

- *Microsoft Windows*: dispositivos com sistema operacional Windows, a partir da versão 10, dão suporte à randomização do endereço MAC. Porém, para que a randomização seja aplicada, o dispositivo deve também possuir um hardware com suporte para tal procedimento. Até o momento, o Windows 10, é o único SO que oferece randomização para dispositivos associados a uma rede WiFi [Mathy Vanhoef 2016].
- *Android*: o Sistema Operacional Android oferece suporte à randomização de endereços MACs, para requisições de sondagens, a partir de sua versão 6.0. Para utilizar a randomização, é importante que o dispositivo ofereça suporte de hardware e drivers necessários. Algumas aplicações com suporte a alteração manual ou randômica de endereços MACs estão disponíveis para Android, porém é necessário privilégio de aceso de root ao dispositivo [Mathy Vanhoef 2016].

- iOS: este foi o primeiro SO a oferecer a randomização de endereço MAC para ser utilizado durante o processo de busca por APs ao alcance (sondagem). A partir da versão 8, os dispositivos iOS da *Apple* passaram a utilizar a randomização. Assim como no SO Android, abordado anteriormente, os endereços randômicos nesse SO, são utilizados apenas quando o dispositivo está fazendo a sondagem por redes sem fio ao alcance. Após a conexão com o AP, passa a ser utilizado o endereço real do dispositivo [Technologies 2015] [Jeremy Martin 2017].

3.1. Cenário de Testes

Os testes foram realizados na UFSM (Universidade Federal de Santa Maria), no prédio do CTISM (Colégio Técnico Industrial), em um ambiente sem qualquer modificação, ou seja, sujeito a interferências de sinais provindos tanto de outros dispositivos, como de outros pontos de acesso próximos.

Utilizou-se um roteador com o *Firmware OpenWRT* instalado, configurado em modo monitor, onde foram implementados *scripts* para obtenção dos *probes requests* emitidos pelos dispositivos. O *Script* utilizado na captura dos *probes* foi desenvolvido na linguagem *Shell Script* e utiliza as funcionalidades do *software TCPdump* para isso. Dentre os dados capturados estão o horário de captura, a intensidade do sinal e o endereço MAC do dispositivo.

Foram selecionados quatro dispositivos para a realização dos testes. Dentre eles um dispositivo com SO Windows versão 10, dois com o SO Android versão 5 e 7, e por fim um dispositivo com SO iOS versão 10. Inicialmente, foram identificados os endereços MACs originais de cada um dos dispositivos, disponíveis nas informações de configuração do sistema. Os primeiros testes foram realizados, com os dispositivos já conectados em um AP da rede do CTISM.

Foi definido um intervalo de cinco minutos para análise do comportamento do dispositivo, captando e analisando os *probe requests*. Após a coleta dos dados com dispositivo conectado, na segunda etapa, foram realizados testes com os dispositivos desconectados de qualquer rede WiFi, porém ainda com a interface ativa.

4. Resultados e Conclusão

A partir da coleta de dados, foi possível observar que o dispositivo com o Android 5, conforme esperado (por não possuir suporte à randomização), não apresentou uma randomização do seu endereço em questão. Foi possível identificá-lo através do endereço MAC, tanto conectado, quanto desconectado. Vale comparar a coleta de dados tanto de dispositivos que, em teoria, fornecem a possibilidade de randomização, com dispositivos que não possuem suporte. Dessa forma, pode-se observar as diferenças que existem em ambos, na prática.

O dispositivo com Android 7 e o dispositivo com Windows 10, apesar de possuírem suporte a randomização, não apresentaram nenhum tipo de alteração em seus endereços MACs reais. O mesmo endereço foi distribuído durante as requisições quando conectado ao AP e durante a realização de sondagem. Para que a randomização ocorra, é necessário estar implementada no software e no hardware do dispositivo. Embora ambos os SOs possuam suporte à randomização, possivelmente, o hardware de ambos não seja compatível com a funcionalidade.

Dos sistemas operacionais testados, o único que se mostrou capaz de realizar a randomização, conforme prometido pelo fabricante, foi o iOS 10. De acordo com os testes, apenas quando o dispositivo estava conectado, que o seu MAC verdadeiro foi divulgado.

Com a análise dos resultados dos testes obtidos, foi possível perceber que o dispositivo com iOS 10 não pode ser identificado por meio do mesmo endereço que distribuía quando desconectado. Já o de Android 5 apresenta um grande número de divulgações do MAC verdadeiro, o que aumenta a precisão da localização de um dispositivo, utilizando a potência do sinal das requisições. Apesar de apresentar uma diferença significativa no número de divulgações, a versão do Android 7, ainda apresenta falhas quanto à randomização, ou seja, continuou a divulgar seu endereço verdadeiro.

A privacidade do usuário é um fator que deve receber atenção das empresas desenvolvedoras de Sistema Operacional. Apesar dos esforços para prover maior privacidade aos usuários, os resultados mostram que a randomização do endereço MAC, aparentemente efetiva em teoria, ainda não é adotada de forma plena por todos os SOs, na prática.

Referências

- Ali, S. and Khusro, S. (2016). Mobile phone sensing: A new application paradigm. *Indian Journal of Science and Technology*, 9(19).
- Célestin Matte, Mathieu Cunche, F. R. M. V. (2016). Defeating mac address randomization through timing attacks.
- de Sao Paulo, F. (2017). Número de smartphones em uso no brasil chega a 168 milhões. <http://www1.folha.uol.com.br/mercado/2016/04/1761310-numero-de-smartphones-em-uso-no-brasil-chega-a-168-milhoes-diz-estudo.shtml>. May 24, 2017.
- Eastlake 3rd, D. and Abley, J. (2013). Iana considerations and ietf protocol and documentation usage for ieee 802 parameters. Technical report.
- Jeremy Martin, Travis Mayberry, C. D. L. F. L. B. C. R. E. C. R. D. B. (2017). A study of mac address randomization in mobile devices and when it fails.
- Martin, J., Mayberry, T., Donahue, C., Foppe, L., Brown, L., Riggins, C., Rye, E. C., and Brown, D. (2017). A study of mac address randomization in mobile devices and when it fails. *arXiv preprint arXiv:1703.02874*.
- Mathieu Cunche, C. M. (2016). On wi-fi tracking and the pitfalls of mac address randomization.
- Mathy Vanhoef, Célestin Matte, M. C. L. S. C. F. P. (2016). Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms.
- Patil, P. and Kokil, A. (2015). Wifipi-tracking at mass events. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–4. IEEE.
- Technologies, Z. (2015). Analysis of ios 8 mac randomization on locationing.
- Vanhoef, M., Matte, C., Cunche, M., Cardoso, L. S., and Piessens, F. (2016). Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 413–424. ACM.

Mecanismo de Autenticação de Dispositivos para Internet das Coisas

Jonathan Monteiro Araujo , André Peres

¹Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Porto Alegre

jonathanmaraujo@gmail.com, andreperes@poa.ifrs.edu.br

Abstract. Considering the growth in the internet of things solutions and the number of security vulnerabilities in these solutions, it becomes evident the increased need of security mechanisms for these devices. Implementation failures in one of the main attributes of security, the authentication, is one of the exploited spots that make these devices perform malicious actions. This paper describes the development of an authentication mechanism whose aim is to be used in the devices authentication and communication process in devices, therefore improving their security. The Arduino platform was used in this project to implement and validate the mechanism. As result of this project and the studies carried out it was obtained .a mechanism was developed, composed by: a device register system, an authentication API and a solution able to realize the device authentication in the server.

Resumo. Considerando a grande e crescente oferta de soluções para a internet das coisas e o número de falhas de segurança envolvendo tais soluções, fica evidente a necessidade de incremento de mecanismos para a implementação de segurança nesses dispositivos. Falhas na implementação de um dos principais atributos de segurança, a autenticação, é um dos pontos explorados para que estes dispositivos aceitem realizar ações maliciosas. Este trabalho descreve o desenvolvimento de um mecanismo cuja proposta é a de ser utilizado no processo de autenticação e comunicação de dispositivos, oferecendo assim um pouco mais de segurança aos mesmos. No trabalho foi utilizada a plataforma arduino para a implementação e validação do mecanismo. Como resultado deste projeto e dos estudos realizados foi desenvolvido um mecanismo composto por: uma aplicação de cadastro de dispositivos, uma API de autenticação e uma solução capaz de realizar a autenticação do dispositivo no servidor.

1. Introdução

Atualmente percebe-se que a tecnologia está cada vez mais integrada na vida da população. Um dos aspectos mais notáveis dessa integração pode ser percebido no uso frequente de soluções baseadas em *IoT* (*Internet of Things*). Segundo a *Gartner*, empresa referência no mercado de tecnologia desde 1979, no ano de 2016 o número de dispositivos baseados em *IoT* chegará a 6,4 bilhões, o que corresponde a um aumento de 30% em relação a 2015, com a expectativa de atingir a marca de 20,8 bilhões até 2020 [Gartner 2015]. Desde *smarthomes* (casas inteligentes) até os *wearables* (tecnologias vestíveis), a *IoT* vem crescendo e se integrando ao cotidiano.

No contexto de internet das coisas, uma "coisa" é definida como um objeto capaz de receber dados do ambiente através de sensores, e possivelmente alterar esse ambiente através de atuadores além de comunicar-se com outros objetos [Alecrim 2016]. Os dados recebidos pelo objeto precisam ser processados, sendo esse processamento realizado por um dispositivo controlador conectado à rede.

Com o rápido avanço da oferta de soluções *IoT* e o desejo de difusão dessa tecnologia por parte dos fabricantes, tem-se um cenário onde nota-se pouca preocupação por parte dos desenvolvedores em relação a configurações de soluções de segurança disponíveis para autenticação de dispositivos. Isto resulta em uma possível vulnerabilidade quando leva-se em conta que esses dispositivos podem ser responsáveis pelo gerenciamento de aspectos sensíveis de usuários [Barcena and Wueest 2015].

Este trabalho tem por objetivo o desenvolvimento de um mecanismo de autenticação entre dispositivos na *IoT*. Para o desenvolvimento deste mecanismo será utilizado um dispositivo comumente empregado na prototipação de soluções *IoT*, a plataforma Arduino.

Devido às limitações de processamento e memória desta plataforma, a hipótese deste trabalho parte da seguinte questão: seria possível empregar um mecanismo de autenticação a ser utilizado no desenvolvimento de soluções em *IoT* que utilizam Arduino?

2. Proposta

Visando o aumento da segurança em redes de dispositivos *IoT* e principalmente com o intuito de aplicação de regras de autenticação entre os dispositivos, foi desenvolvido um mecanismo capaz de identificar e autenticar tais dispositivos. Ao ser enviada uma nova transmissão o mecanismo é capaz de verificar em um servidor de autenticação se o dispositivo solicitante está autenticado na rede e, desta forma, apto a receber informações. A solução também deve ser utilizada pelo dispositivo que receberá as informações, pois o mesmo somente deverá ser capaz de receber transmissões de dispositivos autenticados na rede.

O público-alvo deste mecanismo são desenvolvedores que têm como objetivo a construção de soluções de *IoT*, e garantindo com o uso do mecanismo que seus dispositivos somente receberão dados de outros dispositivos autenticados, evitando dessa forma que suas soluções sejam comprometidas e utilizadas com outros fins.

A estrutura de comunicação entre os dispositivos da rede é composta por um servidor e seus dispositivos, o servidor é responsável pelo cadastro, autenticação e comunicação dos dispositivos.

Para o cadastro dos dispositivos foi implementada uma página web, onde é cadastrado o nome e o endereço de rede do dispositivo. Após o envio do formulário com as informações do dispositivo, são gerados: um número identificador do dispositivo, uma senha e uma chave de criptografia. Essas informações são armazenadas em dois locais: primeiramente no banco de dados do servidor e em seguida em um arquivo de texto que é, após a confirmação do cadastro, disponibilizado para download pelo usuário. O arquivo, denominado "auth.txt", precisa ser gravado em um cartão micro SD que por sua vez deve ser inserido no leitor de cartões de uma shield ethernet (placa para comunicação do Arduino), conforme ilustrado na figura 1(A).

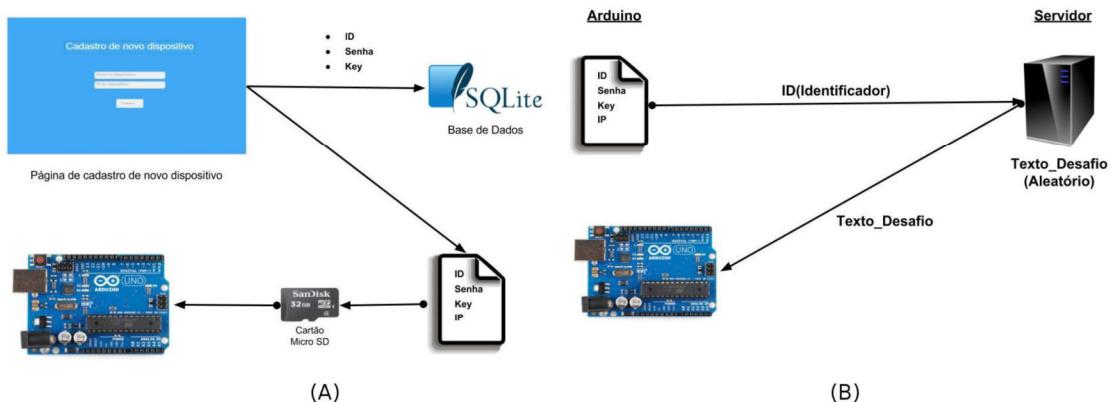
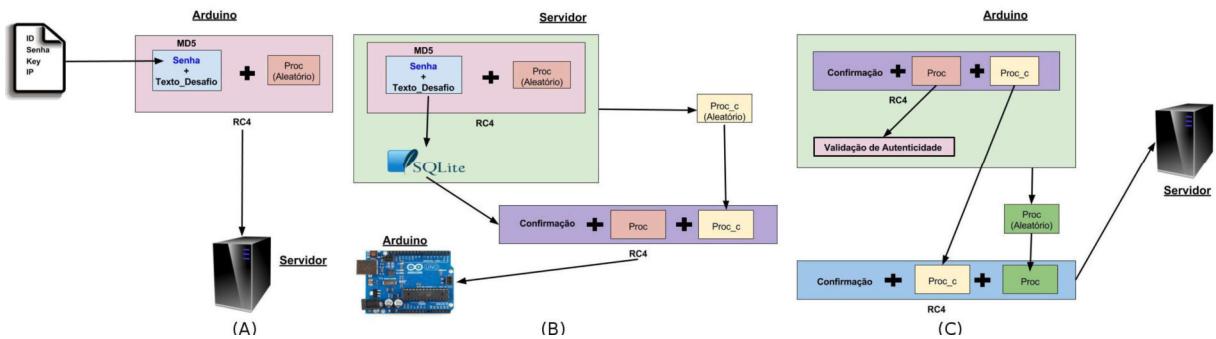


Figura 1. Cadastro de novo dispositivo

Após cadastrado no servidor toda a comunicação entre os dispositivos de uma rede irá passar pelo servidor de autenticação. Para que este possa transmitir informações de um membro da rede para outro de forma segura é indispensável o estabelecimento de um canal de comunicação criptografado. Para o estabelecimento desse canal é preciso que o dispositivo passe pelo seguinte processo de autenticação:

1. O arduino irá ler seu número de identificação no arquivo armazenado no cartão micro SD e enviá-lo para o servidor. Este irá gerar um texto aleatório (texto desafio) e enviá-lo para o arduino, conforme descrito na figura 1(B).
2. O arduino ao receber o texto irá concatená-lo com a senha, que está no arquivo "auth.txt", e calculará o *Hash MD5* do resultado dessa concatenação. Com o *Hash* calculado o dispositivo irá gerar um valor aleatório de 2 bytes denominado "proc". Este é adicionado ao final do *Hash*. O resultado desse processo é então cifrado, utilizando o algoritmo de chave simétrica RC4, com a chave de criptografia gerada no momento do cadastro do dispositivo e armazenada no cartão SD. Em seguida o texto cifrado é enviado para o servidor. Esse processo está demonstrado na figura 2(A).
3. O texto criptografado é decifrado no servidor e uma validação das informações é realizada com base nos registros da base de dados. Caso a validação ocorra com sucesso o dispositivo está autenticado. O servidor então irá enviar uma mensagem de confirmação cifrada contendo o "proc" gerado pelo arduino, seguido de um novo valor aleatório gerado pelo servidor denominado "proc_c", conforme mostrado na figura 2(B).
4. O dispositivo então decifra a mensagem de confirmação e valida o "proc". A partir dessa etapa ambos servidor e dispositivo estão autenticados entre si. A mensagem enviada em seguida contém a informação que deverá ser recebida pelo próximo arduino da rede e deverá conter o "proc_c" do servidor e um novo "proc" gerado pelo arduino. Dessa forma é estabelecido um canal criptografado de comunicação, e a cada nova troca de mensagens, entre o servidor e o arduino, os valores de "proc" e "proc_c" são alterados, conforme demonstrado na figura 2(C).

A implementação dos valores aleatórios "proc" e "proc_c" possuem como principal finalidade a prevenção de ataques do tipo *replay* ou *playback* [Dickson 2016]. Esse tipo de ataque consiste na interceptação e no reenvio de um ou mais pacotes válidos que não

**Figura 2. Autenticação**

pertencem ao atacante, ataques dessa natureza são muito comuns, haja vista que o atacante nem mesmo precisa ter acesso às chaves de criptografia utilizadas no tráfego.

3. Tecnologias Empregadas e Metodologia

Inicialmente para o desenvolvimento do mecanismo foi utilizado como base o arduino de modelo *UNO*, ou genuino UNO. Esse modelo de arduino havia sido escolhido devido ao fato de ser popular e acessível, considerando a sua disponibilidade e custo. Entretanto devido a limitações de hardware do modelo *UNO* viu-se a necessidade de migração para o arduino do modelo *MEGA 2560*, ou Genuino *MEGA 2560*.

Para o desenvolvimento deste mecanismo foi definida a utilização do Arduino Ethernet Shield V2, que dispõem de uma interface do tipo RJ45 para conexão com a rede através de um cabo padrão. Outra vantagem na utilização deste shield consiste na disponibilização por padrão de uma interface para cartão do tipo micro-SD, o qual é útil no mecanismo para armazenar informações empregadas no processo de autenticação.

Para cifrar e decifrar os textos requisitados na autenticação foi definido o uso do algoritmo RC4. Criado em 1967 por Ronald Rivest o algoritmo RC4 ou ARC4 consiste em um algoritmo de chave simétrica de cifra de fluxo [Stallings 2014], ou seja, o processo de cifragem e decifragem independem do tamanho da chave e as operações são orientadas a byte. O RC4 desempenha a tarefa fundamental de garantir a confidencialidade na transmissão das informações no mecanismo, e demonstrou um bom desempenho ao ser implementado na plataforma arduino, sendo executado rapidamente considerando as limitações da plataforma. Um problema enfrentado com a cifragem utilizando RC4 foi o uso pelo algoritmo de caracteres especiais, esse problema foi tratado através do uso de codificação *Base64* no texto já cifrado.

4. Cenário de Testes e Resultados Obtidos

O cenário de testes foi construído para validar o processo de autenticação e envio de uma mensagem entre dois dispositivos arduino. Neste cenário foram empregados os seguintes ativos: um atacante (executor do teste), dois dispositivos arduino e um servidor de autenticação.

Os testes foram realizados através da execução de um software na rede capaz de analisar o tráfego dos dados, neste teste foi usado o *Wireshark*. A figura 3 apresenta uma

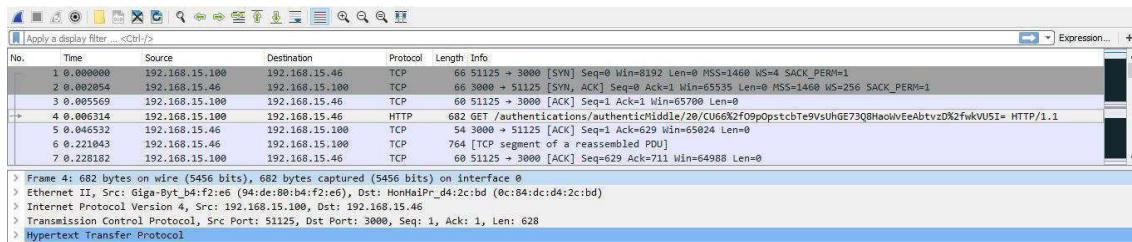


Figura 3. Exemplo de captura de pacote no software Wireshark.

captura de pacote do mecanismo, é possível ver que na requisição obtida é apresentado o texto cifrado assim como o identificador do dispositivo.

Para simular um dispositivo malicioso na rede que tenta se autenticar através do envio das requisições repetidas (ataque de *replay*) para o servidor de autenticação utilizou-se o software *Postman*. A figura 4 demonstra uma tentativa de autenticação que resultou em falha pois foi re-enviada para o *webservice* uma requisição, o erro ocorreu pois o "proc" fornecido já havia sido utilizado anteriormente.



Figura 4. Exemplo de requisição no software Postman.

A figura 5 apresenta os logs de monitoramento de um arduino, nele é possível ver os passos de um processo de autenticação bem sucedido.

```
COM6 (Arduino/Genuino Mega or Mega 2560)
|| Send
Authentication Example v1.0 release 30/06/2017
Local IP: 192.168.15.100
Send an 'g' in serial monitor to start authentication
Authentication Start Initiated
Sending request: GET /authentications/authenticStart/20 HTTP/1.1
Challenge received = b5b10971d2fffe6f
Authentication Start Finished

Authentication Middle Initiated
Sending request: GET /authentications/authenticMiddle/20/x9iQ9wTiwhrlat9Xz+4sKmGQ04pUU0w4CJz63QL09f3Dw== HTTP/1.1
Received: T2QhrAR%2f0WCpZMwRjRh73WFUlvRAWwsTcm7g==
Authentication Middle Finished

Authentication Finish Initiated
Message: Turn_On
Sending request: GET /authentications/authenticFinish/20/oleWmwu4wESLA5sMyhOu5fmVQRU= HTTP/1.1
Received Decrypted: Message received
Authentication Finish Finished

Awaiting new orders...
```

Figura 5. Processo de autenticação de dispositivo.

O projeto resultou em uma solução capaz de autenticar o envio e o recebimento de informações entre os dispositivos de uma rede, servindo como mediador das transmissões.

O mecanismo foi concebido de forma a ser utilizado por desenvolvedores de soluções IoT, fornecendo assim maior segurança em suas aplicações.

5. Conclusão

O desafio deste trabalho foi o de desenvolver um mecanismo de autenticação capaz de ampliar a segurança de soluções de IoT que utilizam a plataforma Arduino. A capacidade de processamento e a disponibilidade de memória do Arduino acabam por limitar o número de soluções eficazes para este mecanismo.

Considera-se que o mecanismo desenvolvido atinge os objetivos propostos e apresenta como contribuição uma possibilidade de uso e estudo para desenvolvedores de aplicações IoT.

Através dos estudos realizados e apresentados, conclui-se que a IoT é uma área em expansão e que logo a população irá depender do uso dessas soluções. Todavia a dependência dessas tecnologias para a realização de tarefas fundamentais torna a questão de garantia de segurança nestas soluções um assunto relevante e de suma importância, portanto novos estudos são necessários e novas soluções precisam ser desenvolvidas e difundidas.

Durante o desenvolvimento e a execução dos testes do mecanismo alguns pontos de melhoramento do software foram levantados, levando-se em consideração o desempenho, a qualidade do código desenvolvido e a visão de um produto final.

Considerando que este trabalho objetiva primariamente a garantia de autenticidade de dispositivos e todos os atributos componentes da segurança da informação, fica clara a necessidade de novas soluções que contemplem esses atributos. Para tal uma boa solução seria o desenvolvimento de um *framework* visando a disponibilização de métodos e mecanismos implementados com as boas práticas de segurança, dessa forma os desenvolvedores de aplicações IoT poderão, de certa forma, abstrair a segurança e concentrar seus esforços no desenvolvimento da aplicação em si.

Referências

- Alecrim, E. (2016). O que é Internet das Coisas (Internet of Things)? [Online: <http://www.infowester.com/iot.php> Acessado em 18/11/2016].
- Barcena, M. B. and Wueest, C. (2015). Insecurity in the Internet of Things. [Online:<https://www.symantec.com/content/dam/symantec/docs/white-papers/insecurity-in-the-internet-of-things-en.pdf>].
- Dickson, B. (2016). How to Prevent Replay Attacks on Your Website. [Online: <https://www.sitepoint.com/how-to-prevent-replay-attacks-on-your-website/> Acessado em 21/05/2017].
- Gartner (2015). Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015. [Online: <http://www.gartner.com/newsroom/id/3165317> Acessado em 10/11/2016].
- Stallings, W. (2014). *Criptografia e segurança de redes Princípios e práticas*, volume 4. PEARSON Prentice Hall.

Proposição de um Sistema de Autenticação Simplificado e Interativo com Dispositivo IoT

**Fabio Lopes Brezolin¹, Erciles Andrei Bellei¹, Jucélia Giacomelli Beux¹,
Marco A. Sandini Trentin¹, Angelo Elias Dalzotto², Joao Mário L. Brezolin³**

¹ Programa de Pós-Graduação em Computação Aplicada (PPGCA)
Universidade de Passo Fundo – Caixa Postal 611 – 99.001-970 – Passo Fundo – RS

² Curso de Engenharia de Computação – Universidade de Passo Fundo

³ Instituto Federal Sul-rio-grandense – Passo Fundo – RS

{71856, 168729, 68428, trentin, 150633}@upf.br,
joao.brezolin@passofundo.ifrsul.edu.br

Abstract. *IoT devices become popular and the use of their resources extends into the possibility of creating interactive environments. However, this technology still seems to be restricted to people who have technical mastery of equipment and systems capable of supporting these environments. This study presents the development of an interactive authentication system with an IoT device, which allows people unfamiliar with technologies to understand the access control context while interacting with their resources in the access system made available on a web platform for visualization of data generated by the IoT device.*

Resumo. *Dispositivos IoT se popularizam e a utilização de seus recursos se estende na possibilidade de criar ambientes interativos. Entretanto, essas tecnologias ainda aparentam ser restritas a pessoas com domínio técnico de equipamentos e sistemas que dão suporte a esses ambientes. Neste estudo, é apresentado o desenvolvimento de um sistema de autenticação interativo com um dispositivo IoT que possibilita que pessoas não familiarizadas com tais tecnologias compreendam o contexto de controle de acesso e interajam com seus recursos, por meio um sistema de acesso disponibilizado em uma plataforma web para visualização dos dados oriundos do dispositivo IoT.*

1. Introdução

A convergência de diversas tecnologias surgidas nas últimas décadas permitiu a criação de ambientes interativos repletos de recursos e possibilidades. A Internet das Coisas (IoT) representa um novo paradigma que permite que as pessoas interajam com um ambiente por meio de dispositivos eletrônicos, sem ter de conhecer aspectos técnicos dos sistemas [Weiser 1991]. Nesses ambientes inteligentes, apresentam-se novos recursos de baixo custo e com ênfase na mobilidade. O desafio, nesse contexto, é desenvolver plataformas ou sistemas seguros e, ao mesmo tempo, acessíveis para o usuário comum.

Entretanto, a complexidade de aliar recursos computacionais heterogêneos nem sempre é traduzida em sistemas fáceis de serem utilizados pelo usuário comum. Para estabelecer uma experiência agradável e intuitiva, é preciso desenvolver soluções que abstraiam características técnicas de funcionamento e disponibilizem serviços de forma prática. Nessa perspectiva, a IoT pode permitir que as pessoas que até então não tinham

contato com dispositivos eletrônicos em rede passem a se familiarizar com essa tecnologia.

Um dos desafios associados a essa tecnologia é controlar a forma que seus serviços serão acessados por pessoas que possuem autorização. Sistemas habituais de autenticação valem-se de usuários identificados com credenciais, como *login* e senha, e um administrador de segurança e permissões que precisa do acesso a um computador. Entretanto, dispositivos de IoT podem ter a mobilidade como característica central, fazendo com que a brevidade de liberação de um administrador de segurança tenha papel central na Experiência do Usuário – UX, de *User Experience*.

A UX orienta-se com questões de contexto, como por exemplo o *design* estético, que garante que um botão tenha aparência agradável, enquanto o *design* funcional garante que o botão ative a função desejada corretamente. O *design* de experiência de usuário garante que o estético e o funcional operem com o contexto do resto do produto, garantindo que atenda às necessidades que se espera atingir com ele [Garrett 2011]. Ao projetar sistemas colocando o usuário como o centro do processo de desenvolvimento, consegue-se manter o foco das necessidades do usuário final e pode-se salvar tempo de desenvolvimento, evitando possíveis erros de escopo e aceitação [Lowdermilk 2013].

Para planejar um produto conectado no conceito de IoT, é necessária uma abordagem detalhada de experiência de usuário. Afinal, envolve muitas camadas e nem todas estão visíveis, porém há uma necessidade de colaboração entre diversas áreas. As novas tecnologias, incluindo a IoT, viabilizaram muitas possibilidades, sistemas e soluções acessíveis aos consumidores antes jamais imaginados. O estudo de Shi *et al.* (2017) apresenta um modelo de autenticação para dispositivos IoT utilizando o sinal de Wi-Fi para perceber a movimentação e gestos do usuário e disponibilizar serviços de dispositivos. Seu trabalho apresentou uma evolução na segurança dos sistemas de acesso ao possibilitar uma personalização da segurança dos dispositivos.

Considera-se a hipótese de que um sistema móvel de autenticação, no qual o administrador de segurança, de posse de um smartphone identificado, pode decidir se autoriza ou não o acesso a um dispositivo. A proposta deste estudo é a apresentação de uma abordagem para sua utilização em controle de acesso em casos simples, que mais tarde será testada com usuários para validação de aceitação. Essa proposta é demonstrada com um sistema web integrado a um dispositivo IoT que obtém dados de variáveis ambientais sobre a qualidade do ar e apresenta essas informações em uma página web. A interface da página web permite ao usuário se identificar de forma acessível e intuitiva, onde o administrador de segurança pode, por meio de um *smartphone*, liberar o acesso ao sistema, independentemente de sua localização. Da mesma forma, a abordagem proposta visa abreviar o controle de acesso para torná-lo simplificado e alinhado a casos simples, proporcionando ao usuário uma experiência agradável de interação.

No capítulo 2, apresenta-se o Dispositivo Brezobomba, detalhando o projeto de hardware do dispositivo IoT envolvido no estudo. Nesse capítulo também é apresentado o modelo de autenticação do dispositivo. Por fim, em Considerações Finais estão algumas observações sobre os objetivos alcançados e os próximos passos deste projeto.

2. O Dispositivo Brezobomba

Tecnologias IoT vêm sendo utilizadas para criar ambientes interativos, porém poucos estudos abordam a usabilidade dos sistemas de acesso desses dispositivos. Para criar um cenário de testes que possibilite avaliar a interação entre as pessoas e um dispositivo IoT,

foi desenvolvido o dispositivo Brezobomba. O diferencial desse dispositivo é sua simplicidade de utilização, em uma forma de introduzir pessoas que estão familiarizadas com o uso de computadores e smartphones, porém sem conhecimento avançado de como se relacionar com aparelhos que permitam interação.

O objetivo da Brezobomba é capturar dados do ambiente através de sensores, disponibilizar localmente essa informação e também transmiti-la para um servidor web, onde os dados podem ser repassados para visualização de usuários. O dispositivo possui uma estrutura em ferro de um extintor de incêndio, como reaproveitamento de materiais, que permite sua aplicação em ambientes externos, preservando a integridade dos elementos eletrônicos em seu interior. O projeto da Brezobomba, visto na Figura 1, busca traduzir a urgência da preocupação com a qualidade do ar com o desenho de uma bomba.



Figura 1. O Dispositivo Brezobomba

Os elementos de hardware que compõem a Brezobomba estão esquematizados na ilustração da Figura 2 e explicados no texto subsequente.

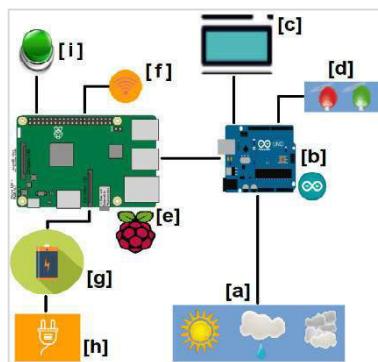


Figura 2. Arquitetura de hardware da Brezobomba

O dispositivo possui sensores [a] que aferem no ambiente os dados de umidade, temperatura e índice de CO₂ e transmitem os mesmos para um Arduino Uno [b]. Esse tem a função de exibir os dados em um monitor de LCD local [c] e controlar um sistema de aviso com luzes LED [d], no qual a concentração de CO₂ é representada com a luz verde, para casos de concentração até 420 ppm, ou vermelha, para valores acima de 420 ppm. O Arduino Uno [b] também transmite os dados para publicação através de um Raspberry Pi [e], que é responsável por receber esses dados e transmiti-los através de uma conexão Wi-Fi [f] para um servidor web. O Raspberry Pi fornece energia a todo o dispositivo, suportado por uma bateria [g], que é carregada através de uma tomada externa [h]. Essa estrutura permite que a Brezobomba tenha sua aplicação por determinado tempo em locais diversos, independentemente da existência de uma fonte de energia próxima, desde que conte com a presença de sinal Wi-Fi. Por fim, é disponibilizado o botão de

interação [i] que, quando executado, informa em tempo real os dados coletados assim como o aviso de alerta de concentração de CO₂ em níveis críticos.

Ao inicializar a Brezobomba, o sistema operacional Raspbian do Raspberry Pi é inicializado. Com ele, é carregado um software, desenvolvido em linguagem Python, que trabalha na aquisição de dados. As informações de umidade e temperatura chegam ao Raspberry por meio de um protocolo serial com um sensor DHT22. O Arduino Uno captura leituras analógicas de um sensor MQ-135, que é um sensor eletroquímico que identifica a concentração de CO₂. A comunicação entre o Arduino e o Raspberry acontece por meio do barramento I2C. O Arduino executa um firmware simples I2C-slave no qual realiza leituras da porta analógica escolhida.

2.1. Arquitetura do Sistema

Uma plataforma, após receber os dados capturados pela Brezobomba, tem a atribuição de armazenar os dados aferidos em um banco de dados relacional e publicá-los em tempo real, para visualização em uma interface gráfica em uma página web. O ciclo dos dados e organização dos componentes da plataforma é ilustrado na Figura 3.

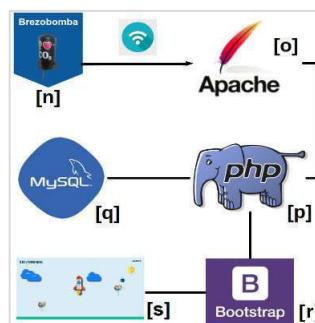


Figura 3. Arquitetura da plataforma

A transmissão dos dados aferidos pela Brezobomba [n] para o restante da plataforma ocorre a cada 60 segundos, independentemente da existência de requisições de usuários. Para recepção dos dados, a plataforma conta um servidor web Apache [o]. Os dados recebidos são armazenados em um banco de dados relacional MySQL [q], que é acessado e gerenciado com webservices criados em linguagem PHP [p]. Além disso, acontece a comunicação entre os webservices e a interface gráfica, criada com o framework Bootstrap [r], para exibição [s] dos dados aos usuários.

A interface, que é exibida após o controle de acesso concedido ao usuário, tem o objetivo de tornar a informação comprehensível também de maneira lúdica. A Figura 4 exemplifica como são apresentados na interface os dados capturados pela Brezobomba em um determinado ambiente.

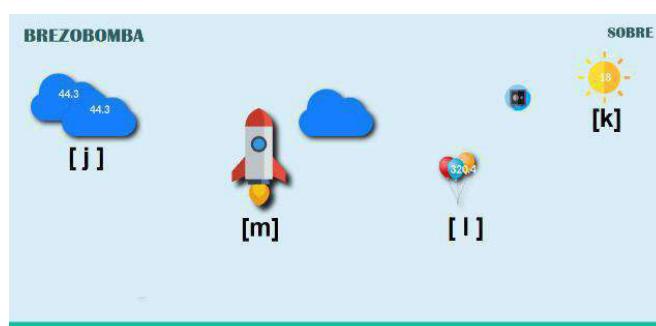


Figura 4. Interface gráfica da página web para visualização dos dados

Os dados de umidade e temperatura são representados pelas nuvens [j] e pelo sol [k], que no caso da Figura 4 estão, respectivamente, com 44.3 % e 18 °C. No conjunto de balões [l], é apresentada a concentração de CO₂, em 320,4 ppm no caso ilustrado. O sistema verifica se a concentração de CO₂ está acima de 420 ppm e, se sim, apresenta uma tela de aviso de perigo para aquele ambiente. A interface gráfica permite ao usuário, por meio do ícone da Brezobomba [m], utilizar o botão de interação do dispositivo para simular o aviso de perigo de níveis críticos de CO₂.

2.2. O Controle de Acesso e Abordagem de Ensino

Roman *et al.* (2011) explicam que os sistemas IoT para se tornarem totalmente incorporados à realidade de utilização, devem ser testados quanto sua segurança. Apontam, ainda, que as tradicionais técnicas de segurança não são suficientes para garantir todas as necessidades desses novos ambientes, cabendo aos pesquisadores descobrir a real extensão de tais obstáculos. Vermesan e Friess (2013) esclarecem que a segurança de dispositivos IoT requer uma variedade de controles de acesso associados a papéis de usuários e esquemas de utilização. A heterogeneidade e diversidade dos dispositivos requerem o desenvolvimento de um controle de acesso leve e adaptado ao contexto de aplicação.

Na Figura 5 é ilustrado o modelo de funcionamento do controle de acesso do sistema da Brezobomba, envolvendo usuários, administradores e a manipulação da interface web.

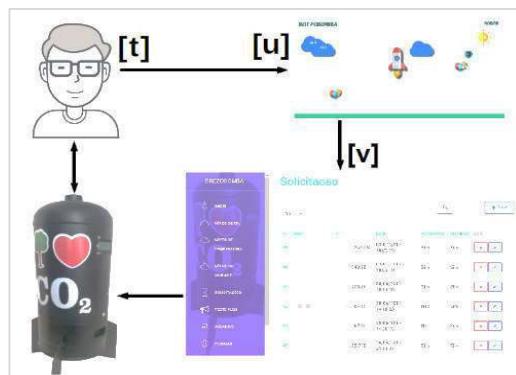


Figura 5. Esquema de autenticação de acesso

O usuário [t] faz uma requisição de acesso clicando no ícone da Brezobomba na interface gráfica da web [u], e se identifica informando seu nome ou vinculando sua rede social. A plataforma envia a solicitação ao painel de controle de acesso [v], que notifica com uma mensagem *push* o smartphone de um administrador responsável pela decisão de autorizar a requisição. Se autenticado pela rede social e autorizado, o usuário passa a ter direito de solicitar o botão de interação do dispositivo a qualquer momento. Por outro lado, se autenticado pelo nome, toda interação é intermediada pela autorização prévia de um administrador. O modelo de acesso ao sistema Brezobomba é gerenciado em uma interface web acessível de fácil compreensão. A interface web do administrador, apresentada na Figura 6, foi estruturada de maneira responsiva como forma para que possa ser utilizada em vários dispositivos e por pessoas leigas em tecnologia. Para ter acesso é necessário autenticar-se com nome ou a rede social.

O sistema é um modelo que tem uma abordagem educacional relevante, envolvendo e desafiando pessoas, independentemente da faixa etária ou grande conhecimento prévio, a conhecer e entender conceitos tecnológicos. A Brezobomba pode

envolver diversas áreas, sendo possível realizar trabalhos interdisciplinares em espaços públicos ou educacionais em escolas. Como é recorrente a necessidade de soluções de aprendizagem móveis sustentáveis sem incorrer em enormes custos em termos de infraestrutura [Yadav 2017], a Brezobomba se torna uma ferramenta em potencial.



Figura 6. Interface web de administrador

3. Considerações Finais

Este trabalho buscou apresentar um sistema de autenticação interativo com dispositivo IoT e demonstrar que é possível utilizar as tecnologias disponíveis, gratuitas e de baixo custo, para implementação de aparelhos IoT. Buscou-se estabelecer um acesso seguro a partir de uma política efetiva de controle de acesso simplificado. Como trabalhos futuros, pretende-se a execução de testes com usuários, vislumbrando a compreensão do contexto de utilização, para validar o objetivo da abordagem proposta, verificar sua aceitação, usabilidade e aspectos de interação do sistema, com grupos específicos em espaços públicos. Aspectos de experiência de usuário são fundamentais na construção de sistemas, inclusive no controle de acesso e sua integração com dispositivos IoT.

Referências

- Garrett, J. J. (2011). *The elements of user experience : user-centered design for the Web and beyond*. Thousand Oaks: New Riders.
- Lowdermilk, T. (2013). *User-centered design*. New York: O'Reilly Media.
- Roman, R., Najera, P. and Lopez, J. (sep 2011). Securing the Internet of Things. *Computer*, v. 44, n. 9, p. 51–58.
- Shi, C., Liu, J., Liu, H. and Chen, Y. (2017). Smart User Authentication through Actuation of Daily Activities Leveraging WiFi-enabled IoT. In *18th International Symposium on Mobile Ad Hoc Networking and Computing*. ACM Press.
- Vermesan, O. and Friess, P. (2013). *Internet of things : converging technologies for smart environments and integrated ecosystems*. Aalborg: River Publishers.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, v. 265, n. 3, p. 94–104.
- Yadav, D. (2017). Low-Cost Mobile Learning Solutions for Community Health Workers. *26th International Conference on World Wide Web Companion*, p. 729–734.

Protocolo de busca a Testes de Penetração em dispositivos móveis

Guilherme Leal Kaiser, Daniel Dalalana Bertoglio

Universidade Feevale – Novo Hamburgo – RS – Brasil

guilhermelealkaiser@gmail.com, dalalana@feevale.br

Abstract. This article describes the development for a search protocol based on the systematic mapping model, focusing on penetration testing towards mobile devices. It presents data on the importance of the focus to tests in these devices, followed by the systematic mapping model, which through it were developed the planning and conduction stages, ending with the quality analysis of the articles returned from the searches. This protocol is the basis for an upcoming study in order to propose a detailed analysis on the theme, aiming among other reasons to answer the research questions presented here.

Resumo. Este artigo descreve o desenvolvimento de um protocolo de busca baseado no modelo de mapeamento sistemático, com foco em testes de penetração voltado a dispositivos móveis. Ele apresenta dados da importância do foco a testes nesses dispositivos, seguido do modelo de mapeamento sistemático. Através dele foram desenvolvidas as etapas de planejamento e condução, finalizando com a análise de qualidade dos artigos retornados das buscas. Esse protocolo, por fim, é a base para um próximo estudo a fim de propor uma análise detalhada sobre o tema, visando, dentre outras razões, responder as questões de pesquisa aqui apresentadas.

1. Introdução

Atualmente, a segurança de informações tem representado com notoriedade parte das pesquisas relacionadas ao tratamento, prevenção e proteção de dados, processos e tecnologias. Em paralelo a isso, com o crescente uso de dispositivos móveis e o consequente aumento no tráfego de dados gerados pelos mesmos, as preocupações com riscos e vulnerabilidades ampliaram a atuação dessas pesquisas para contribuições voltadas a teste e avaliação de segurança. Uma das técnicas de teste de segurança é o teste de penetração, mais conhecido como *Pentest*. *Pentest*, segundo Bertoglio (2017), é o nome dado à tentativa controlada de penetrar um sistema ou rede a fim de detectar vulnerabilidades.

Segundo o terceiro relatório anual emitido pela Juniper Networks Inc (2013), a crescente dependência de dispositivos inteligentes provou ser um alvo irresistível para os invasores, pois eles estão rapidamente eclipsando computadores na era pós-PC. Nesse sentido, é igualmente relevante indicar a grande proliferação de softwares maliciosos nas diversas plataformas móveis, em particular na plataforma Android, a qual não obteve uma resposta rápida dos fornecedores de produtos de segurança da informação [Braga 2012].

Por esse motivo, foi constatada a importância de gerar uma pesquisa voltada a área, para obter como resultado os últimos estudos e práticas relacionadas.

O estudo, portanto, foi feito de acordo com o modelo de Mapeamento Sistemático proposto por Petersen et al. (2008), que o divide em 3 fases: Planejamento, Condução e Apresentação. No entanto, este protocolo de busca será focado apenas nas etapas de Planejamento e Condução, descritos nas seções quatro e cinco, após as sessões dois e três que trazem uma visão geral sobre *Pentest* e Mapeamento Sistemático, respectivamente. Finalizando, então, com a sessão de lições aprendidas e principais contribuições, seguida da conclusão.

2. Pentest

A utilização de *Pentest* visa encontrar vulnerabilidades em um determinado sistema ou rede. Ele também é referido como "*hacking ético*" porque os testadores de penetração investigam o sistema alvo do ponto de vista de um invasor, relatando fraquezas em vez de explorá-las [Bohme 2010]. Há inúmeros benefícios de testes de penetração nos negócios, bem como perspectiva técnica. Algumas das principais razões para a adoção de testes de penetração são: problemas de segurança, priorizar riscos de segurança, perda financeira [Stefinko 2016].

Um *Pentest* pode ser dividido numa série de fases, que quando colocadas juntas formam uma metodologia abrangente para completar um teste de penetração [Ramos 2013]. Dentre elas tem-se o escopo de metas, o recolhimento de informações, a descoberta de destino, a enumeração do alvo, o mapeamento de vulnerabilidades, a engenharia social, a exploração de destino, a escalação de privilégios, a manutenção do acesso e a documentação e relatórios [Stefinko 2016].

3. Mapeamento Sistemático

O protocolo de busca foi baseado no modelo de mapeamento sistemático que, segundo Santos (2015), é uma forma de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma questão de pesquisa particular. Conforme Petersen et al. (2008), dentre as etapas essenciais do processo estão a definição de questões de pesquisa, a realização de pesquisas relevantes, triagem de documentos, palavras-chave de abstrações e extração e mapeamento de dados.

Todas as etapas do mapeamento sistemático estão apresentadas na Figura 1, que traz o fluxo em que elas ocorrem durante o mapeamento.

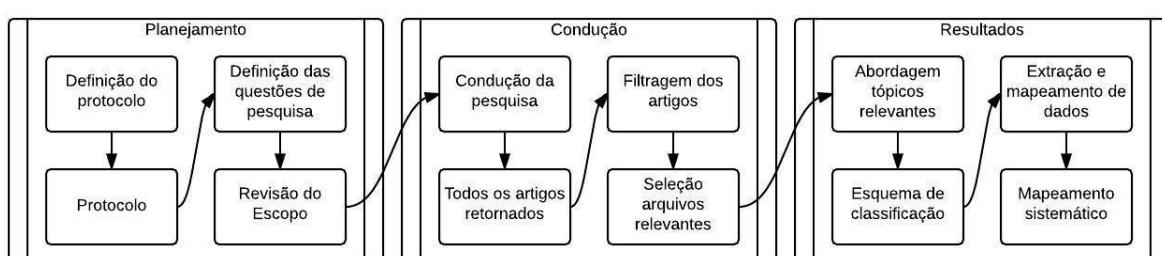


Figura 1. Processo de estudo de mapeamento sistemático.

Este artigo, conforme mencionado anteriormente, trará apenas o protocolo final da busca para um futuro mapeamento sistemático, que englobará as partes de planejamento e condução, descritas na sequência.

4. Planejamento

Esta pesquisa tem como foco planejar e executar uma busca nos moldes de um mapeamento sistemático com base no tema de testes de penetração em dispositivos móveis.

A estrutura da questão desta pesquisa foi baseada segundo a proposta trazida por Kitchenham [2007 apud Bertoglio 2017] que a organiza nos critérios de PICO (*Population, Intervention, Comparison, Outcome*):

- População (*Population*): artigos de pesquisa relacionados à área de Segurança da Informação.
- Intervenção (*Intervention*): testes de penetração.
- Controle (*Comparison*): dispositivos móveis.
- Resultados (*Outcome*): obter uma análise sobre testes de penetração em dispositivos móveis para identificar as ferramentas, metodologias de testes e possíveis falhas de segurança encontradas nos modelos analisados.

Foram descritas cinco questões de pesquisa (Q) e estas são apresentadas na sequência.

Q1. Quais são as principais ferramentas utilizadas neste modelo de *Pentest*?

Q2. Quais as principais falhas identificadas?

Q3. Quais os dispositivos móveis mais utilizados nos testes?

Q4. Quais os cenários de execução dos *Pentests* em dispositivos móveis?

Q5. Quais as principais limitações/desafios?

Com relação ao processo de pesquisa, foram selecionadas bases de dados que contém modelo de busca baseado em palavras chave em artigos da área da ciência da computação, escritos em português ou inglês. Dentre as selecionadas estão ACM Digital Library, IEEE Xplore, Scopus e Springer Link.

Já os termos e sinônimos usados na *string* de busca foram divididos também com base nos critérios de PICO, mostrados anteriormente. A **Tabela 1** apresenta essas divisões.

Tabela 1. Termos e sinônimos organizados de acordo com a estrutura (critérios de PICO)

ESTRUTURA	TERMOS	SINÔNIMOS
<i>População</i>	<i>Security Information</i>	
<i>Intervenção</i>	<i>Penetration Test</i>	<i>Security Test, Security Testing, Penetration Testing, Pentest</i>
<i>Controle</i>	<i>Mobile Devices</i>	<i>Mobile</i>
<i>Resultados</i>	<i>Tool</i>	<i>Tools, Software, Program, Application</i>
	<i>Vulnerability</i>	<i>Flaw</i>

	<i>Mobile Operation System</i>	<i>Android, Ios</i>
	<i>Model</i>	<i>Process, Method, Framework, Methodology</i>
	<i>Scenarios</i>	<i>Context, Environment</i>
	<i>Challenges</i>	<i>Open Problems, Open research topics</i>

Nas *strings* de busca foi utilizado o operador “OR” para selecionar os termos e sinônimos, e o operador “AND” para selecionar os termos da estrutura da questão: população, intervenção, controle e resultados. Devido ao retorno negativo de artigos com a *string* completa nas bases Scopus, IEEE e ACM, foi necessário realizar as buscas com *strings* reduzidas conforme apresentado na Figura 2:

Scopus:
(“Security Test” OR “Security Testing” OR “Penetration Test” OR “Penetration Testing” OR “Pentest”) AND (“Mobile Devices” OR “Mobile”) AND (“Tool” OR “Tools” OR “Software” OR “Program” OR “Application”) AND (“Vulnerability” OR “Flaw”)
IEEE e ACM:
(“Security Test” OR “Security Testing” OR “Penetration Test” OR “Penetration Testing” OR “Pentest”) AND (“Mobile Devices” OR “Mobile”)

Figura 2. *Strings* reduzidas utilizadas para a busca nas bases de dados especificadas.

Foram descritos dois critérios de inclusão (IC) e dois de exclusão (EC), que são de grande importância para a apuração dos resultados. Tais critérios são responsáveis por apoiar a seleção dos artigos apropriados e são empregados para reduzir o número de artigos que retornam das *engines* de busca [Bertoglio 2017].

- IC1.** O tema principal traz um estudo sobre *Pentests* em dispositivos móveis;
- IC2.** O estudo principal propõe um modo para prover *Pentests* em dispositivos móveis;

- EC1.** O estudo principal não está relacionado diretamente a *Pentests* em dispositivos móveis;

- EC2.** O estudo principal não contém algum tipo de avaliação para demonstrar os resultados;

Na parte de avaliação de qualidade (*Quality Assessment* - QA), que visa mensurar a relevância de cada um dos estudos retornados, foram definidos três critérios, identificados com as questões:

- QA1.** O estudo apresenta uma contribuição ao tema de *Pentest* em dispositivos móveis?

- QA2.** O estudo descreve as ferramentas ou modelos utilizados?

- QA3.** Apresenta uma avaliação das ferramentas e/ou modelos utilizados em *Pentests* direcionados a dispositivos móveis?

Para cada uma das questões dos critérios de qualidade é utilizada a seguinte pontuação: Y (sim) = 1; P (parcialmente) = 0,5, N (não) = 0. Com isso, o *Score* (soma das três questões) pode classificar os estudos em: 0 ou 0,5 (limitado), 1 (regular), 1,5 (bom), 2 (muito bom) e 2,5 ou 3 (excelente).

O processo de seleção foi dividido em quatro etapas: busca nas bases de dados, eliminação das redundâncias, seleção final e avaliação da qualidade. Estas são descritas na sequência.

- Busca nas bases de dados: nesta etapa foram utilizadas as *strings* de busca geradas a partir dos termos e sinônimos, para fazer a procura nas bases de dados selecionadas.
- Eliminação das redundâncias: após a busca foram eliminadas e armazenadas as redundâncias.
- Seleção final: nesta etapa o título e o resumo de cada artigo retornado das buscas são lidos, e com base nos critérios de inclusão e exclusão é feita a seleção dos que serão utilizados.
- Avaliação da qualidade: concluída a etapa de seleção final, os artigos selecionados são lidos e analisados de acordo com os critérios de qualidade.

5. Condução

A condução deste protocolo detalha os resultados obtidos após as etapas do processo de seleção. Tendo como resultado um total de 286 artigos após a primeira leitura, 41 foram eliminados e arquivados na fase de eliminação de redundâncias. Já na terceira, que representa a seleção final, atingiu-se como resultado 33 artigos. Na etapa final do processo de seleção, os 33 artigos selecionados foram avaliados com base nos critérios de qualidade mostrados anteriormente. Segundo Bertoglio (2017), tais critérios ajudam a avaliar a confiabilidade dos estudos. A **Tabela 3** traz o resultado dessa avaliação feita por artigo, detalhando o ano e a referência de cada um deles. Ela exibe nas colunas 1, 2, 3 os *scores* relacionadas a *Quality Assessment* (QA) e na coluna *Sc* traz o *score* final, classificando cada artigo de acordo o Sc obtido, conforme mostra a coluna Des.

Tabela 3. Resultados da análise da qualidade por artigo

Estudos		QA		Qualidade		Estudos		QA		Qualidade					
ID	Referência	Ano	1	2	3	Sc	Des	ID	Referência	Ano	1	2	3	Sc	Des
1	Morais	2011	P	Y	Y	2,5	E	18	Sadeghi	2014	P	P	N	1	R
2	Habib	2008	P	Y	P	2	M	19	Jadhav	2015	Y	P	P	2	M
3	Mahmood	2012	Y	Y	Y	3	E	20	Wang	2015	Y	Y	P	2,5	E
4	Habib	2009	Y	Y	Y	3	E	21	Lee	2015	Y	Y	Y	3	E
5	Wang	2015	P	Y	N	1,5	B	22	Knorr	2015	P	P	Y	2	M
6	Wu	2014	Y	Y	Y	3	E	23	Koivunen*	2016				0	L
7	Wu	2015	P	Y	N	1,5	B	24	Yang	2016	Y	Y	Y	3	E
8	Gagnon	2016	Y	Y	Y	3	E	25	Debbabi	2005	Y	Y	Y	3	E
9	Salva	2015	P	Y	Y	2,5	E	26	Knorr	2015	Y	Y	Y	3	E
10	Brandt	2014	Y	Y	Y	3	E	27	Hunt	2013	Y	Y	Y	3	E

11	Salva	2013	P	P	P	1,5	B	28	Avancini	2013	Y	P	P	2	M
12	Fahrianto	2016	Y	P	P	2	M	29	Bojjagani*	2016				0	L
13	Debbabi*	2006				0	L	30	Badura	2009	P	P	P	1,5	B
14	Abgrall	2014	P	P	Y	2	M	31	Javed	2014	Y	Y	Y	3	E
15	Morais	2012	P	Y	Y	2,5	E	32	Mulliner	2009	Y	Y	Y	3	E
16	Noponen	2008	Y	Y	P	2,5	E	33	Mulliner	2006	Y	P	P	2	M
17	Salva*	2013				0	L								

Legenda: Y: Sim, N: Não, P: Parcialmente, Sc: Score, Des: Descrição, B: Bom, M: Muito Bom, E: Excelente, R: Regular, L: Limitado

*Artigos que não se obteve acesso para avaliação da qualidade.

6. Lições Aprendidas e Principais Contribuições

Após a análise de qualidade concluída, obteve-se um protocolo de busca base para realização de um mapeamento sistemático, que pode, além disso, ser uma fonte de pesquisa focada ao assunto, provendo estudos que foram classificados de acordo com a sua relevância ao mesmo. Além disso na análise de qualidade foram considerados como que respondendo parcialmente a primeira questão, estudos que demonstram ou provem maneiras de examinar os dispositivos ou aplicativos a fim de identificar possíveis vulnerabilidades. Tais estudos não foram desconsiderados por seu assunto não ser diretamente relacionado a *Pentest* porque as análises trazidas por eles poderão ser usadas na fase de mapeamento de vulnerabilidades dos testes.

Por fim, como principal contribuição e motivação está à busca em estabelecer aos interessados no assunto uma amostra de como estão os principais estudos, junto com a sua análise inicial já concluída. Indicando, além disso, os autores que desempenharam algum esforço nesta área, para se necessário poder ser feita uma busca específica a outros trabalhos realizados por eles.

7. Conclusão

Como citado anteriormente, a área relacionada a testes de segurança tem crescido nos últimos tempos, tendo como grande fator a busca pela segurança da informação. E *Pentests* aparecem como um dos principais, pois visam verificar vulnerabilidades para, a partir delas, analisar seu risco e impacto ao sistema direcionado, a fim de prevenir possíveis ataques.

Foi visando o crescente estudo nesta área que o protocolo de busca teve como foco preparar a pesquisa para um futuro mapeamento sistemático a testes de penetração em dispositivos móveis, já que esses dispositivos estão a cada dia mais presentes, armazenando, realizando tarefas e transações com dados de valor significativo para os usuários.

A partir deste tema o protocolo de busca obteve o conteúdo primário para o estudo nesta área, após toda a busca, seleção e análise da qualidade dos artigos relevantes. Na última etapa pode-se chegar a uma qualidade significativa, já que 72% dos estudos obtiveram score final entre 2 e 3 pontos, classificando-os, assim, como muito bons ou excelentes. Outro ponto importante a ser analisado é que também 72% dos estudos finais que tiveram a sua

qualidade avaliada foram realizados nos últimos cinco anos, ou seja, são estudos recentes na área.

Com isso foi possível perceber o quanto importante é o enfoque deste protocolo, reunindo informações sobre um estudo relevante para o contexto atual, mas que tem suas pesquisas e testes muito recentes ainda. E é isso o que motiva o próximo trabalho, que a partir do conteúdo aqui obtido irá propor uma análise detalhada sobre este enfoque e modelo de teste de segurança, visando, dentre outras razões, responder as cinco questões de pesquisa apresentadas neste protocolo.

Referências

- Petersen K, Feldt R, Mujtaba S, Mattsson M. (2008). “Systematic mapping studies in software engineering”. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. EASE’08. British Computer Society, Swinton. pp 68–77.
- Bertoglio D. Dalalana, Zorzo A. Francisco (2017). “Overview and open issues on Penetration test”.
- Juniper networks - Mobile Threat Center Third Annual Mobile Threats Report: March 2012 through March 2013.
- Habib S. Mahbub, Jacob Cyril, Olovsson Tomas (2008). “A Practical Analysis of the Robustness and Stability of the Network Stack in Smartphones”. In: Department of Computer Science & Engineering, Chalmers University of Technology, Gothenburg, SE-41296, Sweden.
- Mahmood Riyadh, Esfahani Naeem, Kacem Thabet, Mirzaei Nariman, et al. (2012). “A Whitebox Approach for Automated Security Testing of Android Applications on the Cloud”. Disponível em: Computer Science Department George Mason University.
- Bohme Rainer, Felegyhazi Mark (2010). “Optimal Information Security Investment with Penetration Testing”. In: International Computer Science Institute, Berkeley, California
- Stefinko Yaroslav, Piskozub Andrian, Banakh Roman (2016). “Manual and Automated penetration testing. Benefits and Drawbacks. Modern tendency”.
- Braga A. Melo, Nascimento E. Nogueira, Palma L. Rodrigues, Rosa R. Pereira (2012). “Introdução à Segurança de Dispositivos Móveis Modernos – Um Estudo de Caso em Android”.
- Ramos J. J. Afonso (2013). “Sistema Automático para Realização de Testes de Penetração”
- Morais Anderson, Cavalli Ana, Martins Eliane (2011). “A model-based attack injection approach for security validation”.
- Habib Sheikh Mahbub, Jacob Cyril, Olovsson Tomas (2009). “An Analysis of the Robustness and Stability of the Network Stack in Symbian-based Smartphones”.
- Wang Yong (2015). “An Automated Virtual Security Testing Platform for Android Mobile Apps”.

- Wu Daoyuan, Chang Rocky K. C. (2014). “Analyzing Android Browser Apps for file:// Vulnerabilities”.
- Wu Jingzheng, Wu Yanjun, Wu Zhifei, Yang Mutian, et. al (2015). “AndroidFuzzer: Detecting android vulnerabilities in fuzzing cloud”.
- Gagnon François, Ferland Marc-Antoine, Fortier Marc-Antoine, Desloges Simon, et. al (2016). “AndroSSL: A Platform to Test Android Applications Connection Security”.
- Salva Sébastien, Zafimiharisoa Stassia R. (2015). “APSET, an Android aPplication SEcurity Testing tool for detecting intent-based vulnerabilities”.
- Brandt N. Benecke, Stamp Mark (2014). “Automating NFC message sending for good and evil”.
- Salva Sébastien, Zafimiharisoa Stassia R. (2013). “Data vulnerability detection by security testing for Android applications”.
- Fahrianto Feri, Lubis M. Fadil, Fiade Andrew (2016). “Denial-of-Service attack Possibilities on NFC Technology”.
- Abgrall Erwan, Traon Y. Le, Gombault Sylvain, Monperrus Martin (2014). “Empirical Investigation of the Web Browser Attack Surface under Cross-Site Scripting: An Urgent Need for Systematic Security Regression Testing”.
- Morais Anderson, Hwang Iksoon, Cavalli Ana, Martins Eliane (2012). “Generating attack scenarios for the system security validation”.
- Noponen Sami, Karppinen Kaarina (2008). “Information Security of Remote File Transfers with Mobile Devices”.
- Sadeghi Alireza, Esfahani Naeem, Malek Sam (2014). “Mining the Categorized Software Repositories to Improve the Analysis of Security Vulnerabilities”.
- Jadhav Suyash, Oh Tae, Kim Y. Ho, Kim J. Nyeo (2015). “Mobile device penetration testing framework and platform for the mobile device security course”.
- Wang Yong, Alshboul Yazan (2015). “Mobile Security Testing Approaches and Challenges”.
- Lee W. Hao, Ramanujam M. Srirangam, Krishnan S. P. T. (2015). “On designing an efficient distributed black-box fuzzing system for mobile devices”.
- Knorr Konstantin, Aspinall David, Wolters Maria (2015). “On the Privacy, Security and Safety of Blood Pressure and Diabetes Apps”.
- Yang Yaping, Cai Lizhi, Zhang Yanguo (2016). “Research on non-authorized privilege escalation detection of android applications”.
- Debbabi Mourad, Saleh Mohamed, Talhi Chamseddine, Zhioua Sami (2005). “Security Analysis of Wireless Java”.
- Knorr Konstantin, Aspinall David (2015). “Security Testing for Android mHealth Apps”.
- Hunt Ray (2013). “Security testing in Android networks - A practical case study”.

- Avancini Andrea, Ceccato Mariano (2013). “Security testing of the communication among Android applications”.
- Badura Thomas, Becher Michael (2009). “Testing the Symbian OS Platform Security Architecture”.
- Javed Ashar, Schwenk Jörg (2014). “Towards Elimination of Cross-Site Scripting on Mobile Versions of Web Applications”.
- Mulliner Collin (2009). “Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones”.
- Mulliner Collin, Vigna Giovanni (2006). “Vulnerability Analysis of MMS User Agents”.
- Santos M. Auréli, Barreto R. da Silva (2015). “Mapeamento Sistemático”.
- Salva Sébastien, Zafimiharisoa Stassia R., Laurencot Patrice (2013). “Intent security testing: An Approach to testing the Intent-based vulnerability of Android components”.
- Koivunen Lauri, Rauti Sampsa, Leppänen Ville, et al (2016). “Proceedings - 2016 International Conference on Software Security and Assurance, ICSSA 2016”.
- Debbabi Mourad, Saleh Mohamed, Talhi Chamseddine, Zhioua Sami (2006). “Embedded Java Security: Security for Mobile Devices”.
- Bojjagani Sriramulu (2015). “STAMBA: Security testing for android mobile banking apps”.

Uma proposta de arquitetura para identificação de anomalias em redes IoT utilizando registros de Logs

Jonathan O. Preuss¹, Bolívar M. Silva¹, Raul C. Nunes¹

¹Centro de Tecnologia – Universidade Federal de Santa Maria (UFSM)

Caixa Postal 1000 – 97105-900 – Santa Maria – RS – Brasil

Programa de Pós-Graduação em Ciência da Computação

Santa Maria, R.S.

{bolivar, jonathan.preuss}@redes.ufsm.br, ceretta@inf.ufsm.br

Resumo. *IoT tem sido amplamente empregado em infraestruturas críticas onde auxiliam no monitoramento, gestão e tomada de decisão. Com isso existe a preocupação em relação a segurança e integridade desses ambientes, a falha desses dispositivos IoT em ambientes críticos podem causar uma catástrofe. Devido à grande quantidade e heterogeneidade de dados gerados em ambientes IoT, métodos tradicionais para detecção de anomalias não são eficazes. Este artigo apresenta uma arquitetura para identificação de anomalias em dispositivos IoT, através da análise de logs utilizando ferramentas de Big Data. A arquitetura proposta, apresenta uma metodologia de análise para detecção de anomalias em tempo hábil, para grandes volumes de dados heterogêneos de redes IoT.*

Abstract. *IoT has been widely used in critical infrastructures where they assist in monitoring, management and decision making. With this there is concern regarding the security and integrity of these environments, the failure of these IoT devices in critical environments can cause a catastrophe. Due to the large amount and heterogeneity of data generated in IoT environments, traditional methods for detecting anomalies are not effective. This article presents an architecture for identifying anomalies in IoT devices, through the analysis of logs using Big Data tools. The proposed architecture presents a methodology of analysis for the detection of anomalies in a timely manner, for large volumes of heterogeneous data of IoT networks.*

1. Introdução

A Internet das Coisas (*IoT*) apesar de não ser um tema tão recente, tem sido amplamente estudado e discutido pela comunidade científica, principalmente, nos últimos anos. Um ambiente de *IoT* é composto por dispositivos e tecnologias diferentes, consequentemente é heterogênea e volumosa a geração de dados nesses ambientes [Umar Ahsan 2016]. Segundo [Qian Zhu 2010], em redes *IoT*, são necessários *gateways IoT*, que são encarregados de tornar possível a interação entre elementos de redes e tecnologias heterogêneas. Uma das aplicações do paradigma de *IoT* é o seu emprego em atividades de auxílio e monitoramento de infraestruturas críticas, cuja a falha de um dispositivo pode resultar em perdas financeiras ou catástrofes. Exemplos de uso são demonstrados em [Dan Koo 2015], que faz uso de *IoT* para o sensoriamento de uma estação de água. E

em [Kyle E. Benson 2016] que utiliza uma rede *IoT* para o monitoramento de atividades sísmicas.

Em paralelo ao amplo emprego de sistemas *IoT*, surgem vários problemas no que diz respeito à segurança, como vulnerabilidades de autenticidade e integridade dos dados e dispositivos. Devido às características simplistas dos *hardwares* utilizados em muitos dos dispositivos que compõem a *IoT*, as técnicas de segurança, tradicionais tornam-se inviáveis [Jing et al. 2014], e se tratando de ambientes de infraestruturas críticas, a agilidade na detecção de anomalias e falhas, podem evitar desastres. Nesse contexto, este trabalho apresenta uma proposta de arquitetura que visa prover segurança e integridade de redes *IoT*, através da detecção de comportamentos anômalos de dispositivos *IoT*, essa detecção é realizada através da análise de eventos de *log* gerados por dispositivos conectados aos *gateways*. O presente trabalho está organizado da seguinte forma. Seções 2. Revisão Bibliográfica, tratando os temas abordados ao longo deste trabalho, seções de Trabalhos Relacionados, demonstrando alguns trabalhos os quais autores tratam de um tema semelhante ao deste trabalho, seção 4. Arquitetura Proposta será descrito a arquitetura proposta e seu funcionamento e por fim as seções de 5. Conclusão e 6. Trabalhos Futuros.

2. Revisão Bibliográfica

Essa seção apresenta uma breve descrição dos temas abordadas nesse trabalho, visando fornecer embasamento ao leitor sobre o que se tratam. Internet das coisas é um conceito onde elementos presentes no dia a dia passam a interagir através de uma rede de dados. Em [Pallavi Sethi 2017], os autores definem *IoT* como sendo um aglomerado de diferentes tecnologias que operam em conjunto. Segundo [Janice Canedo 2016], em ambientes de *IoT*, os dois principais desafios quanto a detecção de anomalias, são: a heterogeneidade dos elementos e o grande volume de dados gerados.

Em um contexto onde existem dados diferentes sendo gerados em grande quantidade e com alta velocidade, uma boa abordagem para a análise desses dados é utilizar o conceito de *Big Data*. Em [McKinsey Global Institute 2011], o conceito de *Big Data* refere-se ao grupo de dados que não poderiam ser obtidos, administrados e armazenados pelos tradicionais sistemas de banco de dados. Existe uma variedade de ferramentas e sistemas para trabalhar com esse conceito. Nesse trabalho são utilizadas as ferramentas, Apache Hadoop¹, Hadoop File System², Apache Flume³ e Apache Spark⁴.

3. Trabalhos Relacionados

Existe um grande número de trabalhos que propõem as mais variadas técnicas e algoritmos para detecção de anomalias em ambientes *IoT*, como por exemplo técnicas de inspeção de pacotes apresentado em [Douglas H. Summerville and Chen 2015], ou técnicas de *data mining* apresentadas por [Janice Canedo 2016]. Porém, esses trabalhos se direcionam para a detecção de anomalias em aplicações específicas. Diante de ambientes com grandes diversidades tecnológicas, torna-se difícil o emprego dessas técnicas.

¹Apache Hadoop "<https://hadoop.apache.org/>"

²Hadoop File System "https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html"

³Apache Flume "<https://flume.apache.org/>"

⁴Apache Spark "<https://spark.apache.org/>"

Alguns autores trabalham com técnicas mais genéricas para detecção, como em [Fu et al. 2011], onde o autor ressalta que a heterogeneidade de uma rede *IoT*, dificulta a implantação de sistemas convencionais de detecção de intrusão. Os autores analisam o comportamento padrão dos dispositivos de *IoT* em relação ao tempo, e sugerem cinco características comportamentais: pulso, subida acentuada, descida acentuada, leitura constante e mudança de padrão, que classificam um comportamento como anômalo. Em [Yanbing Liu 2014] os autores propõem um algoritmo para detecção de intrusão em redes *IoT* utilizando análise de similaridade de pacotes. Esse algoritmo utiliza coeficiente *Jaccard* para realizar as medidas de similaridade entre um conjunto de treinamento sem possíveis anomalias e o tráfego de uma rede *IoT*.

Nos trabalhos apresentados nessa sessão, os autores utilizam algoritmos e técnicas genéricas para detecção de anomalias em ambientes *IoT*, baseando-se na análise comportamental do tráfego, porém não são trabalhadas formas de coleta de dados, além de não utilizar nenhuma técnica para execução dos algoritmos de detecção, diante de um grande fluxo de dados. O presente trabalho propõe uma arquitetura que realiza o emprego de uma ferramenta para a coleta das informações dos dispositivos *IoT* e efetua o processamento desses dados, onde é possível realizar análises comportamentais ou de similaridades, de forma ágil, aproximando-se a uma análise em tempo real.

4. Arquitetura Proposta

O escopo desse trabalho engloba a Internet das Coisas e os diversos dispositivos que a compõe. Devido as características já citadas desse tipo de rede, como as diversas fontes heterogêneas, a arquitetura apresentada traz as tecnologias e técnicas necessárias para alcançar o objetivo proposto inicialmente.

O Apache Flume possui a habilidade de operar com múltiplos *Agents* coletando informações de diferentes fontes. Dessa forma, em um cenário onde existam aplicações heterogêneas conectadas a um mesmo *gateway*, é possível criar e configurar agentes individuais para cada aplicação e seus respectivos dispositivos. Uma vez que os *gateways* estão concentrando o fluxo de dados da rede em um único ponto, posicionar agentes Flume nesses pontos garantem acesso aos *logs* de eventos de todos os dispositivos conectados nesse *gateway*. A figura 1 ilustra a arquitetura proposta.

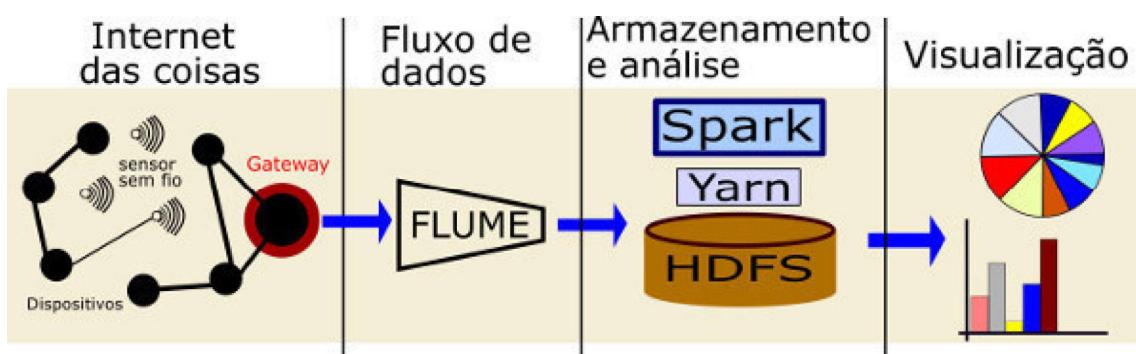


Figura 1. Arquitetura proposta

Da esquerda para a direita, temos os dispositivos que compõem as redes *IoT*, representados por círculos pretos. Os círculos maiores (com borda vermelha) representam

os dispositivos *gateways*, que concentram as informações de determinados conjuntos de sensores. Seguindo a sequência, temos o Apache Flume, que coleta, agrupa e transporta os dados dos *gateways*, criando um fluxo de informações e encaminhando para ser processado (Spark) e armazenado (HDFS). Finalmente, à direita da figura, é apresentado a parte da arquitetura referente à visualização. Diz respeito à interface com o usuário. É onde serão apresentados os alertas e demais tipos de informações, pertinentes ao administrador da rede.

Após configurar os agentes Flume em cada ponto de coleta e iniciar o processo de transmissão dos dados de *log* para destino onde está operando o ambiente Spark, inicia-se a fase de tratamento desse fluxo. O primeiro elemento a tratar esses dados é o Spark *Streaming* que irá fragmentar esse bloco de dados em pequenas partes e os encaminhar para um segundo elemento que é o Spark *engine*, onde é executada a aplicação de análise. O fluxograma representado na figura 2 exemplifica as funções que a aplicação necessita para operar adequadamente.

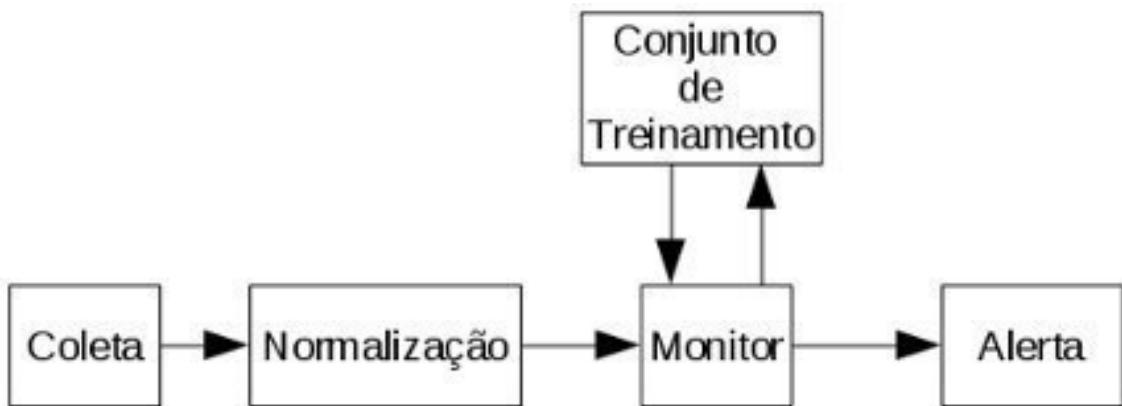


Figura 2. Fluxograma de funções da aplicação utilizada para análise de anomalias

A aplicação de análise representada na figura 2 exemplifica as funções que a aplicação necessita para operar adequadamente. A primeira função é chamada de "coleta". Essa função irá delimitar o contexto do Spark, indicando onde estão localizados os fragmentos de dados oriundos do *Streaming*, e delimitará o modo de operação do *cluster* Spark. A segunda função é a de normalização. Essa função irá extrair e normalizar os dados úteis, de acordo com a necessidade de cada tipo de dado. Após isso, os monitores armazenarão na base de dados HDFS.

Para cada aplicação monitorada pela arquitetura proposta nesse trabalho, deve existir um conjunto de treinamento que contenha dados normalizados de comportamento tido como normal para aquela aplicação. Baseado nesse conjunto de treinamento a função do monitor é comparar os dados provenientes de um fluxo e classificá-los como anômalo ou normal. Nesse ponto é executado a análise do comportamento dos dados, baseado nas cinco características definidas por [Fu et al. 2011]. A última função da aplicação é o Alerta, que é responsável por sinalizar a ocorrência de uma anomalia no comportamento de um dispositivo. O Alerta pode ser configurado como uma mensagem em uma página *web* sinalizando em qual dispositivo e o tipo de anomalia que está ocorrendo.

5. Conclusão

O uso de Internet das Coisas para aplicações críticas está crescendo nos últimos anos, com isso surge a preocupação com a integridade e segurança desses ambientes. Devido à grande quantidade e principalmente diferença dos dados gerados pelos dispositivos das redes *IoT*, técnicas de detecção anomalias convencionais não são bem sucedidas. Através do estudo dos trabalhos relacionados e das ferramentas existentes, para processamento e análise rápida de grandes quantidades de dados, foi apresentado uma arquitetura que pode suprir algumas das dificuldades apresentadas de detecção de anomalias no contexto de *IoT*. Dado as características citadas da arquitetura proposta, enfatizando a capacidade de coleta, análise de grandes quantidades de dados heterogêneos, espera-se que a arquitetura proposta realize a análises de possíveis falhas, em tempo de execução ou tão próximo quanto possível. Dessa forma, redes *IoT* com sistemas sensíveis que necessitam ser altamente tolerantes a falhas, podem reduzir o tempo necessário para a identificação de falhas, bem como na prevenção das mesmas.

6. Trabalhos Futuros

Como trabalhos futuros, pretende-se implementar a arquitetura proposta em um cenário de testes, para validação e análise da real viabilidade de utilização do modelo proposto. Para resultados mais precisos, o ideal seria utilizar um cenário com o maior número de dispositivos possível, como sensores de temperatura, sensores de pressão, câmeras de vigilância entre outros.

Referências

- Dan Koo, Kalyan Piratla, J. M. C. (2015). Towards sustainable water supply: Schematic development of big data collection using internet of things (iot). *International Conference on Sustainable Design, Engineering and Construction*.
- Douglas H. Summerville, K. M. Z. and Chen, Y. (2015). Ultra-lightweight deep packet anomaly detection for internet of things devices. In *Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance*. IEEE.
- Fu, R., Zheng, K., Zhang, D., and Yang, Y. (2011). An intrusion detection scheme based on anomaly mining in internet of things.
- Janice Canedo, Anthony Skjellum, S. G. C. o. E. (2016). Using machine learning to secure iot systems. In *Privacy, Security and Trust (PST), 2016 14th Annual Conference on*. IEEE.
- Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., and Qiu, D. (2014). Security of the internet of things: Perspectives and challenges. *Wireless Networks*, 20(8):2481–2501.
- Kyle E. Benson, Qing Han, K. K. P. N. N. V. (2016). Resilient overlays for iot-based community infrastructure communications. *Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on*.
- McKinsey Global Institute, James Manyika, M. C. B. B. J. B. R. D. C. R. A. H. B. (2011). *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. McKinsey Global Institute, 1th edition.

- Pallavi Sethi, S. R. S. (2017). Internet of things: Architectures, protocols, and applications. In *Journal of Electrical and Computer Engineering Volume 2017* (2017). Hindawi.
- Qian Zhu, Ruicong Wang, Q. C. Y. L. W. Q. (2010). IoT gateway: Bridging wireless sensor networks into internet of things. *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*.
- Umar Ahsan, A. B. (2016). A review on big data analysis and internet of things. In *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE.
- Yanbing Liu, Q. W. (2014). A lightweight anomaly mining algorithm in the internet of things. *Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on*.

Anomaly-based Web Application Firewall using HTTP-specific features and One-Class SVM

Nico Epp¹, Ralf Funk¹, Cristian Cappo¹

¹ Facultad Politécnica – Universidad Nacional de Asunción
San Lorenzo – Paraguay

{nicoeppfriesen, ralffunk0}@gmail.com, ccappo@pol.una.py

Abstract. *Vulnerabilities in web applications pose great risks because they can be exploited by malicious attackers through the Internet. Web Application Firewalls placed in front of these applications can help to minimize these risks. In this paper, we present such a firewall based on anomaly detection that aims to detect anomalous HTTP requests using One Class SVM classifier. Our work uses expert knowledge about the HTTP request structure to build feature extraction methods that improve the detection rates. We include a link to the online repository that contains the code of our implementation for the purpose of reproducibility and extensibility.*

1. Introduction

Web applications have become mainstream in the last decade. Their vulnerabilities pose a great risk because they can easily be exploited by malicious attackers through the Internet. Web Application Firewalls (WAF) can be placed in front of these applications to minimize the risk of attacks [Torрано-Giménez 2015].

According to the detection method, a WAF can be either *misuse-based*, when it looks for known attack signatures, or *anomaly-based*, when it aims to differentiate between normal and anomalous HTTP requests, where the anomalies can include malicious attacks [Torрано-Giménez 2015]. In order to accomplish this differentiation task, the WAF needs to recognize characteristics about the individual HTTP requests. This kind of WAF has two phases, namely training and detection. During the training phase, it extracts models from observed normal requests. During the detection phase, it compares the incoming requests to these models and flags them as anomalous if they deviate significantly from normal requests previously seen. A great strength of this approach lies with the fact that the WAF only needs to be retrained if there are changes in the applications it protects and not because of the appearance of new attacks [Kruegel and Vigna 2003].

For the anomaly detection process of the WAF, one possible strategy is to face it as a classification problem with a single positive class, denominated One Class Classification (OCC) problem [Khan and Madden 2014]. This way the normal HTTP requests belong to this single positive class and all the anomalies will be categorized as not belonging to this class. This OCC approach has the advantage of not needing labeled anomalous data for training, only normal requests are required. One methodology to solve OCC problems is to employ tools from the area of Machine Learning, one of those being the *One-Class SVM*, which has been used with great success for this kind of problems [Khan and Madden 2014]. The characteristics of HTTP requests need to be expressed as numeric feature vectors in order to be used by this classifier.

In this article, we present a WAF based on anomaly detection that uses features selected with expert knowledge about HTTP requests and also uses *One-Class* SVM classifier to detect anomalous requests. In Section 2 we present related works, in Section 3 we describe our proposed WAF, Section 4 contains our test results and we finish this paper with some conclusions and ideas for future work in Section 5.

2. Related works

In [Kruegel and Vigna 2003] and [Kruegel et al. 2005] the authors use expert knowledge about the structure of HTTP requests in their anomaly detection system. They describe how the query string of an HTTP request consists of an ordered list of n pairs of parameters and their corresponding values and how they use statistical methods to build anomaly models of the values for each of the parameters. Some of the models they use are the length of the values, their character distribution, structural inference, parameter presence, among others. For example, one of their models describes the string length of the different values that appear for the parameter *username* in the URL */login*. The approach of these authors differs from ours in that they use statistical methods to detect anomalous requests while we employ *One-Class* SVM. Similar HTTP features and statistical detection methods are used in [Torran-Giménez 2015].

The *One-Class* SVM has been used successfully in many areas, including text classification, face recognition, spam detection, anomaly detection, among others [Khan and Madden 2014]. In [Tran et al. 2004], network traffic statistics are computed with *tcpreplay* and fed to a *One-Class* SVM to detect anomalous network packets. In [Perdisci et al. 2006], byte n -grams are extracted of the payload of network packets and passed to an ensemble of *One-Class* SVMs to detect mimicry attacks. The authors in [Parhizkar and Abadi 2015] extract a fixed number of features from HTTP requests and use an ensemble of *One-Class* SVMs to detect attacks in web traffic. This last paper also uses the same data sets we employ, and later we compare our results with theirs. It is worth noting that the three cited works use a fixed number of features, while in our WAF this number depends on the quantity of parameters in the training data.

3. Proposal

In this section, we describe the four main parts that make up our WAF, namely a routing step, a data preprocessing step, a classification step and lastly a response step. Our implementation was done in the programming language *Python* using the Machine Learning library *scikit-learn* and the code is accessible in our online repository under <https://github.com/nico-ralf-ii-fpuna/paper>.

3.1. Routing step

Since requests to the same URL show a greater similarity to each other than to requests to other URLs, our routing step groups the incoming requests by URL and HTTP method. This way during the training phase the anomaly models can be built independently for each group, resulting in a more precise description of the normal requests within each group. Consequently, during the detection phase there are more accurate models of normal requests available, which help to identify requests that are anomalous within their corresponding groups, even though they would be considered normal in other groups.

3.2. Data preprocessing step

Our data preprocessing step extracts a vector of numeric features from each HTTP request, which is then passed to the classification step. We use feature extraction methods that yield a total of 10 numeric features. Four of our features indicate total length, number of digits, number of letters and number of non-alphanumeric characters. We based these on [Kruegel and Vigna 2003] and [Nguyen et al. 2011], adding some own extensions. Another five features, also based on [Kruegel and Vigna 2003], represent the bins of a method called character distribution. Our last feature represents the entropy according to Shannon [Dobrushin and Prelov 2011], an idea taken from [Nguyen et al. 2011].

Our feature extraction methods are applied to the whole request, including HTTP method, URL, query string, headers, and body. Additionally, we employ the previously mentioned approach used in [Kruegel and Vigna 2003] and apply our extraction methods on each of the parameter values in the query string. We extend this approach to also analyze the parameter values in the body of the requests if there are any.

The obtained feature vector that represents a request will have $m \times (1+n)$ features. Here m is the number of features returned by our feature extraction methods for each value. The 1 represents the whole request and n is the number of parameters that are present in the query string and body. It's important to note that n might be different for each group of URL and HTTP method. During the training phase, all the different parameters from a group of requests are listed and given a fixed order for building the feature vector. This way, during the detection phase our WAF extracts the features of the values whose parameters have been seen in the training phase, assuring that their position within the vector is preserved. For example, if during training our WAF gets POST request which all have only the parameters *username* and *password* in its body, then $m = 10$ and $n = 2$, resulting in each request being represented by a vector of 30 features.

We are aware that our analysis of parameter values during the detection phase is only applied to parameters observed in training. This gives way to the risk of an intruder including an attack inside the value of a previously unseen parameter. But since our feature extraction methods are also applied to the whole request, these attacks can still be detected and do not simply bypass our WAF.

3.3. Classification step

During the training phase, our WAF scales and normalizes the numeric feature vectors produced by the previous step and trains one *One-Class* SVM classifier for each group of URL and HTTP method. In this high dimensional vector space, the classifier tries to find boundaries to the regions in which the feature vectors of normal request reside, drawing these borders as tight as possible to exclude future vectors of anomalous requests, but loose enough to still include future normal requests not seen in the training phase [Perdisci et al. 2006]. This way the WAF obtains a model, a trained classifier, that generalizes the characteristics of normal HTTP requests within each group.

During the detection phase, the incoming requests are routed to their corresponding group, the feature extraction methods are applied and the trained classifier for that group checks if this new feature vector falls inside the established boundaries, marking it as normal if it does. All other requests will be denoted as anomalous.

The *One-Class* SVM takes a few parameters that influence its behavior. One of these parameters, denominated nu or ν , is used to set an upper bound to the fraction of training requests that may fall outside of the established boundaries. This gives the classifier some flexibility when drawing the borders in order to achieve higher generalization capabilities and also achieves robustness in case of some anomalies in the training data [Schölkopf et al. 2001]. Sometimes it can be difficult to find boundaries in the given vector space, so the classifier can use a so called kernel function to simulate a higher dimensionality in which it is easier to draw the separating borders. We tested three kernels, namely the linear, the polynomial and the *Radial Basis Function* (RBF) kernel. We obtained the best classification results with the last one. The RBF kernel also has a parameter called $gamma$ or γ that influences the shape of the boundaries [Tran et al. 2004].

3.4. Response step

Our WAF can be configured to respond in different ways to the classification results. If a request is considered normal, it is forwarded to its intended destination. Otherwise, the WAF logs the result and optionally can also block that request, preventing possible attacks from reaching the applications. Our implementation could be extended, for example, to send alarms about anomalies to the people responsible for the system.

4. Experiments and results

For the quantitative evaluation of the performance of our WAF, we used the public data sets CSIC 2010 [Torrano-Giménez et al. 2010] and CSIC TORPEDA 2012 [Torrano-Giménez et al. 2012], which contain labeled HTTP requests to an e-commerce web application. Given that our WAF groups the requests by URL and HTTP method, we grouped the requests from both data sets by these criteria and selected those groups that had more than 100 samples of each of the two categories, normal and anomalous. This left us with 18 groups, totaling 40,130 normal and 42,444 anomalous HTTP requests.

4.1. Detection effectiveness test

In a first test, we used the aforementioned data to measure the detection effectiveness of our WAF. To demonstrate the added value of analyzing the parameter values, we tested two different scenarios. Firstly, we applied our feature extraction methods only to the whole request, obtaining a fixed number of 10 features. Secondly, we included the analysis of parameter values, yielding different number of features for each group, as described in the previous section. Additionally, for the selection of ν and γ for each group, we tested values in the range $[0.0001 : 0.1]$ and chose those that gave the best result in each group. Table 1 shows average and standard deviation of true positive rate (TPR), false positive rate (FPR) and F_1 score of our results. F_1 score uses the number of true positives (TP), false positives (FP) and false negatives (FN), and it is expressed as $F_1 = \frac{2TP}{2TP+FP+FN}$, where values closer to 1 indicate better results.

Table 1. Results of detection effectiveness test for all 18 groups

Scenario	average TPR	average FPR	average F_1	best F_1
using only the whole request	0.91 ± 0.11	0.31 ± 0.34	0.82 ± 0.18	1.00
including parameter analysis	0.95 ± 0.05	0.09 ± 0.11	0.93 ± 0.07	1.00

Our results show that the second scenario improves the overall detection, with TPR and F_1 raising from 0.91 and 0.82 to 0.95 and 0.93 respectively and FPR lowering from 0.31 to 0.09, even though both scenarios have groups that achieve a perfect score. These results demonstrate the usefulness of including the analysis of parameter values.

Comparing our results with related works that use the same data sets that we employ, we find that in [Parhizkar and Abadi 2015] a TPR of 0.96 and a FPR of 0.03 is reported for the ensemble of *One-Class* SVMs. Two popular *misuse-based* WAFs, Mod-Security¹ and PHPIDS², were used with default rules on the first data set, CSIC 2010, obtaining a TPR of only 0.55 and 0.29 respectively [Giménez and Cappo 2015], significantly lower than the results we achieved with our WAF.

4.2. Response time analysis

Our second test aimed to quantify the effects our WAF has on the response time of the web applications it protects. To that end, we set up a simple application and measured the response time in three different scenarios; in one the requests went directly to the application, in the second the traffic was routed through our WAF but with detection disabled, and in the third scenario with enabled detection. The second and third scenarios show an increase of 2 and 4 ms respectively when compared to the scenario without the WAF. This delay is only a small increase compared to the overall roundtrip time of HTTP traffic, so our WAF should not have noticeable effects on the response time.

4.3. Training time analysis

In our third test, we analyzed how the training time of our WAF relates to the amount of requests used. Since groups may have different durations because of an unequal number of features, the highest numbers give an upper bound. We observed that the time per request stays close to 2 ms in our test setup with 10,000 requests. Our numbers show that the training time has an almost linear relationship to the amount of training data.

5. Conclusions

The WAF we present in this article uses the *One-Class* SVM classifier to tackle the task of detecting anomalous HTTP requests in order to protect web applications from possible attacks. Following and extending the ideas obtained from other authors, we built feature extraction methods that take advantage of the structure of HTTP requests, specifically the values of individual parameters, to represent these requests with vectors which contain more useful information. Our tests show a significant improvement in the detection results when including the analysis of parameter values. Furthermore, our implementation shows that it is fast enough to be used as a WAF, even though the minimization of the processing time was not our primary goal and further optimizations could be made.

One contribution of our paper is the adaptation and extension of the parameter value analysis presented in [Kruegel and Vigna 2003]. The anomaly models employed by the cited authors do not produce numeric vectors for each request, and so we adapted them to make feature extraction methods that are useful for Machine Learning tools, in our case specifically the *One-Class* SVM. As another contribution, we leave a link in Section 3 to

¹www.modsecurity.org

²<https://github.com/PHPIDS/PHPIDS>

the online repository that contains the code of our implementation, which includes the WAF and the tests mentioned in this paper, so that other authors can reproduce our results and have a starting point for further research.

Future works could look into finding additional features to detect anomalous requests. Another research area is other classification tools to be used instead of the *One-Class SVM* to tackle the OCC task. Additionally, our work lacks automatic selection of the values for ν and γ of the classifier and further inquiries can be made in that direction.

References

- Dobrushin, R. and Prelov, V. (2011). Entropy - encyclopedia of mathematics. <http://www.encyclopediaofmath.org/index.php?title=Entropy&oldid=15099>. Accessed: August-2017.
- Giménez, J. and Cappo, C. (2015). Http-ws-ad: Anomaly detector oriented to web applications and services. In *Computing Conference (CLEI), 2015 Latin American*. IEEE.
- Khan, S. and Madden, M. (2014). One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374.
- Kruegel, C. and Vigna, G. (2003). Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*. ACM.
- Kruegel, C., Vigna, G., and Robertson, W. (2005). A multi-model approach to the detection of web-based attacks. *Computer Networks*, 48(5).
- Nguyen, H. T., Torrano-Gimenez, C., Alvarez, G., Petrović, S., and Franke, K. (2011). Application of the generic feature selection measure in detection of web attacks. In *Computational Intelligence in Security for Information Systems*, pages 25–32. Springer, Berlin, Heidelberg.
- Parhizkar, E. and Abadi, M. (2015). Oc-wad: A one-class classifier ensemble approach for anomaly detection in web traffic. In *2015 23rd Iranian Conference on Electrical Engineering (ICEE)*, pages 631–636. IEEE.
- Perdisci, R., Gu, G., and Lee, W. (2006). Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 488–498. IEEE.
- Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.
- Torrano-Giménez, C. (2015). *Study of stochastic and machine learning techniques for anomaly-based web attack detection*. PhD thesis, Universidad Carlos III de Madrid.
- Torrano-Giménez, C., Pérez, A., and Álvarez, G. (2012). Csic torpeda 2012 http data sets. <http://www.tic.itefi.csic.es/torpeda>. Accessed: July-2017.
- Torrano-Giménez, C., Pérez Villegas, A., and Álvarez Marañón, G. (2010). Csic 2010 http data sets. <http://www.isi.csic.es/dataset/>. Accessed: July-2017.
- Tran, Q.-A., Duan, H., and Li, X. (2004). One-class support vector machine for anomaly network traffic detection. *China Education and Research Network (CERNET), Tsinghua University, Main Building*, 310.

Detecção de *Botnets* através da Análise do tráfego DNS e Engenharia Reversa

Juliano Stolpe

Universidade Regional Integrada do Alto Uruguai e das Missões (URI)
Departamento de Engenharias e Ciência da Computação – Santo Ângelo, RS - Brasil.

jstolpe.nti@gmail.com

Abstract. Some botnets uses dynamic addressing techniques such as ip-flux and domain-flux for communication between bots and the command and control server, thus making them more robust for operation and therefore more difficult for detection. This paper details a method to perform botnets detection through DNS traffic analysis along with reverse engineering of malicious code. Such method is a set of procedures to be followed for identification of hosts characterized as bots. Two case studies were presented that were successful with the application of this method.

Resumo. Algumas botnets utilizam-se de técnicas de endereçamento dinâmico como ip-flux e domain-flux para a comunicação entre os bots e o servidor de comando e controle, tornando-as assim mais robustas para a operação, e consequentemente mais difíceis para a detecção. Este artigo detalha um método para realizar a detecção de botnets através da análise do tráfego DNS junto com a engenharia reversa de código malicioso. O método é um conjunto de procedimentos que devem ser seguidos para a obtenção de hosts caracterizados como bots. São apresentados dois estudos de caso que obtiveram êxito com a aplicação deste método.

1. Introdução

Devido ao acesso livre e distribuído do protocolo DNS, aplicações maliciosas também podem fazer consultas para realizar ataques, dentre elas *botnets* que podem ser definidas como um conjunto de máquinas comprometidas que permitem ao atacante o controle remoto dos recursos computacionais para realizar atividades fraudulentas ou ilícitas [McCarty 2003b, Freiling et al. 2005]. Tais máquinas utilizam um software chamado de *bot* (da palavra robô), o qual liga os computadores infectados a uma infraestrutura de Comando e Controle (C&C).

Alguns trabalhos propuseram sistemas, ferramentas e arquiteturas para detecção e mitigação de *botnets*, como [Ceron J. 2010] que definiu uma arquitetura baseada em assinatura de rede de máquinas comprometidas por bots, [Laufer 2005] propôs um sistema de rastreamento de pacotes para descobrir a origem de ataques, [Hossain 2010] propôs a mineração do tráfego DNS para detecção de aplicações de envio de Spam, e [Kaio 2014] apresenta uma metodologia utilizando teoria dos grafos para distinguir consultas padrões de anômalas no tráfego DNS.

A Tabela 1 faz uma compilação dos trabalhos relacionados ao tema, demonstrando as diferenças e semelhanças entre os trabalhos relacionados para o desenvolvimento do método proposto.

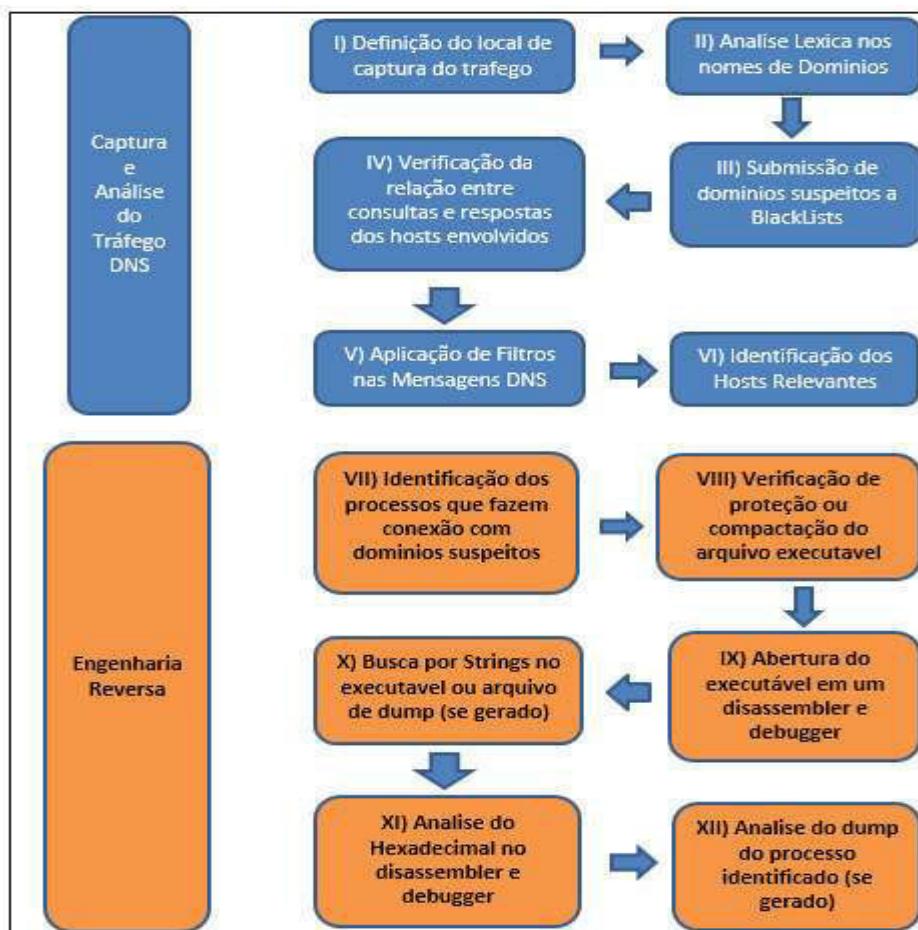
	Ass. de Rede	Eng. Reversa	Rast. de Pacote	Met. Hibrido	Trafego Rede	DNS
Ceron 2010	X					
Laufer 2005			X			
Cunha 2011		X		X	X	
Hossain 2010					X	X
Proposto		X		X	X	X

Tabela 1 – Comparativo entre o método proposto e os trabalhos relacionados

O presente artigo tem como objetivo, definir um método híbrido composto pela análise do tráfego de rede e a engenharia reversa, para detecção de *botnets* que utilizam o serviço de DNS para se proliferar e controlar seus *bots*. A intenção deste trabalho é ser um guia para a detecção com êxito de *botnets* que especificamente utilizam o serviço de DNS. O artigo está organizado da seguinte forma: A seção 2 apresenta o detalhamento do método proposto, a seção 3 apresenta um estudo de caso e os resultados obtidos, e a seção 4 apresenta as conclusões.

2. Método Proposto

O método é dividido em duas partes: a primeira parte trata da captura e análise do tráfego DNS. A segunda parte trata da engenharia reversa feita no código malicioso. A Figura 1 exibe o diagrama da ordem de execução e a própria estrutura do método. Cabe ressaltar que outras ferramentas também podem ser utilizadas para a execução do método, alem das citadas, desde que cumpram o propósito definido em cada etapa.

**Figura 1 - Diagrama Roteiro e Estrutura do Método Proposto**

I) Definição do local de captura do tráfego

O host necessita fazer a interligação entre a rede interna e a rede externa, sendo assim geralmente atribuído a um Proxy, Firewall, Gateway ou Roteador executando algum sistema operacional derivado de Unix. Este host pode possuir uma ou mais placas de rede.

II) Análise Léxica nos nomes de Domínios

A análise léxica é o processo de verificação da formação dos nomes de domínios, isso implica em perceber se os caracteres ou símbolos que formam o nome de domínio não estão fora do padrão de nomes comprehensíveis.

III) Submissão de domínios suspeitos a *BlackLists*

Sites descritos como *blacklists* possuem softwares que fazem a verificação sobre o domínio, e-mails, endereços de IP que foram previamente denunciados como disseminadores de mensagens consideradas SPAM.

IV) Verificação da relação entre consultas e respostas dos hosts envolvidos

O entendimento e classificação de comportamentos através do tráfego DNS podem ser obtidos por dois métodos básicos: A relação de consultas entre os hosts, e como as consultas DNS são realizadas, já que o registro de recursos denota o objetivo da consulta.

V) Aplicação de Filtros nas Mensagens DNS

Quanto ao como as consultas DNS são realizadas, devem ser levados em consideração: Estações que enviam consultas mal formadas, respostas que possuem erros de nomes verificando o MXDOMAIN, e uma grande quantidade de mensagens de erro geradas com um espaço de tempo reduzido.

VI) Identificação dos Hosts Relevantes

Quando um hosts é reincidente em quase todas as verificações e filtros executados.

VII) Identificação dos processos que fazem conexão com domínios suspeitos

Após identificar os hosts relevantes, devem ser verificados os status das conexões ativas nestes hosts e identificar o PID do processo que esta em execução e fazendo a conexão.

VIII) Verificação de proteção ou compactação do arquivo executável

Grande parte dos *malwares* sofrem alterações após a compilação. Eles são protegidos ou compactados, para que não sejam detectados por aplicações anti-virus.

IX) Abertura do executável em um *disassembler* e *debugger*

É preciso desmontar o programa e verificar se o domínio ou endereço IP está contido no executável, ou em um arquivo de *dump* gerado a partir de um processo.

X) Busca por *Strings* no executável ou arquivo de *dump* (se gerado)

As *strings* possuem o endereço IP do servidor, ou o domínio ao qual o software deve se conectar, elas também revelam muito sobre o comportamento do software.

XI) Analise do Hexadecimal no *disassembler* e *debugger*

As vezes pode acontecer de as *strings* não revelarem tudo sobre o programa, tendo a alternativa de verificação do hexadecimal gerado.

XII) Analise do *dump* do processo identificado (se gerado)

Este passo é opcional mas importante, visto que o *dump* contém todo o contexto no qual o processo sofreu o despejo.

3. Estudo de Caso – Detecção de uma *botnet*

Neste estudo de caso é apresentado a detecção de uma *botnet* através da análise do tráfego de DNS. Para demonstrar este estudo de caso, foram configurados um servidor DNS e um servidor de comando e controle (C&C). O domínio criado foi chamado de “botnetstolpe.net” fazendo-se com que o servidor DNS fosse autoritativo sobre este domínio. O ambiente no qual o método foi aplicado é de produção com tráfego de dados reais, composto por aproximadamente 60 dispositivos conectados à internet, dentre eles estações de trabalho, notebooks, servidores e dispositivos móveis. Na configuração do servidor de comando e controle (C&C) utilizou-se o programa DarkDDoS instalado em um sistema Windows 7.

O inicio da aplicação do método começa com a captura do tráfego DNS, sendo executado apartir de um terminal de comando em ambiente Unix, no gateway principal da rede:

tcpdump -i eth1 src port 53 or dst port 53 -w capturaDNS-Dia_Hora.pcap

Posteriormente foi efetuado a analise léxica do nome de domínio. Um endereço IP pode ser identificado como banda larga caso o nome de domínio seja composto por palavras como cpe, vivax, adsl, modem, virtual, e isto serve como um critério básico de exclusão para estes domínios, quando há uma desconfiança na formação dos nomes. O *sniffer Wireshark* permite gerar um arquivo contendo a tradução dos nomes de domínios para endereços IPs, através do resolvedor de endereços na guia de estatísticas. Os dominios listados com o Wireshark foram inseridos em um arquivo de texto e comparados através de um comando linux, com uma lista de dominios baixados do site [<http://urlblacklist.com>], que divide os nomes listados em categorias, como por exemplo, *hacker, mail, spyware, proxy, ddos* entre outros. O comando utilizado foi: ***grep -xf dominios.txt dominiosBaixados.txt***, que retornou em tela no terminal, apenas os nomes de domínios que faziam parte das *blacklists* categorizadas.

Posteriormente foram verificados se cada nome retornado fazia parte de outras *blacklists*, através do site [<http://mxtoolbox.com/blacklists.aspx>] que para alguns domínio retornou positivo. Após obter os domínios classificados como ameaça, efetuou-se a verificação no *dump* de rede, de quais endereços IPs consultaram os domínios listados. O comando utilizado no *Wireshark* para tal verificação foi ***dns.qry.name contains "nomeDoDominio"***.

O próximo passo é entender o padrão de comunicação entre os hosts identificados como possíveis *bots*, sendo necessário relacionar o total de consultas realizadas, com o total de respostas, isto é possível através da visualização de conversações efetuadas entre os hosts, que tambem são mostradas na guia de estatísticas do Wireshark. Fazendo a relação entre as consultas, foi examinado o formato das mesmas, verificando-se a integridade e conteúdo de cada uma. A primeira tarefa foi a verificação da integridade das consultas, através das flags das mensagens DNS, como: “***dns.flags.rcode == 1***” e “***dns.flags.rcode ==3***” no filtro do *Wireshark*, é possível obter as consultas que retornaram com o status de MXDOMAIN (*No Such Name*), que podem corresponder a servidores de comando e controle. Outro modo é através da análise visual do *dump* de rede, procurando por diferentes endereços IPs consultando um único endereço IP, em um intervalo de tempo muito curto. Os hosts que apresentarem estes comportamentos serão tratados como “hosts relevantes”, pois apresentam o comportamento de um servidor que envia um comando para que os clientes(*bots*) respondam a um comando, todos ao mesmo tempo. De posse dos hosts relevantes e os domínios suspeitos, efetuou-se nova análise no

trafego DNS em todos arquivos de captura, fazendo-se a correlação entre quais os hosts que acessaram quais domínios, assim como a frequência em que acessavam. O comando para a verificação dos domínios em cada arquivo de captura foi: ***dnsqry.name contains "nomeDoDominio"***.

Foi necessário relacionar o total de consultas realizadas com o total de respostas, procurando por diferentes endereços IPs consultando um único endereço IP, em um intervalo de tempo muito curto. Este fato pode ser observado em dois momentos no trafego DNS capturado em um dos dias, onde alguns endereços IPs consultam o servidor DNS no mesmo segundo. No primeiro momento as 10:03:03 acontece a consulta de diferentes endereços procurando pelo servidor do domínio “botnetstolpe.net”.

As Figuras 2 e 3 exibem o momento do ataque, com uma diferença de 24 segundos entre a consulta dos *bots* pelo domínio, e o ataque propriamente dito.

Executou-se um ataque de DoS (Negação de Serviço) em um roteador na rede. No segundo momento as 11:35:01 acontece o segundo ataque, e o trafego demonstra as consultas efetuadas pelos *bots*.

10:03:03	192.168.1.97	192.168.1.210	DNS	86 Standard query 0x648a A serverbot.botnetstolpe.net
10:03:03	192.168.1.210	192.168.1.97	DNS	150 Standard query response 0x648a A serverbot.botnetstolpe.net A 192.168.1.125
10:03:03	192.168.1.70	192.168.1.210	DNS	86 Standard query 0x16f8 A serverbot.botnetstolpe.net
10:03:03	192.168.1.210	192.168.1.70	DNS	150 Standard query response 0x16f8 A serverbot.botnetstolpe.net A 192.168.1.125
10:03:03	192.168.1.63	192.168.1.210	DNS	86 Standard query 0x98b4 A serverbot.botnetstolpe.net
10:03:03	192.168.1.210	192.168.1.63	DNS	150 Standard query response 0x98b4 A serverbot.botnetstolpe.net A 192.168.1.125
10:03:03	192.168.1.95	192.168.1.210	DNS	86 Standard query 0x81f2 A serverbot.botnetstolpe.net
10:03:03	192.168.1.210	192.168.1.95	DNS	150 Standard query response 0x81f2 A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.66	192.168.1.210	DNS	86 Standard query 0x644a A serverbot.botnetstolpe.net
11:35:01	192.168.1.210	192.168.1.66	DNS	150 Standard query response 0x644a A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.70	192.168.1.210	DNS	86 Standard query 0x7ac9 A serverbot.botnetstolpe.net
11:35:01	192.168.1.95	192.168.1.210	DNS	86 Standard query 0x4b45 A serverbot.botnetstolpe.net
11:35:01	192.168.1.210	192.168.1.70	DNS	150 Standard query response 0x7ac9 A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.210	192.168.1.95	DNS	150 Standard query response 0x4b45 A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.97	192.168.1.210	DNS	86 Standard query 0x055e A serverbot.botnetstolpe.net
11:35:01	192.168.1.210	192.168.1.97	DNS	150 Standard query response 0x055e A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.63	192.168.1.210	DNS	86 Standard query 0x1102 A serverbot.botnetstolpe.net
11:35:01	192.168.1.210	192.168.1.63	DNS	150 Standard query response 0x1102 A serverbot.botnetstolpe.net A 192.168.1.125

Figura 2 - Trafego suspeito no servidor DNS

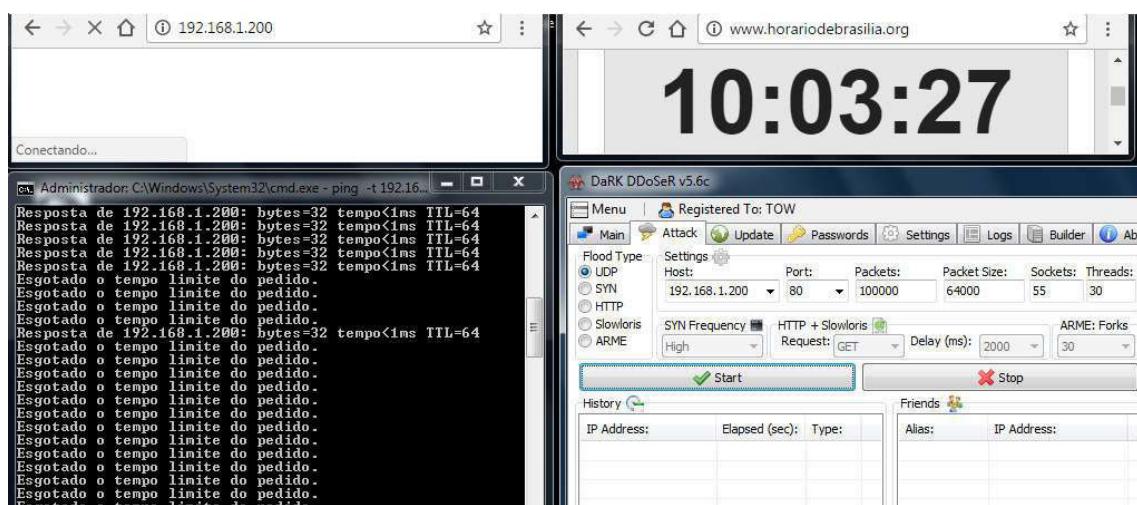


Figura 3 - Ataque de negação de serviço (DoS) no roteador

Para a identificar os hosts relevantes, efetuou-se o cálculo da quantidade de vezes que um determinado endereço IP é exibido em um dia e em um filtro. Ao todo obteve-se 8 hosts que apresentaram alguma característica inerente aos filtros. E neste caso o fator que mais contribuiu para a relevância do host ser considerado suspeito, foi a sua inserção no grupo dos hosts acessando ao mesmo tempo um único IP.

Ao inicializar o sistema operacional dos hosts, que neste momento são considerados bots verificou-se o status das conexões ativas através do comando “netstat –o” para identificar o PID do processo que está fazendo a conexão. A Figura 4 mostra que há uma conexão

de um processo chamado botClient.exe do host 192.168.1.70, na porta 5555 de um servidor chamado SERVERBOT.

```

Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [versão 6.1.7601]
Copyright © 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Windows\system32>netstat -o

Conexões ativas

  Proto  Endereço local          Endereço externo        Estado      PID
    TCP   192.168.1.70:49158      SERVERBOT:5555       ESTABLISHED  2808

C:\Windows\system32>

```

Proto	Endereço local	Endereço externo	Estado	PID
TCP	192.168.1.70:49158	SERVERBOT:5555	ESTABLISHED	2808

Figura 4 - Conexão de um bot com o Servidor

Após obter o arquivo botClient.exe o mesmo foi submetido ao programa EXEinfoPE para verificação de possível ofuscação no código. Entretanto foi possível observar que o mesmo não possuia nenhuma proteção, e que este arquivo foi desenvolvido na linguagem de programação Delphi.

Ao abrir o executável no *disassembler* IDA PRO e no *debugger* OllyDBG, não foram encontradas nenhuma referência ao servidor, como nome do servidor, porta ou endereço IP, tanto na busca por *strings* quanto no hexadecimal. Apenas foram encontradas referências a classes de programação, chaves do registro do Windows, e *System Calls* do sistema operacional.

Entretanto ao fazer o *dump* de memória com o processo em execução, gerou-se um arquivo chamado botClient.DMP, e este foi submetido ao IDA PRO para análise.

Ao verificar este arquivo foi possível constatar o endereço do servidor de comando e controle chamado “serverbot.botnetstolpe.net”, ao qual o *bot* conectava-se, como mostra a Figura 5.

```

seg000:000B7EA0 00 02 00 18 00 00 00 00 00 00 00 00 1B 00 00 00 73  .-+-----+...S.
seg000:000B7EB0 00 65 00 72 00 76 00 65 00 72 00 62 00 6F 00 74  .e.r.v.e.r.b.o.t
seg000:000B7EC0 00 2E 00 62 00 6F 00 74 00 6E 00 65 00 74 00 73  ...b.o.t.n.e.t.s
seg000:000B7ED0 00 74 00 6F 00 6C 00 70 00 65 00 2E 00 6E 00 65  .t.o.l.p.e....n.e
seg000:000B7EE0 00 74 00 00 00 01 00 00 00 00 00 00 00 50 00  .t.... .....P,
seg000:000B7FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Figura 5 - Domínio encontrado no Arquivo de Despejo

Ou seja, neste caso a detecção do domínio que liga o *bot* ao servidor de comando e controle, só foi possível através do arquivo de despejo de memória, executado sobre o processo em execução. Isso demonstra que, mesmo sendo parte opcional do método proposto, são obtidos resultados satisfatórios.

4. Conclusão

O presente trabalho teve por objetivo descrever e aplicar um método híbrido composto pela análise do tráfego DNS em conjunto com a engenharia reversa de código malicioso, para detecção de botnets.

Após a aplicação do método, o mesmo mostrou-se satisfatório, pois além do objetivo principal que é a detecção de *botnets*, ele também consegue distinguir anomalias no tráfego de DNS, quando estas são causadas por algum tipo de *malware*.

A contribuição acadêmica deste trabalho é importante por demonstrar que em algum momento entre a comunicação de programas maliciosos, com um servidor ou controlador de *malwares*, é possível detectar comportamentos estranhos ou anômalos para mitigar qualquer espécie de ameaça que utiliza o protocolo DNS contra uma rede de computadores.

Referências Bibliográficas

- [Araújo et al.2010] J. M. Araújo Filho (2010, pt. 2) Ciberterrorismo e Cibercrime: o Brasil está preparado?
- [Binsalleh et al. 2010] Binsalleh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., and Wang, L. (2010). On the analysis of the zeus botnet crimeware toolkit. In Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on, pages 31–38.
- [Ceron J., Granville L., Tarouco L] João Marcelo Ceron, Lisandro Zambenedetti Granville, Liane Margarida Rockenbach Tarouco (2010) Uma Arquitetura Baseada em Assinaturas para Mitigação de Botnets.
- [Ceron et al.2010] Ceron, João Marcelo (2010) Arquitetura Distribuída e Automatizada para mitigação de botnets baseada em analise dinamica de malwares.
- [Cunha Neto et al.2011] Cunha Neto, Raimunho Pereira da (2011) Sistema de Detecção de intrusos em ataques oriundos de botnets utilizando metodo de detecção hibrido.- São Luis PPGEE.
- [Craig 2007] Craig A. Schiller, Botnets - The Killer Web App, Singress 2007.
- [Davis et al. 2008] Davis, C., Fernandez, J., Neville, S., and McHugh, J. (2008). Sybil attacks as a mitigation strategy against the storm botnet. In Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on, pages 32–40.
- [Egele et al. 2008] Egele, M., Scholte, T., Kirda, E., and Kruegel, C. (2008). A survey on automated dynamic malware-analysis techniques and tools. ACM Comput. Surv., 44(2):6:1–6:42.
- [Ferreira 2013] Ferreira Pedro, Detecção de Botnets, IPB 2013.
- [Hossain et al. 2010] Soraya Sybele Hossain, Detecção de aplicações envio de Spam através da mineração do tráfego DNS(2010).
- [Kaio 2014] Kaio Rafael, Identificação e Caracterização de Comportamentos Suspeitos Através da Análise do Tráfego DNS. SBSEG 2014.
- [LAUFER et al.2005] LAUFER RAFAEL PINAUD Rastreamento de Pacotes IP contra Ataques de Negação de Serviço [Rio de Janeiro] 2005 XIII, 93 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2005)
- [McCarty 2003b] McCarty, B. (2003b). Botnets: Big and bigger. IEEE Security and Privacy, 1(4):87–90. cited By (since 1996)41.
- [Morimoto 2013] Morimoto, Carlos Eduardo. Servidores Linux – Guia Prático, 1º ed. SulEditores 2013.
- [Mota Filho 2013] Mota Filho, João Eriberto, Análise de Trafego em redes TCP/IP. 1o.ed.Novatec 2013.

Propondo uma Análise de Risco focada na singularidade da Internet das Coisas

Silvio Beskow¹, Érico S. Rocha¹

¹Curso Superior de Tecnologia em Segurança da Informação
Universidade do Vale do Rio dos Sinos (UNISINOS)
Porto Alegre – RS – Brasil

segbeskow@gmail.com, ericor@unisinos.br

Abstract. *The Internet of Things connects countless devices exchanging information among themselves. These devices are produced by various manufacturers with restricted computer resources. They are part of a wide range of human daily activities, thus linking diverse multiple digital resources to the physical world. This article is a fragment of a research in progress. Its main objective is to present the reasons that justify the need for improvements in IoT systems risk analysis. To live up to this challenge, it focuses on the purpose of such devices, on their computational capacity, on the infrastructure they require and on possible impacts on users.*

Resumo. *A Internet das Coisas conecta uma infinidade de dispositivos que trocam informações entre si. Esses dispositivos são produzidos pelos mais variados fabricantes e com recursos computacionais restritos. Eles, por sua vez, fazem parte das mais diversas atividades da vida humana, interligando, assim, os mais diversos recursos digitais ao mundo físico. Este artigo é um fragmento de uma pesquisa em andamento. Seu principal objetivo é apresentar as razões que justificam a necessidade de melhorias na Análise de Risco de sistemas IoT. Para realizar esse desafio, apoia o foco no propósito desses dispositivos, na sua capacidade computacional, na infraestrutura que eles necessitam e nos possíveis impactos causados aos seus usuários.*

1. Introdução

A Internet das Coisas (IoT) é uma evolução do sistema de rede atual. São mais dinâmicas, são interoperáveis e autoconfiguráveis. Além disso, são redes constituídas por “coisas” que possuem identidade, atributos físicos e personalidades virtuais. Integram-se à rede através das suas interfaces inteligentes [Vermesan et al. 2011].

“Nesta época de conectividade eletrônica universal, de vírus e hackers, de espionagem eletrônica e fraude eletrônica, não há realmente um tempo em que a segurança não importa” [Stallings 2014]. Estamos vivendo um novo momento no qual o conhecimento humano, herdado e cultivado de muitas gerações, coloca a humanidade sob uma nova perspectiva evolutiva de atividades na rede para troca de informações.

Os desafios da Internet das Coisas, quanto ao volume de dispositivos, aplicativos e pessoas, fomentam estudos sobre como tornar esta rede mais confiável e convidativa aos seus usuários. A motivação desse estudo se estabelece no fato de que

alguns dos dispositivos (de capacidade computacional limitada) que são conectados à rede, não apresentam os requisitos mínimos para a aplicação das tecnologias de Segurança da Informação disponíveis. Nesse contexto, objetiva-se salientar a necessidade de melhorias na Análise de Risco para destacar as singularidades do sistema durante o planejamento desse sistema.

2. Internet das Coisas

A Internet das Coisas começou a receber maior atenção após a popularização das comunicações sem fio. Isso ocorreu pelo volume de dispositivos (coisas como portas, luzes da casa, eletrodomésticos etc.) que são de uso comum dos seres humanos, ligados das mais variadas formas e para as mais diferentes finalidades.

2.1. Conceito

Os conceitos de IoT propostos por [Singer 2012] tem como base: a) o paradigma que intersecciona os saberes e a tecnologia proposto por [Atzori et al. 2010]; b) a relação entre homens e máquinas, [Greenfield 2010]; c) a posição reativa (conectividade conhecida), e a proativa (entre homens e máquinas e entre máquinas e máquinas), sendo que as comunicações máquina a máquina (M2M) são autônomas, proposta por [Van Kranenburg et al. 2011]¹. Assim, com a reunião desses três conceitos, o autor propõe uma definição operacional da IoT que “*a considera como um paradigma computacional com implicações profundas no relacionamento entre homens e objetos*”[Singer 2012]².

O conceito de que IoT é uma rede dinâmica, composta de coisas físicas e virtuais, únicas, interoperáveis, autônomas, e que tornam as redes inteligentes por serem compostas por dispositivos inteligentes, foi proposto pela Strategic Research Roadmap da Cluster of European Research Projects on the Internet of Things. Essas “coisas inteligentes” podem ou não necessitar de intervenção humana e são capazes de conectar o mundo real e o virtual. Possuem a capacidade de manipular os dados com os quais interagem, assim é necessário observar as questões de privacidade e segurança[Vermesan et al. 2011]³.

Com base nessas definições, propõe-se que a IoT incorpore os aspectos de segurança em suas variadas definições da seguinte forma: objetos inteligentes, privados, que interoperem por meios seguros, com dados suficientes e limitados para realizar as tarefas as quais são designadas e consentidas por seus proprietários.

3. Segurança da Informação

A Segurança da Informação tem sua base na tríade: confidencialidade, integridade e disponibilidade (CID). A partir dessa tríade, ela se aprofunda em todas as áreas que tenham a informação como ativo a ser protegido.

Destaca-se que a segurança inter-redes já é uma tarefa que exige um grande esforço técnico e do negócio. Então, pode-se imaginar se tudo e todos estiverem

¹Retirado do trabalho The Internet of Things, p.9.

²fragmento retirado ipsis litteris p.5.

³Interpretação e resumo do autor de fragmento de texto lido em, p.6.

interconectados? Este fato já é uma realidade em expansão hoje, onde os esforços para a proteção de todos os usuários, seus dados e suas múltiplas interconexões já encontram seus desafios específicos [Stallings 2014].

4. Referencial Teórico

No trabalho *Internet of Things (IoT): Smart and Secure Service Delivery*, a IoT é definida como uma rede que apresenta uma diversidade de dispositivos conectados, com benefícios e vulnerabilidades. Afirma que os serviços para a IoT estão sendo desenvolvidos e implementados sem considerar a segurança [BERTINO et al. 2016]. Sustenta a necessidade do planejamento organizacional e seguro dos sistemas IoT.

O artigo *Managing the risk of the Internet of Things*, define IoT como uma rede composta de dispositivos de capacidade computacional limitada. Entende-se aqui, aqueles objetos corriqueiros, de uso diário (ex.: exemplo uma cafeteira) e que não são considerados computadores. Todavia, a Internet das Coisas faz uso da computação tradicional enviando ou recebendo dados como, por exemplo, um medidor de temperatura enviando e recebendo dados de um servidor na nuvem [Sorebo 2015]. A proposta do artigo é trabalhar com uma Gestão de Risco mais focada no impacto futuro e não na vulnerabilidade. Para tanto, afirma que é preciso, em um primeiro momento, definir qual o propósito de uso do dispositivo IoT. Depreende-se que ele pode ser usado para vários objetivos distintos e variar o tipo de risco em decorrência do seu propósito. Assim, uma linha desse processo seria: definir o uso, verificar as estruturas de suporte necessárias, analisar as restrições técnicas dos dispositivos, verificar o alinhamento com os objetivos do negócio e estimar o seu valor operacional.

Irshad (2016), avaliou sistematicamente e revisou os frameworks de Gerenciamento de Segurança da Informação relacionados à Internet das Coisas. Incentiva a definição de uma estratégia de governança com foco na segurança dos seus ativos [Irshad 2016]. Seu estudo é relevante para o presente artigo por apresentar uma Arquitetura IoT relacionando as camadas e os tipos de tecnologias. Também se pode salientar a questão do enquadramento adequado dos dispositivos.

Rob Clyde, um dos diretores do conselho da *Information Systems Audit and Control Association* (ISACA), assinalou o potencial que a IoT e a Realidade Aumentada têm de se tornarem uma fonte sem precedentes de valor e oportunidade, bem como o potencial de risco para a segurança das informações. Acrescentou que indivíduos e empresas precisam acelerar seus esforços para desenvolver essas tecnologias, ao mesmo tempo que aprendem a gerenciar os seus riscos. Esse aprendizado é de grande importância para que não comprometa a capacidade de inovação das empresas [ISACA 2016]. O diretor ressalta que não deve haver simplesmente um não uso da tecnologia em função das suas fraquezas, mas um esforço de gerência para minimizar os riscos. Nesse caso, tem-se mais uma referência à gestão de segurança como uma alternativa.

Em *Identity Transformed as Internet of Things Invades the Workplace*, o *Chief Information Officer* (CIO), de uma certa organização, solicita um *pentest* à uma organização terceira que comprometa um computador da empresa explorando uma falha de segurança de um teclado wireless conectado a esse computador. O artigo propõe que os dispositivos que estão no espaço de trabalho devem ser identificados

[Lemos 2017]. A existência de uma política de segurança eficaz depende de uma análise de risco tão profunda, quanto mais pervasivos são os dispositivos que poderão ser conectados à rede da organização.

5. Problemas e Desafios Gerais da IoT

Os problemas básicos de segurança na IoT estão na limitação de hardware (processamento, armazenamento e energia) e na interpolaridade dos diversos dispositivos. Abaixo são destacados alguns agravamentos:

- Cada dispositivo apresenta um risco potencial e pode tornar-se um vetor de ataque [BERTINO et al. 2016];
- Falta de identificação dos dispositivos IoT nas organizações;
- A capacidade das redes e a transmissão de dados com relação à segurança dos protocolos de rede e aplicações nos dispositivos IoT [Heer et al. 2011];
- A grande capacidade de processamento, gerenciamento e distribuição de chaves que são exigidas para implementar a criptografia;
- Falta de uma padronização para um desenvolvimento sustentável [Atzori et al. 2010];

Devido à ampla variedade de dispositivos, de fabricantes, às restrições de hardware [Sorebo 2015] e algumas estruturas de redes, destacam-se os desafios considerados importantes:

- Capacidade de adequação padronizada de comunicação entre diversos dispositivos e diferentes fabricantes, chamado de Interpolaridade;
- Desenvolvimento de criptografia leve e um sistema adequado de gerenciamento e distribuição de chaves;
- Reflexão sobre “autoria, propriedade e privacidade das informações” [Singer 2012];
- Capacidade de aderir a conformidade com a legislação [Ziegeldorf et al. 2014];
- Privacidade e segurança, sem inviabilizar a troca de dados entre os dispositivos [Guo et al. 2011];
- Estrutura que permita aos proprietários da informação definirem o seu apetite de risco em relação à **privacidade e a segurança**, deixando-os conscientes dos produtos e serviços IoT que estão consumindo [Vermesan et al. 2011].

6. Problema e Desafio Específicos

Defende-se que uma forma de enfrentar os problemas da IoT está na gestão dos seus riscos. Segundo Sorebo (2015), o problema está no fato de que a gestão de risco tradicional está embasada em como os dispositivos são usados. Mesmo na tentativa de compreender as ameaças e, por conseguinte, propor controles, pode-se apontar em distintas direções e ainda assim não conseguir lidar de forma efetiva com o desenvolvimento contínuo de novos dispositivos e suas novas ameaças [Sorebo 2015].

O autor propõe que se tenha um esforço maior de concentração nos impactos catastróficos que o uso de um dispositivo em uma determinada tarefa poderia causar e não nas vulnerabilidades do mesmo. Para isso, propõe deslocar o foco do como os dispositivos são usados, para focar em quais as tarefas eles irão executar. Um exemplo

disso é um sensor de temperatura em um local público mandando informações de uso comum para internet e o mesmo sensor em uma siderúrgica mandando informações de uso privado para internet. Assim, afirma ainda que iniciando uma análise pelo caso de uso, o objetivo comercial pretendido já se destacará. Também havendo mudança de tarefa desse dispositivo, é evidente que o risco deverá ser reavaliado [Sorebo 2015].

O desafio é propor uma melhoria nos frameworks de análise de risco existentes que conteemple todas as questões ímpares que envolvem a IoT, seus dispositivos e estrutura, com base nos estudos do artigo de Sorebo (2015).

7. Justificativa

Bertino et al. (2016) conclui no seu artigo que os sistemas IoT apresentam uma grande quantidade de vulnerabilidades e afirma que há um risco potencial dos seus dispositivos. Ressalta a necessidade da pesquisa e do desenvolvimento necessários para mitigar ameaças, a fim de evitar e resolver as vulnerabilidades.

Embora os esforços empregados por todas as partes interessadas na sua segurança e confiabilidade, a IoT é um problema de segurança em expansão. Um exemplo disso é o caso da planta de uma indústria química que usa sensores para monitorar reações usadas na fabricação de produtos sem a segurança adequada [Irshad 2016]. Ou ainda, a falha de segurança em um teclado wireless [Lemos 2017].

O esforço de estudar melhorias para a análise de risco em IoT está na importância de conhecer as limitações dos dispositivos e antecipar os seus possíveis impactos negativos no negócio. Há uma diversidade imensa de fabricantes, de hardware, de software, de infraestrutura, cada um com a sua singularidade, o que justifica o empenho em rever a prática atual da análise de risco tradicional.

8. Conclusão

Com base nos estudos realizados e nas dificuldades destacadas em diversas literaturas, acredita-se que gerenciar os riscos é uma importante ferramenta de segurança no projeto de sistemas da Internet das Coisas. Pensando nos desafios gerais da IoT, os problemas com relação à segurança e a privacidades ainda necessitam de desenvolvimento. Por essa razão, o desafio específico desse trabalho buscará preencher essas lacunas com melhorias aos frameworks existentes ou metodologia de base ágil de avaliação de riscos para sistemas IoT.

Ela deverá contribuir para a melhor visualização das capacidades técnicas dos dispositivos, de modo a equacionar os objetivos do negócio, as expectativas das partes interessadas e a legislação, respectivamente. A questão que se destaca hoje, com relação à IoT, principalmente da Segurança da Informação em dispositivos e ambientes restritos, prolonga-se por anos, conforme os trabalhos estudados até o momento. Assim, a Internet das Coisas poderá ser implementada com maior segurança através de uma análise de riscos ímpar, que atue sobre o propósito dos dispositivos que compõe o sistema e seus possíveis impactos futuros.

Referências

- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- BERTINO, E., KIM-KWANG RAYMOND, C., GEORGAKOPOLOUS, D., and NEPAL, S. (2016). Internet of things (iot): Smart and secure service delivery. *ACM Transactions on Internet Technology*, 16(4):22:1 – 22:7.
- Greenfield, A. (2010). *Everyware: The dawning age of ubiquitous computing*. New Riders.
- Guo, B., Zhang, D., and Wang, Z. (2011). Living with internet of things: The emergence of embedded intelligence. In *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pages 297–304. IEEE.
- Heer, T., Garcia-Morchon, O., Hummen, R., Keoh, S. L., Kumar, S. S., and Wehrle, K. (2011). Security challenges in the ip-based internet of things. *Wireless Personal Communications*, 61(3):527–542.
- Irshad, M. (2016). A systematic review of information security frameworks in the internet of things (iot). In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on*, pages 1270–1275. IEEE.
- ISACA (2016). Isaca survey shows us consumers see value in augmented reality, but confidence in internet of things knowledge takes a dive. *Business Wire (English)*.
- Lemos, R. (2017). Identity transformed as internet of things invades the workplace. *Information Security*, pages 8 – 12.
- Singer, T. (2012). Tudo conectado: conceitos e representações da internet das coisas. *Simpósio em tecnologias digitais e sociabilidade*, 10.
- Sorebo, G. (2015). Managing the risk of the internet of things. *Control Engineering*, 62(9):DE27–DE30.
- Stallings, W. (2014). *Cryptography and Network Security: Principles and Practice, International Edition: Principles and Practice*. Pearson Higher Ed.
- Van Kranenburg, R., Anzelmo, E., Bassi, A., Caprio, D., Dodson, S., and Ratto, M. (2011). The internet of things. *A critique of ambient technology and the all-seeing network of RFID, Network Notebooks*, 2.
- Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I. S., Mazura, M., Harrison, M., Eisenhauer, M., et al. (2011). Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends*, 1:9–52.
- Ziegeldorf, J. H., Morschon, O. G., and Wehrle, K. (2014). Privacy in the internet of things: threats and challenges. *Security and Communication Networks*, 7(12):2728–2742.